MASTER

**DO NOT MICROFILM
COVER**

**NATIONAL UTILITY FINANCIAL
STATEMENT MODEL (NUFS).**

Volume III of III:
Software Description

FINAL REPORT

Submitted to:
Division of Coal and Electric Power
Energy Information Administration
U.S. Department of Energy

Prepared Under Contract No. DEAC-01-79EI-10579

October 29, 1981

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

ICF INCORPORATED 1850 K Street, Northwest,
Suite 950, Washington, D. C. 20006

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

NATIONAL UTILITY FINANCIAL
STATEMENT MODEL
(NUFS)

VOLUME III of III:

DOE/EIA/10579--T1-Vol.3

DE83 001096

SOFTWARE DESCRIPTION

Final Report

Submitted to:
Division of Coal and Electric Power,
Energy Information Administration
Department of Energy

by

ICF Incorporated

October 29, 1981

26
DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

SUMMARY

This volume contains a description of the software comprising the National Utility Financial Statement Model (NUFS). This is the third of three volumes describing NUFS provided by ICF Incorporated under contract DEAC-01-79EI-10579. The three volumes are entitled:

- Model Overview and Description
- User's Guide
- Software Guide

TABLE OF CONTENTS

	<u>PAGE</u>
I. INTRODUCTION	1
II. NUFFS.FORT	4
III. NUFFS.RPT.FORT	32
Appendix A: NUFFS.FORM1.FORT and NUFFS.FORM1M.FORT	
Appendix B: NUFFS.MEFS.FORT	

I. INTRODUCTION

The major purpose of this last volume is to give a description as well as a listing of the computer code for the National Utility Financial Statement (NUFS) model. The code for NUFS is contained in two FORTRAN source modules entitled NUFS.FORT and NUFS.RPT.FORT. In addition to the FORTRAN code, these modules contain the Job Control Language (JCL) statements necessary to compile and link edit the FORTRAN and create an executable load module. Each module is thus ready to be submitted as a batch job. Directions for creating the load module may be found in Vol. II, Section V.

The central algorithm for NUFS can be broken into three basic stages (see Figure 1). First, the input data files are read and initial calculations are made. This includes determining:

- Builds, generation, fuel costs (MEFS interface input file);
- Base year assets, revenues, expenses (FORM 1/FORM 1M input files); and
- Regulatory climate (user's input file).

Second, all of the financial accounts are calculated. This includes calculating (for each year forecast):

- Book value of new assets;
- Book depreciation;
- Tax depreciation;
- CWIP;
- Investment tax credits;
- Expenses (fuel, O&M, etc.);
- AFUDC tax savings;
- Interest expenses;
- Revenue requirements and electric revenues; and
- Taxes.

Third, the preparation and output of financial statements and parameters are performed. This includes:

- Building up financial statements;
- Printing financial statements; and
- Calculating and printing financial parameters.

Figure 1
Central Algorithm for NUFS

For a standard analysis of a MEFS run, these basic steps are repeated for each region. Within each regional analysis, the steps are first performed for private utilities and then for public utilities within that region. The results for the private and public utilities are then combined to yield the results for the whole region.

The remainder of this volume is divided into two sections: (i) a description and listing of NUFS.FORT, followed by (ii) a description and listing of NUFS.RPT.FORT. In addition, this volume contains two appendices. Appendix A pertains to the software developed for generating the Form 1/Form 1M files which are described in Section 2 of Volume II. Appendix B pertains to the software developed for generating the MEFS interface files which are described in Section 3 of Volume II.

II. NUFS.FORT

NUFS.FORT consists of the main controlling routine as well as eleven other subroutines. These later subroutines, taken as a group, calculate the majority of the financial accounts. A brief description of each routine in NUF.S.FORT follows (listed in the order in which they appear in NUF.S.FORT) below.

<u>Routine</u>	<u>Description</u>
MAIN	- Controls overall flow of the program.
ASTVAL	- Calculates the book value of new assets.
BKDEP	- Calculates book depreciation (straight-line) for all assets.
TAXDEP	- Calculates tax depreciation, either sum of the years digits or declining balance method for all assets. Also calculates deferred taxes resulting from accelerated tax depreciation.
FLOCWP	- Calculates the flow into the construction work in progress account. Also calculates the amount of AFUDC generated and sales and property taxes.
ITC	- Calculates all accounts concerned with the investment tax credit.
EXPENS	- Calculates fuel costs, stock of nuclear fuel, O&M expenses, and working capital.
AFCINC	- Calculates deferrals from the tax savings due to the debt portion of AFUDC.
INTEXP	- Calculates the embedded costs of preferred stock and long term debt over time. Also calculates interest expenses.
REVS	- Calculates revenue requirements and solves for electric revenues based on the regulatory lag.
TAXES	- Calculates and reconciles current income taxes, booked income taxes and statutory income taxes.
STMTS	- Prepares financial statements--income, balance sheet, and sources and uses.

It should be noted that the first 150 lines of NUF.S.FORT are comments which contain useful information designed to explain the logic of the code. A listing of the code now follows.

```
1. //MI2UNUFS JOB (6376,D02,,15),'COMPILE',TIME=(0,08)
2. //DELD EXEC PGM=IEFBRI4,REGION=10K
3. //DD1 DD DSN=CN6376.MI2.NUFS.LOAD,DISP=(MOD,DELETE),
4. // UNIT=DASD,SPACE=(TRK,(12,1))
5. //FPC EXEC FORTGCL
6. //FORT.SYSIN DD *
7. C
8. C NUFS.FORT. THIS IS THE FIRST OF TWO SOURCE MODULES FOR
9. C THE NATIONAL UTILITY FINANCIAL STATEMENT (NUFS) MODEL. IT CONTAINS THE
10. C MAIN ROUTINE AND THE ACCOUNTING SUBROUTINES. (SEE BELOW)
11. C WRITTEN BY MARK R. INGLIS, ICF INC. OCTOBER 31, 1981
12. C
13. C THIS IS THE MAIN ROUTINE OF THE PROGRAM. IT CONTROLS THE FLOW
14. C AND CALLS THE APPROPRIATE SUBROUTINES.
15. C
16. C THERE ARE CURRENTLY EIGHT OPTIONS. THEY ARE:
17. C
18. C
19. C OPT1 = 1 CALCULATE REVENUE REQUIREMENT BASED ON YEAR END RATE BASE
20. C 2 CALCULATE REVENUE REQUIREMENT BASED ON AVERAGE RATE BASE
21. C
22. C OPT2 = 1 PLANT BUILD SPECIFIC CWIP RATEMAKING TREATMENT ARE INPUT
23. C 2 ALL BUILDS ARE ASSUMED TO HAVE THE SAME CWIP
24. C RATEMAKING TREATMENT.
25. C
26. C OPT3 = 1 NO REGULATORY LAG
27. C 2 ONE YEAR LAG
28. C 3 1 1/2 YEARS LAG
29. C 4 2 YEARS LAG
30. C
31. C OPT4 = 1 SUM-OF-THE-YEARS TAX DEPRECIATION
32. C = 2 DECLINING BALANCE TAX DEPRECIATION
33. C
34. C OPT5 = 1 KEEP CAPITAL STRUCTURE AS INPUT
35. C 2 DO NOT ALLOW LONG TERM DEBT, COMMON AND PREFERRED STOCK
36. C TO BE LOWERED; BUMP UP WORKING CAPITAL INSTEAD
37. C
38. C OPT6 = 1 ONE UTILITY - LINK WITH ICF SUPPLY MODEL
39. C 2 MULTI UTILITIES - LINK WITH MEFS/NCM
40. C
41. C OPT7 = 1 PRIVATES ONLY
42. C 2 PUBLICS ONLY
43. C 3 ANALYZE BOTH PRIVATES & PUBLICS, PRINT ALL RESULTS
44. C 4 ANALYZE BOTH PRIVATES & PUBLICS ,PRINT ONLY REGIONAL RESULTS
45. C 5 OUTPUT ONLY SHORT REPORT
46. C
47. C OPT8 = 1 STANDARD ANALYSIS OF MEFS RUN (OR ICF SUPPLY RUN)
48. C 2 CALCULATE FIXED CHARGE RATE FOR 1 PLANT TYPE
49. C 3 CALCULATE PRICING COEFFICIENT FOR 1 PLANT TYPE
50. C 4 CALCULATE REVENUE REQUIREMENT FOR OLD MONEY
51. C
52. C EXISTING PLANTS SHOULD BE PUT INTO A1(10) LIKE:
53. C
54. C 1 = INTANGIBLE
55. C 2 = STEAM PRODUCTION
```

56. C 3 = NUCLEAR PRODUCTION
57. C 4 = HYDRAULIC PRODUCTION - CONVENTIONAL
58. C 5 = HYDRAULIC PRODUCTION - PUMPED STORAGE
59. C 6 = OTHER PRODUCTION
60. C 7 = TRANSMISSION
61. C 8 = DISTRIBUTION
62. C 9 = GENERAL
63. C 10 = COMMON - ELECTRIC
64. C
65. C NEW PLANTS SHOULD BE PUT INTO A2(NP) LIKE:
66. C
67. C 1 = GAS STEAM
68. C 2 = OIL STEAM
69. C 3 = COAL TYPE 1
70. C 4 = COAL TYPE 2 (DIFFERENT FROM TYPE 1 BECAUSE OF CONST. PROFILE
71. C 5 = PUMPED STORAGE
72. C 6 = HYDRO
73. C 7 = NUCLEAR
74. C 8 = COMBINED CYCLE
75. C 9 = TURBINE
76. C 10 = RETROFITS W/ 3 YEAR CONSTRUCTION SCHEDULE (CREMOD)
77. C 11 = CONVERSIONS AND RETROFITS W/ 1YEAR CONST. SCHEDULE
78. C 12 = OTHER TECHNOLOGIES (SOLAR,GEOTHERMAL,ETC.)
79. C 13 = TRANSMISSION
80. C 14 = DISTRIBUTION
81. C 15 = METERS
82. C 16 = OTHER
83. C
84. C THERE ARE CURRENTLY TWENTY-THREE SUBROUTINES IN THE FINANCIAL
85. C MODEL. THEY CAN PROBABLY BEST BE SEPARATED INTO THREE CATE-
86. C GORIES: 1) PREPARING DATA; 2) THE ACCOUNTING ALGORITHMS; AND
87. C 3) CREATING AND REPORTING THE STATEMENTS AND OTHER FINANCIAL
88. C RATIOS.
89. C
90. C I. PREPARING DATA (LISTED IN ALPHABETICAL ORDER, THESE SUBROUTINES
91. C MAY BE FOUND IN THE SECOND HALF OF NUFS.RPT.FORT.)
92. C A. COMENT - THIS SUBROUTINE STRIPS COMMENTS OUT OF COMMENTED
93. C INPUT FILES.
94. C B. FORM1 - READS IN FILES WHICH HAVE BEEN CREATED BY NUFS.FORM1.FORT
95. C THIS DATA IS HISTORICAL FINANCIAL DATA FOR PRIVATE
96. C UTILITIES FOUND IN THE FPC FORM1.
97. C C. FORM1M - READS IN FILES WHICH HAVE BEEN CREATED BY
98. C NUFS.FORM1M.FORT. THIS DATA IS HISTORICAL FINANCIAL
99. C DATA FOR PUBLIC UTILITIES FOUND IN THE FPC FORM1M
100. C D. MEFS - READS IN FILES WHICH HAVE BEEN CREATED BY THE MEFS
101. C TO NUFS INTERFACE (NUFS.MEFS.FORT). THIS INCLUDES
102. C FORECASTS OF BUILDS,GENERATION, AND FUEL COST. ALSO
103. C WILL MODIFY THESE FORECASTS AS NECESSARY IN ORDER TO
104. C CALCULATE FIXED CHARGE RATES OR PRICING COEFFICIENTS
105. C E. READ - READS IN FILES PREPARED BY USER CONTAINING SUCH ASSUM-
106. C TIONS AS TREATMENT OF: CWIP,DEFERRED TAXES, AND DE-
107. C PRECIATION.
108. C F. SUPPLY - READS IN A DATA FILE WHICH HAS BEEN CREATED BY THE
109. C ICF ELECTRIC UTILITY SUPPLY MODEL. ALSO SETS UP FIX
110. C CHARGE RATE RUN, IF NECESSARY. THIS SUBROUTINE
111. C IS ONLY USED WHEN RUNNING NUFS FOR A SINGLE UTILITY.

```

112. C (IE., NEVER USED FOR MEFS APPLICATIONS)
113. C G. ZEROS1 - SETS VARIABLES EQUAL TO ZERO. IF IS CALLED FOR EACH
114. C REGION/OWNERSHIP COMBINATION ANALYZED WITHIN THE RUN
115. C H. ZEROS2 - SETS VARIABLES EQUAL TO ZERO. IT IS CALLED ONCE FOR
116. C REGION ANALYZED IN THE RUN.
117. C II. ACCOUNTING ALGORITHMS (LISTED IN ORDER CALLED BY MAIN, THESE SUB-
118. C ROUTINES MAY BE FOUND FOLLOWING THE MAIN ROUTINE IN NUFS.FORT.)
119. C A. ASTVAL - CALCULATES THE VALUE OF NEW ASSETS;
120. C B. BKDEP - CALCULATES BOOK DEPRECIATION (STRAIGHT-LINE) FOR
121. C ALL ASSETS.
122. C C. TAXDEP - CALCULATES TAX DEPRECIATION, EITHER SYD OR DE-
123. C CLINING BALANCE FOR ALL ASSETS. WILL THEN
124. C CALCULATE DEFERRED TAXES FROM ADR.
125. C D. FLOCWP - CALCULATE THE FLOW INTO THE CONSTRUCTION WORK
126. C IN PROGRESS ACCOUNT. ALSO, THE AMOUNT OF AFUDC
127. C GENERATED AND SALES AND PROPERTY TAXES.
128. C E. ITC - CALCULATES ALL ACCOUNTS CONCERNED WITH THE
129. C INVESTMENT TAX CREDIT.
130. C F. EXPENS - CALCULATE SALES, FUEL COSTS, AND O&M EXPENSES,
131. C AND WORKING CAPITAL.
132. C G. AFCINC - CALCULATES THE TAX DEFERRALS FROM AFUDC
133. C H. INTXP - CALCULATES INTEREST EXPENSE
134. C I. REVS - CALCULATES REVENUE REQUIREMENTS AND SOLVES FOR ELEC-
135. C TRIC REVENUES BASED ON THE REGULATORY LAG.
136. C J. TAXES - CALCULATES TAXES FOR FINANCIAL PURPOSES.
137. C K. STMTS - PREPARES ALL FINANCIAL STATEMENTS -- INCOME,
138. C BALANCE SHEET, AND SOURCES AND USES.
139. C
140. C III. CREATING AND REPORTING STATEMENTS (LISTED IN ORDER CALLED BY MAIN
141. C THESE SUBROUTINES MAY BE FOUND IN THE FIRST HALF OF NUFS.RPT.FORT
142. C A. REPORT - PRINTS ALL FINANCIAL STATEMENTS CALCULATED IN
143. C STMTS PLUS THE TAX RECONCILIATION.
144. C B. REPT1 - CALCULATES AND PRINTS A NUMBER OF FINANCIAL
145. C RATIOS.
146. C C. REPT2 - CALCULATES AND PRINTS EITHER THE FIXED CHARGE RATE
147. C OR PRICING COEFFICIENTS, WHEN NECESSARY.
148. C D. ADDUP - TAKES PRIVATE AND PUBLIC RESULTS AND ADDS TO
149. C GET TOTALS.
150. C
151. C VARIABLES ENDING IN 1 ARE EXISTING ASSETS AS OF
152. C 12/31/YEAR BEFORE FORECAST.
153. C
154. C VARIABLES ENDING IN 2 ARE NEW ASSETS BUILT OR
155. C ADDED DURING THE FORECAST PERIOD.
156. C
157. C COMMON A1(15),TDEW01(50),DR1(15),
158. C 1 TDEW02(50),A2(50,20),SY2(20),
159. C 2 XNUCFL(50),TDEW0A(50),XNSHRS(50),P(50),
160. C 3 BA1(15),TXSY2(20),
161. C 4 ABDE(50),BA2(50,20),
162. C 5 BASTS2(50,20),BDE(50),
163. C 6 CWPFAC(300),
164. C 7 DR2(20),
165. C 8 NAME1(6),NAME2(6),
166. C 1 T(15),SY1(15),BRTIRE(50),
167. C 1 PRODEF(50),

```

```

168.      2  DEPFAC(20),SALPR1(15),SALPR2(20),TXSY1(15)
169.      COMMON CWIP(80),CONPRO(15,20),BCWIP(80),
170.      1  YFCWIP(80,20),DIVRAT(50),PERLT(50),PERST(50),PERCE(50),
171.      2  PERPS(50),RTNWLT(50),RTST(50),RTCEQ(50),RTNWPS(50),
172.      3  RBCWIP(80),OFFSET(80)
173.      COMMON TDEREG(50),
174.      3  DITC(50),
175.      4  WC(50),UP(50),AMD(50),OLDPS(50),EMPS(50),
176.      5  OLDLT(50),EMDT(50),XINEX(50),XSINEX(50),
177.      6  XLINEX(50),
178.      7  GINF(50),GINFX(70),
179.      8  TXFUEL(50),
180.      9  GENTX(50),IYRS(50),FUEL(50),PROPTX(50),
181.      *  SALETX(50),
182.      1  AFUDCE(80),AFUDCD(80),EXTBF(50),EXTBD(50),XITCF(50),XITCD(50),
183.      2  AITC(50),ADJBDE(50),ADJITC(50),ADJADC(50),TTXEX(50),STATX(50),
184.      3  STATET(50),CIT(50),WACD(50),
185.      5  AAFDC(50),XAFDCD(50),XAFDCF(50),DAFDC(50),
186.      6  COMDIV(50)
187.      COMMON XINCST(3,50,18),
188.      2  BALSHT(3,50,23),REVREQ(3,50,23),FUNDS(3,50,19)
189.      COMMON REVLAG(3,50,5),AFUDC(3,50),TOTSAL(3,50),
190.      2  PLTCAP(3,300),
191.      3  GWHTOT(3,50),GWHNUC(3,50),JYR(300),JPLANT(300),
192.      4  CAPCST(300),AVFLCS(50),FCNUC(50),OM(50),AVGOM(50),
193.      5  SERPER(300),CWPPER(300),RBAFDC(300),BLDTOT(3)
194.      DIMENSION NAME4(6),NAME3(6),IPRIYR(3)
195.      DATA NAME4/'PUBL','IC &','PRI','VATE',' ',' ' /
196.      DATA NPTYPE/16/
197.  C
198.      DIMENSION Y1(4017),Y2(2670),Y3(2080),
199.      2  Y4(12450),Y5(4253),AFCPER(50),ROE(50),COVRAT(50)
200.  C
201.      EQUIVALENCE (Y1(1),A1(1)),(Y2(1),CWIP(1)),
202.      1  (Y3(1),TDEREG(1)),
203.      5  (Y4(1),XINCST(1)),(Y5(1),REVLAG(1))
204.      WRITE(4,9000)
205.  9000  FORMAT(' EXECUTION BEGINNING')
206.  C
207.  C*****
208.  C  FIRST, INPUT VALUES WHICH DEFINE SCOPE OF THIS PARTICULAR RUN:
209.  C      NAME3 - PAGE HEADING FOR MODEL OUTPUT.
210.  C      NYRS - TOTAL NUMBER OF YEARS TO BE FORECAST. INTERNALLY,
211.  C      THE YEARS GO FROM 2-NYRS+1 ALTHOUGH NYRS (NOT
212.  C      NYRS+1) IS INPUT.
213.  C      NRGIUN - NUMBER OF REGIONS TO BE ANALYZED IN THIS RUN
214.  C      OPT7 - 1 ANALYZE PRIVATES ONLY
215.  C      2 ANALYZE PUBLICS ONLY
216.  C      3 ANALYZE BOTH PRIVATES & PUBLICS, PRINT ALL RESULTS
217.  C      4 ANALYZE BOTH PRIVATES & PUBLICS ,PRINT ONLY REGIONAL
218.  C      5 OUTPUT ONLY SHORT REPORT
219.  C      OPT8 - 1 STANDARD ANALYSIS OF MEFS RUN (OR ICF SUPPLY RUN)
220.  C      2 CALCULATE FIXED CHARGE RATE FOR 1 PLANT TYPE
221.  C      3 CALCULATE PRICING COEFFICIENT FOR 1 PLANT TYPE
222.  C      4 CALCULATE REVENUE REQUIREMENT FOR OLD MONEY
223.  C*****

```

```

224.      CALL COMENT(9)
225.      READ(9,101) NAME3
226. 101   FORMAT(6A4)
227.      CALL COMENT(9)
228.      READ(9,*) NYRS,NRGION,OPT7,OPT8
229.      WRITE(4,102) NAME3,NYRS,NRGION
230. 102   FORMAT(' RUN:',6A4,' NUMBER OF YEARS:',I2,' NUM OF REGIONS:',I2)
231. C*****
232. C   REGION/OWNERSHIP COMBINATIONS ARE ANALYZED INDEPENDENTLY AND SUCCESS
233. C   ITYPE IS A VARIABLE WHOSE CURRENT VALUE INDICATES WHICH OWNERSHIP IS
234. C   CURRENTLY BEING ANALYZED BY NUF. IE.,
235. C       ITYPE - CURRENTLY ANALYZED OWNERSHIP TYPE
236. C           1 = PRIVATE
237. C           2 = PUBLIC
238. C           3 = BOTH PRIVATE & PUBLIC
239. C*****
240.      IF (OPT7.EQ.1.0) ITYPE=1
241.      IF (OPT7.EQ.2.0) ITYPE=2
242.      ITER=3
243.      IF (OPT7.LT.3) ITER=1
244.      NYRS=NYRS+1
245.      DO 8000 NIT=1,NRGION
246.      CALL ZEROS2(Y4,Y5)
247.      DO 7000 ICOUNT=1,ITER
248.      IF (OPT7.LT.3.0) GO TO 7006
249.      ITYPE=ICOUNT
250.      IF (ITYPE.EQ.3) GO TO 685
251. 7006   NN=10
252.      IF (ITYPE.EQ.2) NN=11
253.      NNN=12
254.      IF (ITYPE.EQ.1) NNN=13
255.      NNNN=14
256. C
257.      CALL ZEROS1(Y1,Y2,Y3)
258. C
259. C*****
260. C*
261. C*   CALL ALL THE SUBROUTINES
262. C*
263. C*****
264. C
265.      CALL READ(NN,NAME1,NAME2,OPT1,OPT2,OPT3,
266. 1 OPT4,OPT5,OPT6,NPTYPE,TXRT,STXRT,
267. 2 PRTXRT,SATXRT,RITC,DISCR,IYRS,AFDCR,
268. 4 P,XNSHRS,FLPR1,FLPR2,DIVRAT,
269. 5 PERLT,PERST,PERCE,PERPS,RTNWLT,RTST,RTCEQ,RTNWPS,GINF,GINFX,
270. 6 DR1,T,SY1,TSY1,SALPR1,
271. 7 SY2,TSY2,DR2,DEPFAC,SALPR2,
272. 8 CWPPER,RBAFDC,CONPRO,
273. 9 ITYPE,NYRS)
274. C
275.      IF (OPT7.GE.3.0 .AND. ITYPE.GT.1) GO TO 20
276.      IF (OPT6 .NE. 1) GO TO 10
277.      CALL SUPPLY(NNNN,NPLNTS,JYR,JPLANT,PLTCAP,CAPCST,
278. 1 NYRS,GWHTOT,GWHTNUC,AVFLCS,FCNUC,GINFX,AVGOM,SERPER,
279. 2 CWPPER,RBAFDC,TOTSAL,NIT,IFCRYR,IFCRBD,OPT2,OPT8)

```

```

280. 10      IF (OPT6 .NE. 2) GO TO 20
281.          CALL MEFS(NNNN,NPLNTS,JYR,JPLANT,PLTCAP,CAPCST,
282. 1          NYRS,GWHTOT,GWNUC,AVFLCS,FCNUC,
283. 2          AVGOM,TOTSAL,IYRS,GINFX,
284. 3          NIT,IFCRYR,IFCRBD,IPRIYR,IPRIBD,BLDTOT,OPT8)
285. 20      IF (ITYPE.NE.1) GO TO 30
286.          CALL FORM1(NNN,BA1,A1,EMDT,GTXT,OLDLT,REVLG,
287. 1          DITC,PRODEF,TFUEL,OLDPS,EMPS,
288. 2          BALSHT,WC,XINCST,TOTSAL,OPT8)
289. 30      IF (ITYPE.NE.2) GO TO 40
290.          CALL FORM1M(NNN,BA1,A1,EMDT,GTXT,OLDLT,REVLG,
291. 1          TFUEL,
292. 2          BALSHT,WC,XINCST,TOTSAL,OPT8)
293. 40      CALL ASTVAL(NYRS,PLTCAP,CWPFAC,CONPRO,AFDCR,A2,CAPCST,
294. 1          BA2,OPT2,SERPER,
295. 3          NPLNTS,JYR,JPLANT,CWPPER,
296. 5          RBAFDC,ITYPE)
297.          CALL BKDEP(NYRS,T,DR1,A1,BA1,TDEW01,
298. 1          ABDE,A2,SY2,TDEW02,JYR,JPLANT,NPLNTS,
299. 2          BA2,BRTIRE,BASTS2,
300. 3          TDEW0A,BDE,SERPER,SALPR1,SALPR2,SY1,DR2,NPTYPE)
301.          CALL TAXDEP(T,SY1,A1,TXT,NYRS,
302. 1          TSY2,A2,PRODEF,TDEW0A,TDEREG,JYR,JPLANT,NPLNTS,
303. 2          TDEW01,TDEW02,OPT4,DEPFAC,TSY1,FLPR1,EXTBD,EXTBF)
304.          CALL FLOCWP(NPTYPE,NYRS,A2,BA2,CONPRO,YFCWIP,
305. 1          AFDCR,CWIP,BCWIP,SALETX,SATXT,PROPTX,PRTXT,
306. 2          SERPER,SY2,AFUDCE,AFUDCD,AFUDC,
307. 3          WACD,PERLT,PERST,PERCE,PERPS,RTNWL,RTST,RTCEQ,
308. 4          RTNWPS,NPLNTS,JYR,JPLANT,PLTCAP,CWPPER,CAPCST,
309. 5          CWPFAC,RBCWIP,RBAFDC,OFFSET,
310. 6          ITYPE)
311.          CALL ITC(NYRS,CWIP,SY1,T,
312. 1          RITC,A2,SY2,YFCWIP,
313. 3          NPTYPE,DITC,ADJITC,AITC,XITCF,
314. 4          FLPR1,XITCD)
315.          CALL EXPENS(NYRS,AVFLCS,FCNUC,FUEL,GWHTOT,GWNUC,
316. 1          XNUCFL,
317. 3          OM,WC,TFUEL,GTXT,ITYPE,AVGOM)
318.          CALL AFCINC(NPTYPE,BA1,A1,WACD,TXT,T,SY1,
319. 1          NYRS,AFUDCD,SY2,BA2,A2,PRODEF,
320. 2          FLPR2,DAFDC,AAFDC,XAFDCF,XAFDCD)
321.          CALL INTEXP(NYRS,OLDLT,PERLT,BDE,AMD,UP,
322. 1          BCWIP,EMDT,RTNWL,XLINEX,XSINEX,RTST,PERST,XINEX,
323. 2          SY1,T,BA1,NPTYPE,BASTS2,WC,ABDE,XNUCFL,PRODEF,
324. 3          RTNWPS,PERPS,EMPS,OLDPS,DITC,OPT5,OPT8)
325.          CALL REVS(NYRS,AITC,XITCF,RBCWIP,WC,ABDE,PRODEF,
326. 1          TDEREG,TDEW0A,XAFDCF,AAFDC,PERLT,EMDT,PERST,RTST,
327. 2          PERCE,RTCEQ,PERPS,EMPS,OPT3,SERPER,
328. 3          NPTYPE,BA2,OM,UP,STXT,TXT,FUEL,XINEX,XNUCFL,
329. 4          PROPTX,SALETX,GTXT,AFUDCD,BDE,REVLG,TFUEL,
330. 5          OFFSET,TOTSAL,REVREQ,ITYPE,OPT1,OPT8)
331.          CALL TAXES(NYRS,GENTX,REVLG,GTXT,PROPTX,SALETX,STATX,FUEL,
332. 1          OM,BDE,TDEW0A,XINEX,AFUDCE,TXT,ADJADC,STATET,STXT,
333. 2          TTXEX,CIT,AAFDC,XAFDCF,XAFDCD,AFUDC,
334. 3          ITYPE,ADJBDE,ADJITC,AITC,XITCF,EXTBF,EXTBD)
335.          CALL STMTS(NYRS,XINCST,REVLG,OM,FUEL,BDE,GENTX,TTXEX,

```



```

336.      1 AFUDCE,XLINEX,XSINEX,AFUDCD,BALSHT,CWIP,BRTIRE,
337.      2 BA1,BASTS2,BCWIP,ABDE,XNUCFL,WC,AMD,PRODEF,
338.      3 PERLT,PERST,PERCE,PERPS,EMPS,COMDIV,DIVRAT,
339.      4 XINEX,CIT,AFUDC,FUNDS,SY1,T,
340.      5 NPTYPE,ITYPE,DITC,OPT5)
341.      IF (OPT7 .GE. 4) GO TO 675
342.      CALL REPORT(IYRS,XINCST,BALSHT,CIT,
343.      1 TTXEX,STATX,ADJBDE,ADJADC,STATET,AAFDC,XAFDCF,XAFDCD,
344.      2 REVREQ,NYRS,FUNDS,NAME1,NAME2,NAME3,ITYPE,ADJITC,
345.      3 AITC,XITCF,EXTBF,EXTBD,XITCD,OPT1)
346. 675   IF (OPT7 .EQ. 4) GO TO 7000
347.      CALL REPRT1(NYRS,REVLAG,DISCR,TOTSAL,
348.      1 BALSHT,XINCST,IYRS,COVRAT,AFCPER,ROE,
349.      2 AFUDC,COMDIV,NAME1,XNSHRS,P,
350.      3 NAME2,NAME3,ITYPE,RTCEQ,OPT8)
351.      IF (OPT8.NE.1 .AND. OPT7.LT.3) CALL REPRT2(NYRS,NAME1,NAME2,NAME3,
352.      1 DISCR,IYRS,REVLAG,OPT8,GINF,GINFX,JYR,JPLANT,PLTCAP,CAPCST,SY2,
353.      2 IPRIYR,BLDTOT,ITYPE)
354.      GO TO 7000
355. 685   CALL ADDUP(XINCST,BALSHT,FUNDS,REVREQ,REVLAG,AFUDC,NYRS,OPT8)
356.      DO 690 J=1,6
357. 690   NAME2(J)=NAME4(J)
358.      IF (OPT7 .EQ. 5) GO TO 695
359.      CALL REPORT(IYRS,XINCST,BALSHT,CIT,
360.      1 TTXEX,STATX,ADJBDE,ADJADC,STATET,AAFDC,XAFDCF,XAFDCD,
361.      2 REVREQ,NYRS,FUNDS,NAME1,NAME2,NAME3,ITYPE,ADJITC,
362.      3 AITC,XITCF,EXTBF,EXTBD,XITCD,OPT1)
363. 695   CALL REPRT1(NYRS,REVLAG,DISCR,TOTSAL,
364.      1 BALSHT,XINCST,IYRS,COVRAT,AFCPER,ROE,
365.      2 AFUDC,COMDIV,NAME1,XNSHRS,P,
366.      3 NAME2,NAME3,ITYPE,RTCEQ,OPT8)
367.      IF (OPT8.NE.1) CALL REPRT2(NYRS,NAME1,NAME2,NAME3,DISCR,IYRS,
368.      1 REVLAG,OPT8,GINF,GINFX,JYR,JPLANT,PLTCAP,CAPCST,SY2,IPRIYR,
369.      2 BLDTOT,ITYPE)
370. 7000  CONTINUE
371.      CALL SHTRPT(NYRS,IYRS,NAME1,NAME3,AFCPER,ROE,COVRAT,FUNDS,ITYPE)
372. 8000  CONTINUE
373.      STOP
374.      END
375.      SUBROUTINE ASTVAL(NYRS,PLTCAP,CWPFAC,CONPRO,AFDCR,A2,CAPCST,
376.      1 BA2,OPT2,SERPER,
377.      3 NPLNTS,JYR,JPLANT,CWPPER,RBAFDC,ITYPE)
378.      C
379.      C*****
380.      C*
381.      C*      SUBROUTINE ASTVAL
382.      C*      THIS SUBROUTINE CALCULATES THE BOOK VALUE OF NEW ASSETS*
383.      C*
384.      C*      VARIABLES IN SUBROUTINE ASTVAL
385.      C*
386.      C*      BA2 = BOOK VALUE OF NEW ASSETS
387.      C*      A2 = BA2, NET OF CAPITALIZED AFUDC
388.      C*      SERPER = % OF YEAR NEW PLANT IN-SERVICE FOR 1ST YEAR
389.      C*      CWPFAC = DIVIDE INTO COST TO GET VALUE OF NEW PLANT W/O
390.      C*      AFUDC
391.      C*

```

```

392. C*****
393. C
394.     DIMENSION PLTCAP(3,300),CWPFAC(300),CONPRO(15,20),A2(50,20),
395.     1 CAPCST(300),BA2(50,20),SERPER(300),
396.     4 JPLANT(300),JYR(300),CWPPER(300),
397.     6 RBAFDC(300)
398. C***FIRST THE BOOK VALUES OF THE NEW BUILDS ARE CALCULATED. THE
399. C***BOOK VALUE IS A FUNCTION OF CAPACITY,UNIT CAPACITY COST, AND THE
400. C***AFUDC RATE. THE CAPACITY AND CAPACITY COSTS ARE THOSE READ FROM
401. C***THE CAPACITY EXPANSION MODELS (MEFS,NCM,ICF SUPPLY,ETC.).
402.     WRITE(4,10)
403.     10 FORMAT(' BUILD #',5X,'YEAR',5X,'TYPE',5X,'COST (W/O ',
404.     * 'AFUDC)',5X,'BOOK VALUE',5X,'CWPFAC')
405.     DO 311 K=1,NPLNTS
406.     300 IF (OPT2.EQ. 1) GO TO 360
407.         SERPER(K)=SERPER(1)
408.         CWPPER(K)=CWPPER(1)
409.         RBAFDC(K)=RBAFDC(1)
410.     360 I=JYR(K)
411.         J=JPLANT(K)
412.         CWPFAC(K)=0
413.         DO 310 L=1,14
414.             CWPFAC(K)=CONPRO(16-L,J)*((AFDCR+1.0)**(14.5-L))+CWPFAC(K)
415.     310 CONTINUE
416.         CWPFAC(K)=CONPRO(1,J)*((AFDCR+1.0)**((1.0-SERPER(K))/2.0))
417.         * +CWPFAC(K)
418.     330 BA2(I,J)=(PLTCAP(ITYPE,K)*CAPCST(K)*(1-
419.     1 CWPPER(K)))+(PLTCAP(ITYPE,K)*CAPCST(K)*
420.     2 CWPPER(K))/CWPFAC(K))
421.         A2(I,J)=(PLTCAP(ITYPE,K)*CAPCST(K))/CWPFAC(K)
422.         WRITE(4,309) K,I,J,A2(I,J),BA2(I,J),CWPFAC(K)
423.     309 FORMAT(3X,I3,8X,I2,7X,I2,10X,F8.0,10X,F8.0,6X,F6.4)
424.     311 CONTINUE
425. C
426.     RETURN
427.     END
428.     SUBROUTINE BKDEP(NYRS,T,DR1,A1,BA1,TDEW01,
429.     1 ABDE,A2,SY2,TDEW02,JYR,JPLANT,NPLNTS,
430.     2 BA2,BRTIRE,
431.     3 BASTS2,TDEW0A,BDE,SERPER,SALPR1,SALPR2,SY1,DR2,
432.     4 NPTYPE)
433. C
434. C*****
435. C*
436. C* SUBROUTINE BKDEP
437. C* THIS SUBROUTINE CALCULATES BOOK DEPRECIATION AND
438. C* TAX DEPRECIATION WITHOUT ACCELERATION
439. C*
440. C* VARIABLES IN SUBROUTINE BKDEP
441. C* DE1,BDE2,BDE = BOOK DEPRECIATION EXPENSE *
442. C* TDEW01,TDEW02,TDEW0A = TAX DEPRECIATION W/O ACCELERATION
443. C* ABDE = TOTAL ACCUMULATED BOOK DEPRECIATION*
444. C* BASTS2 = BOOK VALUE OF NEW ASSETS AT END OF YEAR
445. C* BRTIRE = BOOK VALUE OF RETIREMENTS BY YEAR *
446. C*
447. C*****

```

```

448. C
449.     DIMENSION T(15),DR1(15),A1(15),
450.     1 BA1(15),TDEW01(50),
451.     2 BDE1(50),ABDE(50),A2(50,20),
452.     3 SY2(20),TDEW02(50),JYR(300),JPLANT(300),
453.     4 DR2(20),BA2(50,20),BDE2(50),
454.     5 BASTS2(50,20),TDEW0A(50),BDE(50),SERPER(300),SALPR1(15),
455.     6 SALPR2(20),SY1(15),BRTIRE(50)
456. C***INITIALIZE.
457.     DO 20 I=1,NYRS
458.     BDE1(I)=0.0
459. 20   BDE2(I)=0.0
460. C
461. C*****
462. C     ***PRE BASE YEAR ASSETS***
463. C     ACCUMULATED DEPRECIATION IN THE BASE YEAR IS CALCULATED FIRST.
464. C     DEPRECIATION IS CALCULATED IN THE SAME MANNER FOR THE CWIP CASE
465. C     AND THE AFUDC CASE. THE ONLY DIFFERENCE IS THE BOOK VALUE.
466. C*****
467. 537 DO 7 J=1,10
468.     ABDE(1)=(T(J))*DR1(J)*BA1(J)+ABDE(1)
469. 7   CONTINUE
470.     DO 5 J=1,10
471.     K=SY1(J)-T(J)+1
472.     K2=K+1
473.     IF (K.GT.NYRS) K=NYRS
474.     DO 540 KK=2,K
475.     TDEW01(KK)=DR1(J)*A1(J)+TDEW01(KK)
476.     BDE1(KK)=DR1(J)*BA1(J)+BDE1(KK)
477. 540 CONTINUE
478.     IF (K2 .GT. NYRS) GO TO 5
479.     TDEW01(K2)=TDEW01(K2)+SALPR1(J)*A1(J)
480.     BDE1(K2)=BDE1(K2)+SALPR1(J)*BA1(J)
481. C*** IF ASSET TYPE RETIRES IN FORECAST HORIZON, ADD ITS BOOK VALUE TO
482. C*** THE RETIREMENT ACCOUNT.
483.     WRITE(4,901) BA1(J),J,K2
484. 901 FORMAT(' ON ASSET RETIREMENT:',F10.1,2X,'TYPE:',I2,2X,'YR:',I2)
485.     BRTIRE(K2)=BRTIRE(K2)+BA1(J)
486. 5   CONTINUE
487. C*****
488. C     ***NEW ASSETS***
489. C*****
490.     DO 10 M=1,NPLNTS
491.     IF (JYR(M).GT.NYRS) GO TO 10
492.     I=JYR(M)
493.     J=JPLANT(M)
494.     TDEW02(I)=(A2(I,J)*DR2(J))*SERPER(M)+TDEW02(I)
495.     BDE2(I)=(BA2(I,J)*DR2(J))*SERPER(M)+BDE2(I)
496.     BASTS2(I,J)=BASTS2(I,J)+BA2(I,J)
497.     K2=I+SY2(J)
498.     IF (K2 .GT. NYRS) GO TO 910
499.     TDEW02(K2)=(A2(I,J)*DR2(J))*(1.0-SERPER(M))+TDEW02(K2)+
500.     1     SALPR2(J)*A2(I,J)
501.     BDE2(K2)=(BA2(I,J)*DR2(J))*(1.0-SERPER(M))+BDE2(K2)+
502.     1     SALPR2(J)*BA2(I,J)
503. 910 K1=I+1

```

```

504.          K=I+SY2(J)-1
505.          IF (K.GT.NYRS) K=NYRS
506.          DO 920 KK=K1,K
507.             TDEW02(KK)=A2(I,J)*DR2(J)+TDEW02(KK)
508.             BDE2(KK)=BA2(I,J)*DR2(J)+BDE2(KK)
509.             BASTS2(KK,J)=BA2(I,J)+BASTS2(KK,J)
510. 920      CONTINUE
511. C*** IF BUILD IS GOING TO RETIRE IN FORECAST HORIZON, THEN ADD ITS BOOK
512. C*** VALUE TO THE RETIREMENT ACCOUNT.
513.          IF (K2 .GT. NYRS) GO TO 10
514.          WRITE(4,900) BA2(I,J),J,K2
515. 900      FORMAT(' NEW ASSET RETIREMENT:',F10.1,' TYPE:',I2,' YR:',I2)
516.          BRTIRE(K2)=BRTIRE(K2)+BA2(I,J)
517. 10      CONTINUE
518. C*****
519. C*** ADD OLD AND NEW ASSET ACCOUNTS.
520. C*****
521.          DO 15 I=2,NYRS
522.             TDEW0A(I)=TDEW01(I)+TDEW02(I)
523.             BDE(I)=BDE1(I)+BDE2(I)
524.             ABDE(I)=ABDE(I-1)+BDE(I)
525. 15      CONTINUE
526. C*****
527. C*** NOW ADJUST ACCUMULATED ACCOUNT TOTALS FOR RETIREMENTS.
528. C*****
529.          DO 200 I=2,NYRS
530.             IF (BRTIRE(I) .EQ. 0.0) GO TO 200
531.             DO 210 J=I,NYRS
532.                ABDE(J)=ABDE(J)-BRTIRE(I)
533. 210      CONTINUE
534. 200      CONTINUE
535. C*** WRITE OUT NEW ASSETS FOR DEBUGGING.
536.          WRITE(4,300)
537. 300      FORMAT(' ',36X,'TOTAL CUMULATIVE NEW ASSETS BY YEAR (ROWS) AND',
538.          * 'BY TYPE (COLS)')
539.          DO 310 I=2,NYRS
540. 310      WRITE(4,320) I,(BASTS2(I,J),J=1,NPTYPE)
541. 320      FORMAT(' ',I2,1X,16(F7.0,1X))
542.          RETURN
543.          END
544.          SUBROUTINE TAXDEP(T,SY1,A1,TXRT,NYRS,
545.          1  TXY2,A2,PRODEF,TDEW0A,TDEREG,JYR,JPLANT,NPLNTS,
546.          2  TDEW01,TDEW02,OPT4,DEPFAC,TXSY1,FLPR1,EXTBD,EXTBF)
547. C
548. C*****
549. C*
550. C*          SUBROUTINE TAXDEP
551. C*          THIS SUBROUTINE CALCULATES TAX DEPRECIATION
552. C*          USING EITHER THE SUM-OF-THE-YEARS OR DECLINING
553. C*          BALANCE METHOD.
554. C*
555. C*          VARIABLES IN SUBROUTINE TAXDEP
556. C*
557. C*          T1 = CURRENT YEAR OF SERVICE FOR EXISTING ASSET
558. C*          TD1 = DEPRECIATION RATE FOR SYD
559. C*          TDE = TOTAL TAX DEPRECIATION

```

```

560. C*      PRODEF = TOTAL DEFERRED FEDERAL INCOME TAXES      *
561. C*      EXTBD = TAX SAVINGS RESULTING FROM ACCELERATED DE- *
562. C*      PRECIATION WHICH ARE DEFERRED.                    *
563. C*      EXTBF = TAX SAVINGS RESULTING FROM ACCELERATED DE- *
564. C*      PRECIATION WHICH ARE FLOWED THROUGH.              *
565. C*      TDEREG = DEPRECIATION USED IN CALCULATING FED INCOME TAXES*
566. C*      FOR REGULATORY PURPOSES
567. C*                                                         *
568. C*****
569. C
570.      DIMENSION T(15),TD1(50,15),SY1(15),A1(15),TDE1(50),
571.      1 TXSY2(20),JYR(300),JPLANT(300),
572.      2 TDE2(50),A2(50,20),TDE(50),
573.      3 PRODEF(50),TDEW0A(50),TDEW01(50),TDEREG(50),
574.      4 TDEW02(50),DEPFAC(20),TXSY1(15),EXTBD(50),EXTBF(50)
575. C***INITIALIZE
576.      DO 10 I=1,NYRS
577.      DO 20 J=1,10
578.      20 TD1(I,J)=0.0
579.      TDE1(I)=0.0
580.      TDE2(I)=0.0
581.      TDE(I)=0.0
582.      10 CONTINUE
583. C
584. C*****
585. C      ***OLD ASSETS***
586. C*****
587. C***EXISTING ASSETS ARE ASSUMED TO BE DEPRECIATED USING SYD METHOD.
588. C***FIRST WE CALCULATE THE TAX DEPRECIATION RATE, THEN THE ACTUAL DEPRE-
589. C***CIATION AND DEFERED TAXES FOR EXISTING ASSETS.
590.      DO 43 J=1,10
591.      T1=T(J)
592.      DO 40 I=2,NYRS
593.      TD1(I,J)=(TXSY1(J)-(T1-1))/(TXSY1(J)*(TXSY1(J)+1)/2)
594.      IF (TD1(I,J).LE.0.0) TD1(I,J)=0
595.      T1=T1+1
596.      TDE1(I)=TDE1(I)+TD1(I,J)*A1(I,J)
597.      40 CONTINUE
598.      43 CONTINUE
599. C***PRODEF(1) IS INPUT FROM THE FORM1. IT IS THE DEFERRED TAXES FOR BAS
600. C***YEAR ASSETS. PRDEF1 IS USED TO 'TRACK' THE VALUE OF THIS DEFERRAL A
601. C***SHOULD NEVER BECOME NEGATIVE. BUT, BECAUSE OF BAD INPUTS, THIS MAY
602. C***HAPPEN. IF IT DOES, IT IS TAKEN CARE OF.
603.      PRDEF1=PRODEF(1)
604.      DO 45 I=2,NYRS
605.      EXTBD(I)=TXRT*(TDE1(I)-TDEW01(I))*(1-FLPR1)
606.      EXTBF(I)=TXRT*(TDE1(I)-TDEW01(I))*(FLPR1)
607.      PRDEF1=PRDEF1+EXTBD(I)
608.      IF (PRDEF1 .GT. 0.0) GO TO 45
609.      EXTBD(I)=EXTBD(I)-PRDEF1
610.      GO TO 60
611.      45 CONTINUE
612. C*****
613. C      ***NEW ASSETS***
614. C*****
615. C***NEW ASSETS CAN BE DEPRECIATED USING SYD OR DECLINING BALANCE.

```

```

616. C***FIRST THE TAX DEPRECIATION IS CALCULATED, THEN THE DERERED ACCOUNTS
617. C***CAN BE CALCULATED.
618. 60 DO 52 M=1,NPLNTS
619.     IF (JYR(M).GT.NYRS) GO TO 52
620.     I=JYR(M)
621.     J=JPLANT(M)
622.     K=I+TXSY2(J)-1
623.     IF (K.GT.NYRS) K=NYRS
624.     IF (OPT4.EQ.2.0) GO TO 491
625. C***THIS SECTION CALCULATES TAX DEPRECIATION USING THE SYD METHOD.
626.     L=0
627.     DO 50 KK=I,K
628.         TDE2(KK)=(TXSY2(J)-L)/(TXSY2(J)*(TXSY2(J)+1)/2)
629.         1      *A2(I,J)+TDE2(KK)
630.         L=L+1
631.     50 CONTINUE
632.     GO TO 52
633. C***THIS SECTION CALCULATES TAX DEP USING THE DECLINING BALANCE METHOD.
634. 491 VALUE=A2(I,J)
635.     COUNT=0
636.     XAD2=0
637.     DEPRAT=DEPFAC(J)/TXSY2(J)
638.     DO 403 KK=I,K
639.         TXDEP=DEPRAT*(VALUE-XAD2)
640.         TEST=(VALUE-XAD2)/(TXSY2(J)-COUNT)
641.         COUNT=COUNT+1
642.         IF (TXDEP.GT.TEST) GO TO 402
643.         TXDEP=TEST
644.     402 XAD2=XAD2+TXDEP
645.         TDE2(KK)=TXDEP+TDE2(KK)
646.     403 CONTINUE
647. 52 CONTINUE
648.     DO 100 I=2,NYRS
649.         TDE(I)=TDE1(I)+TDE2(I)
650.         EXTBFI=(TXRT*(TDE2(I)-TDEWO2(I)))*(FLPR1)
651.         1      +EXTBFI(I)
652.         EXTBDI=(TXRT*(TDE2(I)-TDEWO2(I)))*(1.0-FLPR1)
653.         1      +EXTBDI(I)
654.         PRODEF(I)=PRODEF(I-1)+EXTBDI(I)
655.         TDEREG(I)=(TDEWOA(I)*(1.0-FLPR1))+(TDE(I)*FLPR1)
656.     100 CONTINUE
657.     RETURN
658.     END
659.     SUBROUTINE FLOCWP(NPTYPE,NYRS,A2,BA2,CONPRO,YFCWIP,
660.     1 AFDIC,CWIP,BCWIP,SALETX,SATXRT,PROPTX,PTXRT,SERPER,
661.     2 SY2,AFULCE,AFUDCD,AFUDC,
662.     3 WACD,PERLT,PERST,PERCE,PERPS,RTNWLT,RTST,RTCEQ,RTNWPS,
663.     4 NPLNTS,JYR,JPLANT,PLTCAP,CWPPER,CAPCST,
664.     5 CWPFAC,RBCWIP,RBAFDC,OFFSET,
665.     6 ITYPE)
666. C
667. C*****
668. C*
669. C* SUBROUTINE FLOCWP
670. C* THIS SUBROUTINE CALCULATES THE FLOW INTO
671. C* CWIP ACCOUNTS

```

```

672. C*
673. C*   VARIABLES IN SUBROUTINE FLOCWP
674. C*
675. C*   WACD,WACD15 = WEIGHTED AVERAGE COST OF DEBT
676. C*   WACE15 = WEIGHTED AVERAGE COST OF EQUITY IN YEAR I-15
677. C*   YFCWIP = ANNUAL FLOW INTO CWIP ACCOUNT BY PLANT TYPE
678. C*   SCWIP = CUMULATIVE FLOW INTO CWIP BY PLANT
679. C*   RBCWP2 = AMOUNT OF CWIP IN RATE BASE BY PLANT TYPE
680. C*   BSCWIP = CUMULATIVE FLOW INTO CWIP W/AFUDC BY PLANT
681. C*   BCWIP2 = CUMULATIVE FLOW INTO CWIP W/AFUDC BY PLANT TYPE
682. C*   AFUDCE = AFUDC EQUITY
683. C*   AFUDCD = AFUDC DEBT
684. C*   BCWIP = BOOKED CWIP BY YEAR
685. C*   CWIP = CUMULATIVE CONSTRUCTION EXPENSES, NET OF AFUDC
686. C*   RBCWIP = AMOUNT OF CWIP IN RATE BASE BY YEAR
687. C*   SALETX = SALES TAX CALCULATES ON YEARLY CWIP EXPENDITURES*
688. C*   PROPTX = PROPERTY TAX CALCULATED ON VALUE OF ASSETS
689. C*   AFUDC = TOTAL YEARLY AFUDC
690. C*   OFFSET = AFUDC OFFSET
691. C*****
692. C
693.   DIMENSION A2(50,20),BA2(50,20),
694.   1 FWCIP(80),CONPRO(15,20),YFCWIP(80,20),BCWIP2(80,20),
695.   2 CWIP(80),BCWIP(80),SALETX(50),PROPTX(50),SERPER(300),
696.   3 SY2(20),BSCWIP(80),AFUDCE(80),AFUDCD(80),AFUDC(3,50),
697.   4 WACE15(80),WACD(50),PERLT(50),WACD15(80),
698.   5 PERST(50),PERCE(50),PERPS(50),RTNWLT(50),RTST(50),RTCEQ(50),
699.   6 RTNWPS(50),JYR(300),JPLANT(300),
700.   7 PLTCAP(3,300),CWPPER(300),CAPCST(300),CWPFAC(300),
701.   8 RBCWP2(80,20),RBCWIP(80),
702.   9 RBAFDC(300),OFFSET(80),SCWIP(80)
703. C***INITIALIZE.
704.   NYRS30=NYRS+30
705.   NYRS15=NYRS+15
706.   DO 10 I=1,NYRS30
707.   DO 10 J=1,NPTYPE
708.   BCWIP2(I,J)=0.0
709.   10 RBCWIP2(I,J)=0.0
710. C*****
711. C***FIRST, WEIGHTED AVERAGE COSTS OF EQUITY AND DEBT ARE CALCULATED.
712. C***A VARIABLE NAME ENDING WITH '15' INDICATES THAT THE VARIABLE IS LAGG
713. C***15 YRS. (IE. WACD15(I)=WACD(I-15))
714.   DO 312 I=2,NYRS
715.     WACD(I)=(PERLT(I)*RTNWLT(I)+PERST(I)*RTST(I))/
716.     1 (PERLT(I)*RTNWLT(I)+PERST(I)*RTST(I)+PERCE(I)*RTCEQ(I)+
717.     2 PERPS(I)*RTNWPS(I))
718.     WACE15(I+15)=(PERCE(I)*RTCEQ(I)+PERPS(I)*RTNWPS(I))/
719.     1 (PERLT(I)*RTNWLT(I)+PERST(I)*RTST(I)+PERCE(I)*RTCEQ(I)+
720.     2 PERPS(I)*RTNWPS(I))
721.     WACD15(I+15)=WACD(I)
722.   312 CONTINUE
723.   DO 313 I=1,16
724.     WACE15(I)=WACE15(17)
725.   313 WACD15(I)=WACD15(17)
726.   DO 278 I=NYRS,NYRS15
727.     WACD15(I+15)=WACD(NYRS)

```

```

278.      278      WACE15(I+15)=WACE15(NYRS+15)
279.      C*****
280.      C***NOW THE CWIP ACCOUNTS ARE CALCULATED. THEY ARE A FUNCTION OF NEW
281.      C***BUILDS. IN THIS SECTION, YEAR I ACTUALLY REFERS TO YEAR I-15. THIS
282.      C***IS ONLY FOR COMPUTING CONVENIENCE.
283.      C*****
284.      DO 81 M=1,NPLNTS
285.          I=JYR(M)+15
286.          J=JPLANT(M)
287.          DO 89 II=2,NYRS30
288.              SCWIP(II)=0.0
289.      89          BSCWIP(II)=0
290.          DO 80 K=1,15
291.              FCWIP(I-15+K)=(CONPRO(16-K,J)*PLTCAP(ITYPE,M)*CAPCST(M))
292.              /CWPFC(M)
293.              YFCWIP(I-15+K,J)=YFCWIP(I-15+K,J)+FCWIP(I-15+K)
294.              SCWIP(I-15+K)=SCWIP(I-16+K)+FCWIP(I-15+K)
295.              IF (K.EQ.15) SCWIP(I-15+K)=0.0
296.              DO 520 L=1,K
297.                  IF (K.EQ.15) GO TO 519
298.                  BSCWIP(I-15+K)=(FCWIP(I-14-L+K)*CWPPER(M))+
299.                  (FCWIP(I-14-L+K)*((AFDCR+1.0)**(L-.5))*(1-
300.                  CWPPER(M)))+BSCWIP(I-15+K)
301.                  GO TO 520
302.      519          BSCWIP(I-15+K)=BSCWIP(I-16+K)+FCWIP(I-15+K)*CWPPER(M)+
303.      1          FCWIP(I-15+K)*((1.0+AFDCR)**((1.0-SERPER(M))/2.0)
304.      2          *(1.0-CWPPER(M)))
305.          GO TO 521
306.      520          CONTINUE
307.      521          IF (CWPPER(M).EQ.1.0) RBCWP2(I-15+K,J)=SCWIP(I-15+K)+
308.      1          RBCWP2(I-15+K,J)
309.          IF (CWPPER(M).GT.0.0 .AND. CWPPER(M).LT.1.0 .AND.
310.      1          RBAFDC(M).EQ.0.0) RBCWP2(I-15+K,J)=SCWIP(I-15+K)*
311.      2          CWPPER(M)+RBCWP2(I-15+K,J)
312.          IF (CWPPER(M).NE.1.0 .AND. RBAFDC(M).EQ.1.0)
313.      1          RBCWP2(I-15+K,J)=BSCWIP(I-15+K)+
314.      2          RBCWP2(I-15+K,J)
315.          BCWIP2(I-15+K,J)=BCWIP2(I-15+K,J)+BSCWIP(I-15+K)
316.          AFUDCE(I-15+K)=(BSCWIP(I-15+K)-BSCWIP(I-16+K)-
317.      1          FCWIP(I-15+K))*WACE15(I-15+K)+AFUDCE(I-15+K)
318.          AFUDCD(I-15+K)=(BSCWIP(I-15+K)-BSCWIP(I-16+K)-
319.      1          FCWIP(I-15+K))*WACD15(I-15+K)+AFUDCD(I-15+K)
320.          OFFSET(I-15+K)=(BSCWIP(I-15+K)-BSCWIP(I-16+K)-
321.      1          FCWIP(I-15+K))*RBAFDC(M)+OFFSET(I-15+K)
322.      80          CONTINUE
323.      81          CONTINUE
324.      C
325.          DO 350 I=2,NYRS30
326.              DO 349 J=1,NPTYPE
327.                  CWIP(I)=CWIP(I)+YFCWIP(I,J)
328.                  BCWIP(I)=BCWIP(I)+BCWIP2(I,J)
329.                  RBCWIP(I)=RBCWIP(I)+RBCWP2(I,J)
330.      349          CONTINUE
331.      350          CONTINUE
332.      C***WHEN NEW PLANT COMES ON LINE, ITS VALUE IS REMOVED FROM CWIP ACCOUNT
333.      DO 84 M=1,NPLNTS

```



```

784.          I=JYR(M)+15
785.          J=JPLANT(M)
786.          CWIP(I)=CWIP(I)-A2(I-15,J)
787.          BCWIP(I)=BCWIP(I)-BA2(I-15,J)
788.          IF (CWPPER(M).NE.1.0 .AND. RBAFDC(M).EQ.1.0)
789.            1      RBCWIP(I)=RBCWIP(I)-BA2(I-15,J)
790.      84      CONTINUE
791.          DO 86 I=2,NYRS30
792.            CWIP(I)=CWIP(I-1)+CWIP(I)
793.      86      CONTINUE
794.      C***NOW ALL VARIABLES ARE CORRECTED TO TRUE YEAR.
795.          DO 345 I=1,NYRS
796.            CWIP(I)=CWIP(I+15)
797.            BCWIP(I)=BCWIP(I+15)
798.            RBCWIP(I)=RBCWIP(I+15)
799.            AFUDCE(I)=AFUDCE(I+15)
800.            AFUDCD(I)=AFUDCD(I+15)
801.            OFFSET(I)=OFFSET(I+15)
802.            DO 344 J=1,NPTYPE
803.              YFCWIP(I,J)=YFCWIP(I+15,J)
804.              SALETX(I)=(YFCWIP(I,J)*SATXRT)+SALETX(I)
805.              K=I+SY2(J)-1
806.              IF (K.GT.NYRS) K=NYRS
807.              DO 341 KK=I,K
808.                PROPTX(KK)=(PRTXRT*BA2(I,J))+PROPTX(KK)
809.      341      CONTINUE
810.      344      CONTINUE
811.            AFUDC(ITYPE,I)=AFUDCE(I)+AFUDCD(I)
812.      345      CONTINUE
813.          RETURN
814.          END
815.          SUBROUTINE ITC(NYRS,CWIP,SY1,T,
816.      1  RITC,A2,SY2,YFCWIP,
817.      3  NPTYPE,DITC,ADJITC,AITC,XITCF,FLPR1,
818.      4  XITCD)
819.      C
820.      C*****
821.      C*
822.      C*          SUBROUTINE ITC
823.      C*          THIS SUBROUTINE CALCULATES ITC GENERATED,
824.      C*          AMORTIZED, DEFERRED, ETC.
825.      C*
826.      C*          VARIABLES IN SUBROUTINE ITC
827.      C*
828.      C*          ADJITC = NET ADJUSTMENTS TO DEFERRED ITC ACCOUNT *
829.      C*          DITC = ACCUMULATED DEFERRED ITC
830.      C*          DITCCN = DITC IN BASE YEAR WHICH WAS GENERATED BY*
831.      C*          ASSETS STILL UNDER CONSTRUCTION IN BASE YR
832.      C*          AITC = AMORTIZATION/YEAR OF DEFERRED ITC
833.      C*          XITCD = AMOUNT OF ITC GENERATED, YEAR 1, WHICH *
834.      C*          IS DEFERRED FOR RATEMAKING PURPOSES.
835.      C*          XITCF = AMOUNT OF ITC GENERATED, YEAR 1, WHICH *
836.      C*          IS FLOWED THROUGH FOR RATEMAKING.
837.      C*****
838.      C
839.          DIMENSION CWIP(80),SY1(15),T(15),

```

```

840.      2  A2(50,20),SY2(20),
841.      3  YFCWIP(80,20),
842.      5  DITC(50),ADJITC(50),
843.      6  AITC(50),XITCF(50),XITCD(50)
844.      C*****
845.      C***BASE YEAR DITC IS MADE UP OF TWO PARTS: 1)DITC ON ASSETS WHICH ARE
846.      C***STILL UNDER CONSTRUCTION;2) DITC ON EXISTING ASSETS. CLASS 1 SHOULD
847.      C***NOT BEGIN AMORTIZATION UNTIL ASSET COMES ON LINE (SEE NEW ASSETS),
848.      C***WHILE CLASS 2 SHOULD BE AMORTIZED IMMEDIATELY (IT IS AMORTIZED OVER
849.      C***THE REMAINING LIFE OF EXISTING ASSET 2 (STEAM)).
850.      C*****
851.      DITCCN=CWIP(1)*RITC*(1.0-FLPR1)
852.      K=SY1(2)-T(2)+1
853.      AITC1=(DITC(1)-DITCCN)/(K-1)
854.      IF (K.GT.NYRS) K=NYRS
855.      DO 72 I=2,K
856.      AITC(I)=AITC1+AITC(I)
857.      72  CONTINUE
858.      C*****
859.      C  ***NEW ASSETS***
860.      C*****
861.      C***FIRST, THE AMOUNT OF ITC GENERATED IS CALCULATED.
862.      DO 79 I=2,NYRS
863.      DO 79 J=1,NPTYPE
864.      XITCF(I)=RITC*YFCWIP(I,J)*FLPR1+XITCF(I)
865.      XITCD(I)=RITC*YFCWIP(I,J)*(1.0-FLPR1)+XITCD(I)
866.      79  CONTINUE
867.      C***NOW CALCULATE DEFERRED,AMORTIZATION ETC. OF ITC.
868.      DO 82 I=2,NYRS
869.      DO 81 J=1,NPTYPE
870.      K=I+SY2(J)-1
871.      IF (K.GT.NYRS) K=NYRS
872.      DO 80 KK=I,K
873.      80  AITC(KK)=(RITC*A2(I,J))/SY2(J)*(1.0-FLPR1)
874.      *      +AITC(KK)
875.      81  CONTINUE
876.      82  CONTINUE
877.      C
878.      DO 74 I=2,NYRS
879.      ADJITC(I)=XITCD(I)-AITC(I)
880.      DITC(I)=DITC(I-1)+ADJITC(I)
881.      74  CONTINUE
882.      C***WRITE OUT FOR DEBUGGING PURPOSES.
883.      WRITE(4,100)
884.      100  FORMAT(' YEAR',4X,'DITC',5X,'XITCF',5X,'XITCD',6X,'AITC')
885.      DO 110 I=1,NYRS
886.      110  WRITE(4,120) I,DITC(I),XITCF(I),XITCD(I),AITC(I)
887.      120  FORMAT(2X,I2,3X,4(F8.1,2X))
888.      RETURN
889.      END
890.      SUBROUTINE EXPENS(NYRS,AVFLCS,FCNUC,FUEL,GWHTOT,GWHTNUC,
891.      1  XNUCFL,
892.      3  OM,WC,TFUEL,
893.      4  GTXRT,ITYPE,AVGOM)
894.      C
895.      C*****

```

```

896. C*
897. C*          SUBROUTINE EXPENS
898. C*          THIS SUBROUTINE CALCULATES FUEL, O&M EXPENSES
899. C*          AND W/C
900. C*
901. C*          VARIABLES IN SUBROUTINE EXPENS
902. C*
903. C*          GWHTOT,GWNUC - TOTAL GENERATION AND NUCLEAR GENERATION
904. C*          AVFLCS = AVERAGE FUEL COST (NOMINAL)
905. C*          FCNUC = NUCLEAR FUEL COST (NOMINAL)
906. C*          FUEL = SYSTEM FUEL COST
907. C*          XNUCEX = NUCLEAR FUEL COST
908. C*          XNUCFL = NUCLEAR FUEL ON BALANCE SHEET
909. C*          OM = TOTAL O&M COST
910. C*
911. C*****
912. C
913. C          DIMENSION AVFLCS(50),FCNUC(50),FUEL(50),GWHTOT(3,50),GWNUC(3,50),
914. C          1 XNUCEX(50),XNUCFL(50),
915. C          4 OM(50),WC(50),TXFUEL(50),AVGOM(50)
916. C***CALCULATE FUEL COSTS.
917. C          DO 319 I=2,NYRS
918. C              FUEL(I)=AVFLCS(I)*GWHTOT(ITYPE,I)
919. C              XNUCEX(I)=FCNUC(I)*GWNUC(ITYPE,I)
920. C              XNUCFL(I)=2.0*XNUCEX(I)
921. C          319 CONTINUE
922. C              XNUCFL(1)=XNUCFL(2)
923. C***NOW CALCULATE OPERATION AND MAINTENANCE COSTS (O&M).
924. C          DO 200 I=2,NYRS
925. C          200 OM(I)=AVGOM(I)*GWHTOT(ITYPE,I)
926. C***NOW CALCULATE WORKING CAPITAL AND FUEL COSTS ADJUSTED FOR TAXES.
927. C          550 DO 95 I=2,NYRS
928. C              WC(I)=.125*(OM(I)+FUEL(I))
929. C              TXFUEL(I)=(FUEL(I))/(1-GTXRT)
930. C          95 CONTINUE
931. C          RETURN
932. C          END
933. C          SUBROUTINE AFCINC(NPTYPE,BA1,A1,WACD,TXRT,T,SY1,
934. C          1 NYRS,AFUDCD,SY2,BA2,A2,PRODEF,
935. C          2 FLPR2,DAFDC,AAFDC,XAFDCF,XAFDCD)
936. C
937. C*****
938. C*
939. C*          SUBROUTINE AFCINC
940. C*          THIS SUBROUTINE CALCULATES ACCOUNTS RELATING TO
941. C*          THE TAX SAVINGS FROM THE DEBT PORRION OF AFUDC
942. C*
943. C*          VARIABLES FOR SUBROUTINE AFCINC
944. C*
945. C*          DAFDC = DEFERRED AFUDC TAXSAVINGS
946. C*          AAFDC1,AAFDC2,AAFDC = AMORTIZATION OF AFUDC TAX SAVINGS
947. C*          XAFDCF = GENERATED AFUDC TAX SAVINGS FLOWED-THROUGH*
948. C*          XAFDCD = GENERATED AFUDC TAX SAVINGS WHICH IS DEFERRED*
949. C*****
950. C
951. C          DIMENSION AAFDC1(15),BA1(15),A1(15),WACD(50),T(15),SY1(15),

```

```

952.      1 AFUDCD(80),SY2(20),AAFDC2(50,20),BA2(50,20),A2(50,20),
953.      2 DAFDC(50),AAFDC(50),XAFDCD(50),XAFDCF(50),PRODEF(50)
954.      C***INITIALIZE
955.          DO 10 I=1,NYRS
956.          DO 10 J=1,NPTYPE
957.      10  AAFDC2(I,J)=0.0
958.      C*****
959.      C***OLD ASSETS FIRST: THE AFDC DEFERRAL IS AMORTIZED OVER THE LIFE OF
960.      C***THE ASSET.
961.      C*****
962.          DO 251 J=1,10
963.              AAFDC1(J)=((BA1(J)-A1(J))*WACD(2)*TXRT)/SY1(J)*(1-FLPR2)
964.              DAFDC(1)=(SY1(J)-T(J))*AAFDC1(J)+DAFDC(1)
965.              K=SY1(J)-T(J)+1
966.              IF (K.GT.NYRS) K=NYRS
967.              DO 256 I=2,K
968.                  AAFDC(I)=AAFDC(I)+AAFDC1(J)
969.      256  CONTINUE
970.      251  CONTINUE
971.      C*****
972.      C*** NEW ASSETS**
973.      C***THE DEFERRAL GENERATED IS THE TAX SAVINGS FROM THE DEBT PORTION OF
974.      C***AFUDC.
975.      C*****
976.      C
977.          WRITE(4,20)
978.      20  FORMAT(' YEAR',8X,'DAFDC',10X,'AAFDC',9X,'XAFDCF',9X,'XAFDCD')
979.          DO 253 I=2,NYRS
980.              XAFDCD(I)=AFUDCD(I)*TXRT*(1.0-FLPR2)
981.              XAFDCF(I)=AFUDCD(I)*TXRT*FLPR2
982.      C***THE DEFERRAL IS AMORTIZED OVER THE LIFE OF THE ASSET.
983.          DO 252 J=1,NPTYPE
984.              K=I+SY2(J)-1
985.              IF (K.GT.NYRS) K=NYRS
986.              DO 250 KK=I,K
987.                  AAFDC2(KK,J)=((BA2(I,J)-A2(I,J))*WACD(I)*TXRT)
988.                      1 /SY2(J)*(1.0-FLPR2)+AAFDC2(KK,J)
989.      250  CONTINUE
990.              AAFDC(I)=AAFDC(I)+AAFDC2(I,J)
991.      252  CONTINUE
992.              DAFDC(I)=DAFDC(I-1)+XAFDCD(I)-AAFDC(I)
993.              WRITE(4,270) I,DAFDC(I),AAFDC(I),XAFDCF(I),XAFDCD(I)
994.      270  FORMAT(1X,I2,6X,4(F10.0,5X))
995.      253  CONTINUE
996.      C***THE NET TAX DEFERRAL IS ADDED TO THE PRODEF ACCOUNT. BECAUSE THE
997.      C***BASE YEAR AMOUNT IS ALREADY REFLECTED IN PRODEF(I), ONLY THE CHANGE
998.      C***IS ADDED TO PRODEF
999.          DO 30 I=2,NYRS
1000.      30  PRODEF(I)=PRODEF(I)+(DAFDC(I)-DAFDC(1))
1001.      254  RETURN
1002.      END
1003.      SUBROUTINE INTEXP(NYRS,OLDLT,PERLT,BDE,AMD,UP,
1004.      1 BCWIP,EMDT,RTNWLT,XLINEX,XSINEX,RTST,PERST,XINEX,
1005.      2 SY1,T,BA1,NPTYPE,BASTS2,WC,ABDE,XNUCFL,PRODEF,
1006.      3 RTNWPS,PERPS,EMPS,OLDPS,DITC,OPT5,OPT8)
1007.      C

```

```

1008. C*****
1009. C*
1010. C*          SUBROUTINE INTEXP
1011. C*          THIS SUBROUTINE CALCULATES INTEREST EXPENSE
1012. C*
1013. C*          VARIABLES FOR SUBROUTINE INTEXP
1014. C*
1015. C*          UP = UTILITY PLANT IN SERVICE
1016. C*          AMD = TOTAL VALUE OF ASSETS THAT NEED TO BE FINANCED
1017. C*          OLDPS = STOCK OF EXISTING PREFERRED STOCK
1018. C*          EMPS = EMBEDDED COST OF PREFERRED STOCK
1019. C*          OLDLT = STOCK OF EXISTING BONDS
1020. C*          EMDT = EMBEDDED COST OF DEBT
1021. C*          XLINEX = LONG-TERM INTEREST EXPENSE
1022. C*          XSINEX = SHORT-TERM INTEREST EXPENSE
1023. C*          XINEX = INTEREST EXPENSE
1024. C*
1025. C*****
1026. C
1027. C          DIMENSION OLDLT(50),PERLT(50),BDE(50),AMD(50),UP(50),
1028. C          1 BCWIP(80),EMDT(50),RTNWLT(50),
1029. C          2 XLINEX(50),XSINEX(50),RTST(50),PERST(50),XINEX(50),
1030. C          3 SY1(15),T(15),BA1(15),BASTS2(50,20),
1031. C          4 RTNWPS(50),PERPS(50),EMPS(50),OLDPS(50),PREFER(50),
1032. C          5 WC(50),ABDE(50),XNUCFL(50),PRODEF(50),DITC(50),BONDS(50)
1033. C***INITIALIZE.
1034. C***FIRST CALCULATE VALUE OF ASSETS THAT NEEDS TO BE FINANCED. (AMD=
1035. C***ASSETS MINUS DEFERRALS)
1036. C          DO 93 J=1,10
1037. C              K=SY1(J)-T(J)+1
1038. C              IF (K.GT.NYRS) K=NYRS
1039. C              DO 92 I=1,K
1040. C                  UP(I)=BA1(J)+UP(I)
1041. C          92 CONTINUE
1042. C          93 CONTINUE
1043. C          DO 95 I=2,NYRS
1044. C              DO 94 J=1,NPTYPE
1045. C                  UP(I)=UP(I)+BASTS2(I,J)
1046. C          94 CONTINUE
1047. C          AMD(I)=UP(I)+BCWIP(I)+WC(I)-ABDE(I)-PRODEF(I)-DITC(I)+
1048. C          1 XNUCFL(I)
1049. C          A=AMD(I)+PRODEF(I)+DITC(I)
1050. C          WRITE(4,*) A
1051. C          95 CONTINUE
1052. C***NOW CALCULATE EMBEDDED COST OF PREFERRED STOCK.
1053. C          PREFER = LONG TERM DEBT
1054. C          RETIRE = RETIREMENTS OF BASE YEAR PREFERRED STOCK
1055. C          IF (OPT8 .EQ. 4) GO TO 100
1056. C          PREFER(1)=OLDPS(1)
1057. C          DO 30 I=2,NYRS
1058. C          PREFER(I)=0.0
1059. C          IF ((OPT8.NE.1).AND.(AMD(I).EQ.0.0)) GO TO 30
1060. C          PREFER(I)=AMD(I)*PERPS(I)
1061. C          IF (PREFER(I) .EQ. 0.0) GO TO 30
1062. C          RETIRE=PERPS(I)*BDE(I)
1063. C          OLDPS(I)=OLDPS(I-1)-RETIRE

```

```

1064.          IF (OLDPS(I) .GT. 0.0) GO TO 40
1065.          RETIRE=RETIRE+OLDPS(I)
1066.          OLDPS(I)=0.0
1067.  40      EMPS(I)=(EMPS(I-1)*PREFER(I-1)-EMPS(I)*RETIRE+
1068.          1      RTNWPS(I)*(PREFER(I)-PREFER(I-1)+RETIRE))
1069.          2      /PREFER(I)
1070.  30      CONTINUE
1071.  C***NOW CALCULATE EMBEDDED COST OF LONG TERM DEBT, THEN LONG AND SHORT
1072.  C***TERM INTEREST EXPENSE. THIS IS JUST THE INTEREST RATE OF THE DEBT
1073.  C***MULTIPLIED BY THE AVERAGE OUTSTANDING DEBT OVER THE YEAR.
1074.  C*      BONDS = LONG TERM DEBT
1075.  C*      RETIRE = RETIREMENTS OF BASE YEAR BONDS
1076.          WRITE(4,10)
1077.  10      FORMAT(' YEAR',10X,'OLDLT',11X,'RETIRED',7X,'EMDT',10X,'AMD',6X,
1078.          *      'EMPS')
1079.          IF (OPT8 .EQ. 4) GO TO 100
1080.          BONDS(1)=OLDLT(1)
1081.          DO 91 I=2,NYRS
1082.          BONDS(I)=0.0
1083.          IF ((OPT8.NE.1).AND.(AMD(I).EQ.0.0)) GO TO 91
1084.          BONDS(I)=AMD(I)*PERLT(I)
1085.          RETIRE=PERLT(I)*BDE(I)
1086.          OLDLT(I)=OLDLT(I-1)-RETIRE
1087.          IF (OLDLT(I) .GT. 0.0) GO TO 20
1088.          RETIRE=RETIRE+OLDLT(I)
1089.          OLDLT(I)=0.0
1090.  20      EMDT(I)=(EMDT(I-1)*BONDS(I-1)-EMDT(I)*RETIRE+
1091.          1      RTNWLT(I)*(BONDS(I)-BONDS(I-1)+RETIRE))
1092.          2      /BONDS(I)
1093.          WRITE(4,90) I,OLDLT(I),RETIRE,EMDT(I),AMD(I),EMPS(I)
1094.  90      FORMAT(2X,I2,6X,F10.1,5X,F10.1,5X,F6.4,5X,F10.0,5X,F6.4)
1095.  91      CONTINUE
1096.  C***CALCULATE INTEREST EXPENSES.
1097.          PAST=AMD(2)*PERLT(2)
1098.          DO 300 I=3,NYRS
1099.          PRES=AMD(I)*PERLT(I)
1100.          IF ((OPT5 .EQ. 2) .AND. (PRES .LT. PAST)) PRES=PAST
1101.          XLINEX(I)=EMDT(I)*(PAST+PRES)/2.0
1102.          XSINEX(I)=RTST(I)*(AMD(I)+AMD(I-1))/2.0*PERST(I)
1103.          XINEX(I)=XLINEX(I)+XSINEX(I)
1104.  300      PAST=PRES
1105.          XLINEX(2)=EMDT(2)*AMD(2)*PERLT(2)*(AMD(2)/AMD(3))
1106.          XSINEX(2)=RTST(2)*AMD(2)*PERST(2)*(AMD(2)/AMD(3))
1107.          XINEX(2)=XLINEX(2)+XSINEX(2)
1108.          RETURN
1109.  C***THIS SECTION IS CALLED WHEN OPT8=4 (IE. OLD MONEY RUN). IN THIS
1110.  C***CASE, THE EMBEDDED COST OF DEBT STAYS CONSTANT AT BASE YEAR VALUE.
1111.  100      DO 110 I=2,NYRS
1112.          EMDT(I)=EMDT(I-1)
1113.          EMPS(I)=EMPS(I-1)
1114.          XLINEX(I)=EMDT(I)*(AMD(I)+AMD(I-1))/2.0*PERLT(I)
1115.          XSINEX(I)=RTST(I)*(AMD(I)-AMD(I-1))/2.0*PERST(I)
1116.          XINEX(I)=XLINEX(I)+XSINEX(I)
1117.  110      WRITE(4,120) I,EMDT(I),AMD(I)
1118.  120      FORMAT(2X,I2,36X,F6.4,5X,F10.0)
1119.          RETURN

```

```

1120.      END
1121.      SUBROUTINE REVS(NYRS,AITC,XITCF,RBCWIP,WC,ABDE,PRODEF,
1122.      1  TDEREG,TDEWOA,XAFDCF,AAFDC,PERLT,EMDT,
1123.      2  PERST,RTST,PERCE,RTCEQ,PERPS,EMPS,OPT3,SERPER,
1124.      3  NPTYPE,BA2,OM,UP,STXRT,TXRT,FUEL,XINEX,XNUCFL,
1125.      4  PROPTX,SALETX,GTXT,AFUDCD,BDE,REVLAG,TFUEL,
1126.      5  OFFSET,TOTSAL,REVREQ,ITYPE,OPT1,OPT8)
1127.      C
1128.      C*****
1129.      C*
1130.      C*          SUBROUTINE REVS
1131.      C*          THIS SUBROUTINE SOLVES FOR REVENUE REQUIREMENTS *
1132.      C*          AND THEN DETERMINES ELECTRIC REVENUES BASED *
1133.      C*          ON THE REGULATORY LAG.  IN ADDITION, THE REV-*
1134.      C*          ENUE REQUIREMENT IS BUILT UP FOR OUTPUT PUR-*
1135.      C*          POSES.
1136.      C*
1137.      C*  VARIABLES IN SUBROUTINE REVS
1138.      C*
1139.      C*          GENREG = GENERAL TAXES FOR REGULATORY PURPOSES
1140.      C*          STAREG = STATE TAXES FOR REGULATORY PURPOSES
1141.      C*          FEDREG = FEDERAL TAXES FOR REGULATORY PURPOSES (NET OF *
1142.      C*          ITC AND AFUDCD )
1143.      C*          REVREQ = REVENUE REQUIREMENT BUILT UP
1144.      C*          RR = RATE OF RETURN (WEIGHTED AVERAGE COST OF CAPITAL*
1145.      C*          REVLAG(I,1) = REVENUES W/ NO REGULATORY LAG
1146.      C*          REVLAG(I,2) = REVENUES W/ 1 YEAR REGULATORY LAG
1147.      C*          REVLAG(I,3) = REVENUES W/ 1 1/2 YEARS REGULATORY LAG
1148.      C*          REVLAG(I,4) = REVENUES W/ 2 YEARS REGULATORY LAG
1149.      C*          REVLAG(I,5) = REVENUES W/ ACTUAL LAG SCENARIO
1150.      C*          TDEREG=DEPRECIATION EXPENSE USED IN CALCULATING FEDERAL TAXES FOR
1151.      C*          REGULATORY PURPOSES. (THUS IN RATES)
1152.      C*          PURPOSES.
1153.      C*****
1154.      C*          RB = RATE BASE
1155.      C
1156.      C
1157.      DIMENSION AITC(50),XITCF(50),XNUCFL(50),REVREQ(3,50,23),
1158.      1  TDEREG(50),TDEWOA(50),XAFDCF(50),FEDREG(50),GENREG(50),
1159.      2  STAREG(50),AAFDC(50),RR(50),PERLT(50),EMDT(50),PERST(50),
1160.      3  RTST(50),PERCE(50),RTCEQ(50),PERPS(50),EMPS(50),RBCWIP(80),
1161.      4  WC(50),ABDE(50),PRODEF(50),SERPER(300),BA2(50,20),RB(50),
1162.      5  REVLAG(3,50,5),OM(50),FUEL(50),XINEX(50),
1163.      6  PROPTX(50),SALETX(50),AFUDCD(80),BDE(50),TFUEL(50),
1164.      7  OFFSET(80),UP(50),
1165.      8  TOTSAL(3,50)
1166.      C***FIRST CALCULATE THE YEAR END RATE BASE AND THE RETURN ON RATE BASE.
1167.      DO 10 I=1,NYRS
1168.      RB(I)=UP(I)+RBCWIP(I)+WC(I)-ABDE(I)-PRODEF(I)+XNUCFL(I)
1169.      10  RR(I)=PERLT(I)*EMDT(I)+PERST(I)*RTST(I)+PERCE(I)*RTCEQ(I)
1170.      1  +PERPS(I)*EMPS(I)
1171.      C***IF OPT1 IS NOT 1, THEN RATE BASE IS CALCULATED AS THE AVERAGE VALUE
1172.      C***OVER THE YEAR (EQUALS AVERAGE OF JAN 1 AND DEC 31 RATE BASE).
1173.      IF (OPT1 .EQ. 1) GO TO 20
1174.      DO 30 I=2,NYRS
1175.      J=NYRS+2-I

```

```

1176.      RB1=0.0
1177.      DO 40 K=1,NPTYPE
1178. 40      RB1=RB1+BA2(J,K)
1179.      RB1=RB1+RB(J-1)
1180. 30      RB(J)=(RB(J)+RB1)/2.0
1181. C*****
1182. C***CALCULATE THE REVENUE REQUIREMENT. (THIS BECOMES REVENUES IN
1183. C***THE NO-LAG SCENARIO). THIS LONG EQUATION IS THE SOLUTION OF 4 SIMUL
1184. C***TANEOUS EQUATIONS INVOLVING THE REVENUE REQUIREMENT. THESE EQUATION
1185. C***CAN BE EXTRACTED FROM THE REV REQ BUILD UP FOR OUTPUT (BELOW).
1186. C*****
1187. 20      DO 130 I=2,NYRS
1188.          REVLAG(ITYPE,I,1)=(RB(I)*RR(I)+FUEL(I)+OM(I)+BDE(I)
1189. 1          -AITC(I)-XITCF(I)-AAFDC(I)-XAFDCF(I)
1190. 2          -OFFSET(I)
1191. 3          +(PROPTX(I)+SALETX(I))*(1.0-TXRT)
1192. 4          +(FUEL(I)+OM(I)+TDEWOA(I)+XINEX(I))*(-STXRT+STXRT*TXRT)
1193. 5          +(FUEL(I)+OM(I)+TDEREG(I)+XINEX(I)-AFUDCD(I))*(-TXRT))
1194. 6          /(1.0-GTXRT-STXRT-TXRT+GTXRT*TXRT+STXRT*TXRT)
1195. 130      CONTINUE
1196. C***NOW 'BUILD UP' THE REVENUE REQUIREMENT FOR OUTPUT PURPOSES. THIS
1197. C***SECTION IS ONLY DONE SO THAT THE REVENUE REQUIREMENT BUILDUP CAN BE
1198. C***PRINTED.
1199.      DO 200 I=2,NYRS
1200.          GENREG(I)=(REVLAG(ITYPE,I,1)*GTXRT)+PROPTX(I)+SALETX(I)
1201.          STAREG(I)=(REVLAG(ITYPE,I,1)-FUEL(I)-OM(I)-
1202. 1          TDEWOA(I)-XINEX(I))*STXRT
1203.          FEDREG(I)=(REVLAG(ITYPE,I,1)-OM(I)-FUEL(I)-
1204. 1          GENREG(I)-STAREG(I)-TDEREG(I)-XINEX(I)+AFUDCD(I))*TXRT
1205.          REVREQ(ITYPE,I,1)=UP(I)
1206.          REVREQ(ITYPE,I,2)=RBCWIP(I)
1207.          REVREQ(ITYPE,I,3)=XNUCFL(I)
1208.          REVREQ(ITYPE,I,4)=WC(I)
1209.          REVREQ(ITYPE,I,5)=ABDE(I)
1210.          REVREQ(ITYPE,I,6)=PRODEF(I)
1211.          REVREQ(ITYPE,I,7)=REVREQ(ITYPE,I,1)+REVREQ(ITYPE,I,2)+
1212. 1          REVREQ(ITYPE,I,3)+REVREQ(ITYPE,I,4)-REVREQ(ITYPE,I,5)
1213. 2          -REVREQ(ITYPE,I,6)
1214.          REVREQ(ITYPE,I,8)=RB(I)
1215.          REVREQ(ITYPE,I,9)=RR(I)
1216.          REVREQ(ITYPE,I,10)=REVREQ(ITYPE,I,8)*REVREQ(ITYPE,I,9)
1217.          REVREQ(ITYPE,I,11)=OM(I)
1218.          REVREQ(ITYPE,I,12)=FUEL(I)
1219.          REVREQ(ITYPE,I,13)=BDE(I)
1220.          REVREQ(ITYPE,I,14)=GENREG(I)
1221.          REVREQ(ITYPE,I,15)=STAREG(I)
1222.          REVREQ(ITYPE,I,16)=FEDREG(I)
1223.          REVREQ(ITYPE,I,17)=XITCF(I)
1224.          REVREQ(ITYPE,I,18)=AITC(I)
1225.          REVREQ(ITYPE,I,19)=XAFDCF(I)
1226.          REVREQ(ITYPE,I,20)=AAFDC(I)
1227.          REVREQ(ITYPE,I,21)=OFFSET(I)
1228.          REVREQ(ITYPE,I,22)=REVREQ(ITYPE,I,11)+REVREQ(ITYPE,I,12)+
1229. 1          REVREQ(ITYPE,I,13)+REVREQ(ITYPE,I,14)+REVREQ(ITYPE,I,15)+
1230. 2          REVREQ(ITYPE,I,16)-REVREQ(ITYPE,I,17)-REVREQ(ITYPE,I,18)-
1231. 3          REVREQ(ITYPE,I,19)-REVREQ(ITYPE,I,20)-REVREQ(ITYPE,I,21)

```



```

1232.          REVREQ(ITYPE,I,23)=REVREQ(ITYPE,I,10)+REVREQ(ITYPE,I,22)
1233. 200 CONTINUE
1234. C*****
1235. C  **MAKE REGULATORY LAG ADJUSTMENTS**
1236. C    THE ELECTRIC REVENUES ARE CALCULATED AS A FUNCTION OF REVENUE
1237. C    REQUIREMENTS AND THE REGULATORY LAG.
1238. C*****
1239.          IF (OPT8 .NE. 1) GO TO 320
1240.          DO 312 I=2,NYRS
1241.              REVLAG(ITYPE,I,2)=(TOTSAL(ITYPE,I)*((REVLAG(ITYPE,I-1,1)-
1242. 1          TXFUEL(I-1))/TOTSAL(ITYPE,I-1)))+TXFUEL(I)
1243.              IF (I.EQ.2) GO TO 312
1244.              REVLAG(ITYPE,I,3)=(TOTSAL(ITYPE,I)*.5*((REVLAG(ITYPE,I-2,
1245. 1          1)-TXFUEL(I-2))/TOTSAL(ITYPE,I-2))+((REVLAG(ITYPE,
1246. 2          I-1,1)-TXFUEL(I-1))/TOTSAL(ITYPE,I-1)))+TXFUEL(I)
1247.              REVLAG(ITYPE,I,4)=(TOTSAL(ITYPE,I)*((REVLAG(ITYPE,I-2,1)-
1248. 1          TXFUEL(I-2))/TOTSAL(ITYPE,I-2)))+TXFUEL(I)
1249. 312 CONTINUE
1250.          REVLAG(ITYPE,2,3)=REVLAG(ITYPE,2,2)
1251.          REVLAG(ITYPE,2,4)=REVLAG(ITYPE,2,2)
1252. C*****
1253. C***NOW THE ACTUAL LAG SCENARIO IS PUT IN REVLAG(ITYPE,I,5).
1254. C*****
1255. 320 IF (OPT8 .NE. 1) OPT3=1
1256.          DO 310 I=2,NYRS
1257.              REVLAG(ITYPE,I,5)=REVLAG(ITYPE,I,OPT3)
1258. 310 CONTINUE
1259.          RETURN
1260.          END
1261.          SUBROUTINE TAXES(NYRS,GENTX,REVLAG,GTXT,PROPTY,SALETX,STATX,
1262. 1          FUEL,OM,BDE,TDEWOA,XINEX,AFUDCE,TXRT,ADJADC,STATET,STXRT,
1263. 2          TTXEX,CIT,AAFDC,XAFDCF,XAFDCD,AFUDC,
1264. 3          ITYPE,ADJBDE,ADJITC,AITC,XITCF,EXTBF,EXTBD)
1265. C
1266. C*****
1267. C* *
1268. C* SUBROUTINE TAXES *
1269. C* THIS SUBROUTINE CALCULATES AND RECONCILES THE *
1270. C* FOLLOWING INCOME TAX QUANTITIES: *
1271. C* 1)STATUTORY TAXES (IE. 46% OF PRETAX INCOME *
1272. C* FROM THE INCOME STATEMENT) *
1273. C* 2)INCOME TAX EXPENSE (IE. THE INCOME TAX EX- *
1274. C* PENSE BOOKED ON THE INCOME STATEMENT) *
1275. C* 3)CURRENT PAYABLE INCOME TAXES (IE. THE ACTUAL *
1276. C* TAXES PAYED IN YEAR 10 GOVERNMENT) *
1277. C* GENERAL TAXES ARE ALSO CALCULATED. *
1278. C* *
1279. C* VARIABLES IN SUBROUTINE TAXES *
1280. C* *
1281. C* GENTX = GENERAL TAXES (GROSS RECEIPTS+PROPERTY+SALES) *
1282. C* STATX = FEDERAL STATUTORY INCOME TAXES *
1283. C* ADJADC = TAX REDUCTION FOR AFUDC *
1284. C* STATET = STATE INCOME TAXES, NET OF FED INCOME TAX BENERIT *
1285. C* TTXEX = TOTAL INCOME TAX EXPENSE BOOKED *
1286. C* CIT = CURRENTLY PAYABLE INCOME TAXES *
1287. C* *

```

```

1288. C*****
1289. C
1290. C
1291.     DIMENSION GENTX(50),REVLAG(3,50,5),PROPTX(50),SALETX(50),
1292.     1  STATX(50),FUEL(50),OM(50),BDE(50),TDEWOA(50),XINEX(50),
1293.     2  AFUDCE(50),ADJADC(50),STATET(50),TTXEX(50),AFUDC(3,50),
1294.     3  CIT(50),ADJBDE(50),EXTBF(50),AAFDC(50),XAFDCF(50),XAFDCD(50),
1295.     5  ADJITC(50),AITC(50),XITCF(50),EXTBD(50)
1296.     DO 235 I=2,NYRS
1297.         GENTX(I)=(REVLAG(ITYPE,I,5)*GTXT)+PROPTX(I)+SALETX(I)
1298.         STATX(I)=(REVLAG(ITYPE,I,5)-FUEL(I)-OM(I)-
1299.     1  BDE(I)-GENTX(I)-XINEX(I)+AFUDC(ITYPE,I))*TXRT
1300.         ADJBDE(I)=(BDE(I)-TDEWOA(I))*TXRT
1301.         ADJADC(I)=TXRT*AFUDCE(I)
1302.         STATET(I)=(REVLAG(ITYPE,I,5)-FUEL(I)-
1303.     1  OM(I)-TDEWOA(I)-XINEX(I))*STXRT*(1-TXRT)
1304.         TTXEX(I)=STATX(I)+ADJBDE(I)-ADJADC(I)
1305.     1  -AAFDC(I)-XAFDCF(I)
1306.     2  -AITC(I)-XITCF(I)-EXTBF(I)+STATET(I)
1307.         CIT(I)=TTXEX(I)-ADJITC(I)-
1308.     1  EXTBD(I)-XAFDCD(I)+AAFDC(I)
1309. 235  CONTINUE
1310.     RETURN
1311.     END
1312.     SUBROUTINE STMTS(NYRS,XINCST,REVLAG,OM,FUEL,
1313.     1  BDE,GENTX,TTXEX,AFUDCE,XLINEX,XSINEX,AFUDCD,BALSHT,
1314.     2  CWIP,BRTIRE,BA1,BASTS2,BCWIP,ABDE,XNUCFL,WC,AMD,
1315.     3  PRODEF,PERLT,PERST,PERCE,PERPS,EMPS,
1316.     4  COMDIV,DIVRAT,XINEX,CIT,AFUDC,
1317.     5  FUNDS,SY1,T,
1318.     6  NPTYPE,ITYPE,DITC,OPT5)
1319. C
1320. C
1321. C
1322. C*****
1323. C*
1324. C*     SUBROUTINE STMTS
1325. C*     THIS SUBROUTINE PREPARES ALL FINANCIAL STMTS
1326. C*
1327. C*     VARIABLES FOR SUBROUTINE STMTS
1328. C*
1329. C*     XINCST = INCOME STATEMENT
1330. C*     BALSHT = BALANCE SHEET
1331. C*     PREDIV = PREFERRED DIVIDENDS
1332. C*     COMDIV = COMMON DIVIDENDS
1333. C*     FUNDS = SOURCES & USES STATEMENT
1334. C*
1335. C*****
1336. C
1337.     DIMENSION XINCST(3,50,18),REVLAG(3,50,5),OM(50),FUEL(50),
1338.     1  BDE(50),GENTX(50),TTXEX(50),AFUDCE(80),
1339.     2  XLINEX(50),XSINEX(50),AFUDCD(80),BALSHT(3,50,23),
1340.     3  CWIP(80),BA1(15),BASTS2(50,20),BRTIRE(50),
1341.     4  BCWIP(80),ABDE(50),XNUCFL(50),WC(50),PRODEF(50),
1342.     5  PERLT(50),PERST(50),PERCE(50),PERPS(50),AMD(50),PREDIV(50),
1343.     6  EMPS(50),COMDIV(50),DIVRAT(50),

```

```

1344.      7 XINEX(50),CIT(50),AFUDC(3,50),
1345.      8 FUNDS(3,50,19),SY1(15),T(15),
1346.      1 DITC(50)
1347.      C
1348.      C *****PREPARATION OF INCOME STATEMENT*****
1349.      C
1350.      WRITE(4,534)
1351.      534 FORMAT(1X,'PLACE 6')
1352.      DO 150 I=2,NYRS
1353.          XINCST(ITYPE,I,1)=REVLAG(ITYPE,I,5)
1354.          XINCST(ITYPE,I,2)=OM(I)
1355.          XINCST(ITYPE,I,3)=FUEL(I)
1356.          XINCST(ITYPE,I,4)=BDE(I)
1357.          XINCST(ITYPE,I,5)=GENTX(I)
1358.          XINCST(ITYPE,I,6)=TTXEX(I)
1359.          XINCST(ITYPE,I,7)=XINCST(ITYPE,I,2)+XINCST(ITYPE,I,3)+
1360.      1      XINCST(ITYPE,I,4)+XINCST(ITYPE,I,5)+XINCST(ITYPE,I,6)
1361.          XINCST(ITYPE,I,8)=XINCST(ITYPE,I,1)-XINCST(ITYPE,I,7)
1362.          XINCST(ITYPE,I,9)=AFUDCE(I)
1363.          XINCST(ITYPE,I,10)=XINCST(ITYPE,I,9)
1364.          XINCST(ITYPE,I,11)=XINCST(ITYPE,I,8)+XINCST(ITYPE,I,10)
1365.          XINCST(ITYPE,I,12)=XLINEX(I)
1366.          XINCST(ITYPE,I,13)=XSINEX(I)
1367.          XINCST(ITYPE,I,14)=AFUDCD(I)
1368.          XINCST(ITYPE,I,15)=XINCST(ITYPE,I,12)+XINCST(ITYPE,I,13)-
1369.      1      XINCST(ITYPE,I,14)
1370.          XINCST(ITYPE,I,16)=XINCST(ITYPE,I,11)-XINCST(ITYPE,I,15)
1371.      150 CONTINUE
1372.      C
1373.      C *****PREPARATION OF BALANCE SHEET*****
1374.      C
1375.      DO 169 J=1,10
1376.          K=SY1(J)-T(J)+1
1377.          IF (K.GT.NYRS) K=NYRS
1378.          DO 168 I=2,K
1379.              IF (J.LE.6) BALSHT(ITYPE,I,1)=BALSHT(ITYPE,I,1)+BA1(J)
1380.              IF (J.EQ.7) BALSHT(ITYPE,I,2)=BA1(J)
1381.              IF (J.EQ.8) BALSHT(ITYPE,I,3)=BA1(J)
1382.              IF (J.EQ.9 .OR. J.EQ.10) BALSHT(ITYPE,I,4)=BALSHT(ITYPE,
1383.      1      I,4)+BA1(J)
1384.      168 CONTINUE
1385.      169 CONTINUE
1386.      DO 170 I=2,NYRS
1387.      C***FIRST THE LIABILITIES ARE BUILT UP.
1388.          BALSHT(ITYPE,I,14)=(PEMPS(I)*AMD(I))
1389.          BALSHT(ITYPE,I,17)=(PERLT(I)*AMD(I))
1390.          IF ((OPT5 .EQ. 2) .OR. (I .EQ. 2)) GO TO 10
1391.      C***THE LONG TERM AND PREFERRED DEBT IS NOT ALLOWED TO BE RETIRED.
1392.          IF (BALSHT(ITYPE,I,14).GE.BALSHT(ITYPE,I-1,14)) GO TO 10
1393.          WC(I)=(BALSHT(ITYPE,I-1,14)-BALSHT(ITYPE,I,14))
1394.      *          +(BALSHT(ITYPE,I-1,17)-BALSHT(ITYPE,I,17))
1395.      *          +WC(I)
1396.          BALSHT(ITYPE,I,14)=BALSHT(ITYPE,I-1,14)
1397.          BALSHT(ITYPE,I,17)=BALSHT(ITYPE,I-1,17)
1398.      10      PREDIV(I)=EMPS(I)*(BALSHT(ITYPE,I-1,14)+BALSHT(ITYPE,I,14))/2.0
1399.          XINCST(ITYPE,I,17)=PREDIV(I)

```

```

1400.      XINCST(ITYPE,I,18)=XINCST(ITYPE,I,16)-XINCST(ITYPE,I,17)
1401.      COMDIV(I)=DIVRAT(I)*XINCST(ITYPE,I,18)
1402.      BALSHT(ITYPE,I,15)=XINCST(ITYPE,I,18)-COMDIV(I)+
1403.      1      BALSHT(ITYPE,I-1,15)
1404.      BALSHT(ITYPE,I,13)=PERCE(I)*AMD(I)-BALSHT(ITYPE,I,15)
1405.      IF ((OPT5 .EQ. 2) .OR. (I .EQ. 2)) GO TO 20
1406.      C***THE COMMON STOCK IS NOT ALLOWED TO BE RETIRED
1407.      IF (BALSHT(ITYPE,I,13).GE.BALSHT(ITYPE,I-1,13)) GO TO 20
1408.      WC(I)=(BALSHT(ITYPE,I-1,13)-BALSHT(ITYPE,I,13))
1409.      *      +WC(I)
1410.      BALSHT(ITYPE,I,13)=BALSHT(ITYPE,I-1,13)
1411.      20      BALSHT(ITYPE,I,16)=BALSHT(ITYPE,I,13)+BALSHT(ITYPE,I,14)
1412.      1      +BALSHT(ITYPE,I,15)
1413.      BALSHT(ITYPE,I,18)=(PERST(I)*AMD(I))
1414.      BALSHT(ITYPE,I,19)=BALSHT(ITYPE,I,17)+BALSHT(ITYPE,I,18)
1415.      BALSHT(ITYPE,I,20)=DITC(I)
1416.      BALSHT(ITYPE,I,21)=PRODEF(I)
1417.      BALSHT(ITYPE,I,22)=BALSHT(ITYPE,I,20)+BALSHT(ITYPE,I,21)
1418.      BALSHT(ITYPE,I,23)=BALSHT(ITYPE,I,16)+BALSHT(ITYPE,I,19)+
1419.      1      BALSHT(ITYPE,I,22)
1420.      A=BALSHT(ITYPE,I,19)+BALSHT(ITYPE,I,16)
1421.      WRITE(4,*) A
1422.      C***NEXT THE ASSET SIDE IS BUILT UP.
1423.      DO 176 J=1,12
1424.      176      BALSHT(ITYPE,I,1)=BALSHT(ITYPE,I,1)+BASTS2(I,J)
1425.      BALSHT(ITYPE,I,2)=BASTS2(I,13)+BALSHT(ITYPE,I,2)
1426.      BALSHT(ITYPE,I,3)=BASTS2(I,14)+BALSHT(ITYPE,I,3)
1427.      BALSHT(ITYPE,I,4)=BASTS2(I,15)+BASTS2(I,16)+BALSHT(ITYPE,
1428.      1      I,4)
1429.      BALSHT(ITYPE,I,5)=BCWIP(I)
1430.      BALSHT(ITYPE,I,6)=BALSHT(ITYPE,I,1)+BALSHT(ITYPE,I,2)+
1431.      1      BALSHT(ITYPE,I,3)+BALSHT(ITYPE,I,4)+BALSHT(ITYPE,I,5)
1432.      BALSHT(ITYPE,I,7)=ABDE(I)
1433.      BALSHT(ITYPE,I,8)=BALSHT(ITYPE,I,6)-BALSHT(ITYPE,I,7)
1434.      BALSHT(ITYPE,I,9)=XNUCFL(I)
1435.      BALSHT(ITYPE,I,10)=BALSHT(ITYPE,I,8)+BALSHT(ITYPE,I,9)
1436.      BALSHT(ITYPE,I,11)=WC(I)
1437.      BALSHT(ITYPE,I,12)=BALSHT(ITYPE,I,10)+BALSHT(ITYPE,I,11)
1438.      170      CONTINUE
1439.      C
1440.      C      *****PREPARATION OF SOURCES AND USES OF FUNDS*****
1441.      C
1442.      DO 155 I=2,NYRS
1443.      FUNDS(ITYPE,I,1)=XINCST(ITYPE,I,16)
1444.      FUNDS(ITYPE,I,2)=XINCST(ITYPE,I,4)
1445.      FUNDS(ITYPE,I,3)=BALSHT(ITYPE,I,20)-BALSHT(ITYPE,I-1,20)
1446.      FUNDS(ITYPE,I,4)=BALSHT(ITYPE,I,21)-BALSHT(ITYPE,I-1,21)
1447.      FUNDS(ITYPE,I,5)=AFUDC(ITYPE,I)
1448.      FUNDS(ITYPE,I,6)=FUNDS(ITYPE,I,1)+FUNDS(ITYPE,I,2)+
1449.      1      FUNDS(ITYPE,I,3)+FUNDS(ITYPE,I,4)-FUNDS(ITYPE,I,5)
1450.      FUNDS(ITYPE,I,7)=BALSHT(ITYPE,I,18)-BALSHT(ITYPE,I-1,18)
1451.      FUNDS(ITYPE,I,8)=BALSHT(ITYPE,I,17)-BALSHT(ITYPE,I-1,17)
1452.      FUNDS(ITYPE,I,9)=BALSHT(ITYPE,I,13)-BALSHT(ITYPE,I-1,13)
1453.      FUNDS(ITYPE,I,10)=BALSHT(ITYPE,I,14)-BALSHT(ITYPE,I-1,14)
1454.      FUNDS(ITYPE,I,11)=FUNDS(ITYPE,I,7)+FUNDS(ITYPE,I,8)+
1455.      1      FUNDS(ITYPE,I,9)+FUNDS(ITYPE,I,10)

```

```

1456.          FUNDS(ITYPE,I,12)=FUNDS(ITYPE,I,6)+FUNDS(ITYPE,I,11)
1457.          FUNDS(ITYPE,I,13)=BCWIP(I)-BCWIP(I-1)+(BALSHT(ITYPE,I,1)-
1458.          1      BALSHT(ITYPE,I-1,1))+(BALSHT(ITYPE,I,2)-BALSHT(ITYPE,
1459.          2      I-1,2))+(BALSHT(ITYPE,I,3)-BALSHT(ITYPE,I-1,3))+(BALSHT
1460.          3      (ITYPE,I,4)-BALSHT(ITYPE,I-1,4))
1461.          4      +BRTIRE(I)-AFUDC(ITYPE,I)
1462.          IF (I.EQ.2) FUNDS(ITYPE,I,13)=CWIP(I)-CWIP(I-1)+(BALSHT(
1463.          1      ITYPE,I,6)-BALSHT(ITYPE,I,5))-(BA1(1)+BA1(2)+BA1(3)+
1464.          2      BA1(4)+BA1(5)+BA1(6)+BA1(7)+BA1(8)+BA1(9)+BA1(10))
1465.          FUNDS(ITYPE,I,14)=COMDIV(I)
1466.          FUNDS(ITYPE,I,15)=PREDIV(I)
1467.          FUNDS(ITYPE,I,16)=BALSHT(ITYPE,I,11)-BALSHT(ITYPE,I-1,11)
1468.          FUNDS(ITYPE,I,17)=BALSHT(ITYPE,I,9)-BALSHT(ITYPE,I-1,9)
1469.          IF (I.EQ.2) FUNDS(ITYPE,I,17)=0.0
1470.          FUNDS(ITYPE,I,18)=FUNDS(ITYPE,I,13)+FUNDS(ITYPE,I,14)+
1471.          1      FUNDS(ITYPE,I,15)+FUNDS(ITYPE,I,16)+FUNDS(ITYPE,I,17)
1472.          IF (FUNDS(ITYPE,I,18).EQ.0.0) GO TO 155
1473.          FUNDS(ITYPE,I,19)=FUNDS(ITYPE,I,11)/FUNDS(ITYPE,I,18)*100.0
1474.          155  CONTINUE
1475.          C
1476.          RETURN
1477.          END
1478.          //LKED.SYSLMOD DD DSN=CN6376.MI2.NUFS.LOAD(TEST),
1479.          // DISP=(,CATLG),SPACE=(TRK,(12,2,1),RLSE),UNIT=DASD
1480.          //LKED.SYSLIN DD DSN=&LOADSET,DISP=(OLD,DELETE)
1481.          //      DD *
1482.          INCLUDE SUBRPT(SUBS)
1483.          //LKED.SUBRPT DD DISP=SHR,DSN=CN6376.MI2.NUFS.RPT.LOAD
1484.          /*

```

?

III. NUFS.RPT.FORT

NUFS.RPT.FORT consists of thirteen subroutines. These subroutines can be divided into two categories. The first set of subroutines appearing in NUF.S.RPT.FORT are the reporting subroutines. A list and description of these subroutines follows (listed in the order of which they appear in NUF.S.RPT.FORT) below.

<u>Routine</u>	<u>Description</u>
REPORT	- Prints all major financial statements - Income Statement, Balance Sheet, Tax Reconciliation, Sources and Uses Statement, and Revenue Requirement build-up.
SHTRPT	- Prints out a short report consisting of the values of key financial parameters.
REPRT1	- Calculates and prints a number of financial ratios including interest coverage, actual return on common equity, and AFUDC as a percentage of earnings.
REPRT2	- Calculates and prints either the fixed charge rate, pricing coefficients, or old money revenue requirements, when necessary.
ADDUP	- Adds the private and public results to yield results for total region.

The second set of subroutines are those which read and prepare the input data. A list and description of these subroutines follows (listed in the order of which they appear in NUF.S.RPT.FORT) below.

<u>Routine</u>	<u>Description</u>
COMENT	- This subroutine strips comments out of the commented input files.
FORM1	- Reads in files which have been created by NUF.S.FORM1.FORT. This data is historical financial data for private utilities found in the FERC FORM1.
FORM1M	- Reads in files which have been created by NUF.S.FORM1M.FORT. This data is historical financial data for public utilities found in the FERC FORM1M.
MEFS	- Reads in files which have been created by the MEFS to NUF.S interface (NUF.S.MEFS.FORT). This includes forecasts of builds, generation, and fuel cost.

<u>Routine</u>	<u>Description</u>
READ	- Reads in files prepared by user containing such assumptions as treatment of: CWIP, deferred taxes, and depreciation.
SUPPLY	- Reads in a data file which has been created by the ICF electric utility supply model. This subroutine is only used when running NUFS for a single utility. (i.e., never used for MEFS applications)
ZEROS1	- Sets variables equal to zero. If is called for each region/ownership combination analyzed within the run.
ZEROS2	- Sets variables equal to zero. It is called once for each region analyzed in the run.

A listing of NUFS.RPT.FORT now follows.

```

1. //MIZUNUFS JOB (6376,D02),'COMPILE',TIME=(0,08),MSGCLASS=A
2. //DELD EXEC PGM=IEFBR14,REGION=10K
3. //DD1 DD DSN=CN6376.MI2.NUFS.RPT.LOAD,DISP=(MOD,DELETE),
4. // UNIT=DASD,SPACE=(TRK,(10,1))
5. //FPC EXEC FORTGCL
6. //FORT.SYSIN DD *
7. C NUFS.RPT.FORT. THIS IS THE SECOND OF TWO SOURCE MODULES FOR
8. C THE NATIONAL UTILITY FINANCIAL STATEMENT (NUFS) MODEL. IF CONTAINS
9. C THE SUBROUTINES WHICH WRITE OUT THE OUTPUT AND READ IN THE INPUT
10. C DATA FOR NUFS. THE INPUT SUBROUTINES FOLLOW THE OUTPUT SUBROUTINES.
11. C WRITTEN BY MARK R. INGLIS, ICF INC., NOV 1,1981
12. SUBROUTINE REPORT(IYRS,XINCST,BALSHT,CIT,
13. 1 TTXEX,STATX,ADJBDE,ADJADC,STATET,AAFDC,XAFDCF,XAFDCD,
14. 2 REVREQ,NYRS,FUNDS,NAME1,NAME2,NAME3,
15. 3 ITYPE,ADJITC,AITC,XITCF,EXTBF,EXTBD,XITCD,OPT1)
16. C
17. C*****
18. C* *
19. C* SUBROUTINE REPORT *
20. C* THIS SUBROUTINE PRINTS ALL STATEMENTS *
21. C* *
22. C*****
23. C
24. DIMENSION IYRS(50),XINCST(3,50,18),BALSHT(3,50,23),CIT(50),
25. 1 TTXEX(50),
26. 2 STATX(50),ADJBDE(50),ADJADC(50),STATET(50),
27. 3 REVREQ(3,50,23),AAFDC(50),XAFDCF(50),XAFDCD(50),
28. 4 FUNDS(3,50,19),NAME1(6),NAME2(6),NAME3(6),EXTBF(50),
29. 5 ADJITC(50),AITC(50),XITCF(50),EXTBD(50),XITCD(50),NAME4(5),
30. 6 NAME5(5),TOTAL(50)
31. DATA NAME4/'YEAR','END','RAT','E BA','SE '/,NAME5/'AVER',
32. 1 'AGE ','RATE','BAS','E '
33. C
34. C
35. 889 WRITE(4,332)
36. 332 FORMAT(1X,'PLACE 8')
37. C
38. WRITE(6,151)
39. 151 FORMAT('1'//20X,'***** PRO FORMA STATEMENT OF EARNINGS ',
40. 1 '*****',5(/),75X,'YEAR ENDED DECEMBER 31'/45X,75(' '))
41. WRITE(6,152) NAME1,NAME2,NAME3,(IYRS(I),I=2,8),
42. 1 ((XINCST(ITYPE,I,K),I=2,8),K=1,8)
43. 152 FORMAT(3(/1X,6A4)/50X,7(14,7X)/45X,7(9(' '),2X)/75X,
44. 1 '(DOLLARS IN MILLIONS)'/1X,'ELECTRIC REVENUES',26X,7('$',
45. 2 F9.0,1X)/45X,7(9(' '),2X)/1X,'ELECTRIC EXPENSES'/3X,'OPER',
46. 3 'ATION & MAINTENANCE',19X,7(F9.0,2X)/3X,'FUEL',38X,7(F9.0,
47. 4 2X)/3X,'DEPRECIATION',30X,7(F9.0,2X)/3X,'GENERAL TAXES',29X,
48. 5 7(F9.0,2X)/3X,'INCOME TAXES',30X,7(F9.0,2X)/45X,7(9(' '),2X)/
49. 6 8X,'TOTAL ELECTRIC EXPENSES',14X,7(F9.0,2X)/45X,7(9(' '),
50. 7 2X)/10X,'ELECTRIC OPERATING INCOME',10X,7(F9.0,2X)/45X,
51. 8 7(9(' '),2X)/)
52. WRITE(6,153) ((XINCST(ITYPE,I,K),I=2,8),K=9,18)
53. 153 FORMAT(1X,'OTHER INCOME'/3X,'AFUDC-EQUITY FUNDS',24X,7(F
54. 1 9.0,2X)/45X,7(9(' '),2X)/8X,'TOTAL OTHER INCOME',19X,
55. 2 7(F9.0,2X)/45X,7(9(' '),2X)/10X,'INCOME BEFORE INTEREST',

```



```

56.      3 ' DEDUCTIONS',2X,7(F9.0,2X)/45X,7(9('-',),2X)/1X,'INTEREST',
57.      4 ' DEDUCTIONS'/3X,'INTEREST ON LONG-TERM DEBT',16X,7(F9.0,
58.      5 2X)/3X,'INTEREST ON SHORT-TERM DEBT',15X,7(F9.0,2X)/
59.      5 3X,'AFUDC-BORROWED FUNDS',21X,7((',F9.0,')')/45X,
60.      6 7(9('-',),2X)/8X,'TOTAL INTEREST DEDUCTIONS',12X,7(F9.0,2X)/
61.      7 45X,7(9('-',),2X)/1X,'NET INCOME',34X,7(F9.0,2X)/3X,'DIVI',
62.      8 'ENDS ON PREFERRED STOCKS',13X,7(F9.0,2X)/45X,7(9('-',),2X)
63.      9 /1X,'EARNINGS FOR COMMON STOCK',18X,7('$',F9.0,1X)/45X,
64.      1 7(9('-',),2X)/45X,7(9('-',),2X)/1'')
65.      WRITE(6,152) NAME1,NAME2,NAME3,(IYRS(I),I=9,15),
66.      1 ((XINCST(ITYPE,I,K),I=9,15),K=1,8)
67.      WRITE(6,153) ((XINCST(ITYPE,I,K),I=9,15),K=9,18)
68.      WRITE(6,152) NAME1,NAME2,NAME3,(IYRS(I),I=16,22),
69.      1 ((XINCST(ITYPE,I,K),I=16,22),K=1,8)
70.      WRITE(6,153) ((XINCST(ITYPE,I,K),I=16,22),K=9,18)
71.      IF (NYRS.LT.22) GO TO 174
72.      WRITE(6,152) NAME1,NAME2,NAME3,(IYRS(I),I=23,29),
73.      1 ((XINCST(ITYPE,I,K),I=23,29),K=1,8)
74.      WRITE(6,153) ((XINCST(ITYPE,I,K),I=23,29),K=9,18)
75.      IF (NYRS.LT.29) GO TO 174
76.      WRITE(6,152) NAME1,NAME2,NAME3,(IYRS(I),I=30,36),
77.      1 ((XINCST(ITYPE,I,K),I=30,36),K=1,8)
78.      WRITE(6,153) ((XINCST(ITYPE,I,K),I=30,36),K=9,18)
79.      IF (NYRS.LT.36) GO TO 174
80.      WRITE(6,152) NAME1,NAME2,NAME3,(IYRS(I),I=37,43),
81.      1 ((XINCST(ITYPE,I,K),I=37,43),K=1,8)
82.      WRITE(6,153) ((XINCST(ITYPE,I,K),I=37,43),K=9,18)
83.      C *****PRINT BALANCE SHEET*****
84.      174 WRITE(6,175)
85.      175 FORMAT(20X,'***** PRO FORMA BALANCE SHEET *****',1(/),75X,
86.      1 'DECEMBER 31'/45X,75('-',)/)
87.      WRITE(6,176)NAME1,NAME2,NAME3,(IYRS(I),I=2,8),((BALSHT(ITYPE,I,K),
88.      1 I=2,8),K=1,8)
89.      176 FORMAT(3(/1X,6A4)/50X,7(14,7X)/45X,7(9('-',),2X)/75X,'(DO',
90.      1 'LLARS IN MILLIONS)'/15X,'ASSETS & OTHER DEBITS'/1X,'U',
91.      2 'TILITY PLANT'/3X,'PRODUCTION',32X,7(F9.0,2X)/3X,'TRANS',
92.      3 'MISSION',30X,7(F9.0,2X)/3X,'DISTRIBUTION',30X,7(F9.0,
93.      4 2X)/3X,'OTHER',37X,7(F9.0,2X)/3X,'CONSTRUCTION WORK',
94.      5 ' IN PROGRESS',13X,7(F9.0,2X)/45X,7(9('-',),2X)/8X,
95.      6 'TOTAL UTILITY PLANT',18X,7(F9.0,2X)/45X,7(9('-',),2X)/
96.      7 3X,'ACCUMULATED DEPRECIATION',18X,7(F9.0,2X)/45X,
97.      8 7(9('-',),2X)/8X,'NET UTILITY PLANT, LESS NUCLEAR FUEL',
98.      9 1X,7(F9.0,2X)/45X,7(9('-',),2X))
99.      WRITE(6,177) ((BALSHT(ITYPE,I,K),I=2,8),K=9,16)
100.     177 FORMAT(3X,'NET NUCLEAR FUEL',26X,7(F9.0,2X)/45X,7(9('-',),
101.     1 2X)/8X,'NET UTILITY PLANT',20X,7(F9.0,2X)/45X,7(9('-',),
102.     2 2X)/1X,'WORKING CAPITAL',29X,7(F9.0,2X)/45X,7(9('-',),
103.     3 2X)/8X,'TOTAL ASSETS & OTHER DEBITS',10X,7(F9.0,2X)/
104.     4 45X,7(9('-',),2X)/45X,7(9('-',),2X)//15X,'LIABILITIES',
105.     5 ' & OTHER CREDITS'//1X,'PROPRIETARY CAPITAL'/3X,'COM',
106.     6 'MON STOCK',30X,7(F9.0,2X)/3X,'PREFERRED STOCK',27X,
107.     7 7(F9.0,2X)/3X,'RETAINED EARNINGS',25X,7(F9.0,2X)/45X,
108.     8 7(9('-',),2X)/8X,'TOTAL PROPRIETARY CAPITAL',12X,
109.     9 7(F9.0,2X)/45X,7(9('-',),2X))
110.     WRITE(6,178) ((BALSHT(ITYPE,I,K),I=2,8),K=17,23)
111.     178 FORMAT(1X,'LONG-TERM DEBT',30X,7(F9.0,2X)/1X,'SHORT-TERM DEBT'

```

```

112.      1  ,29X,7(F9.0,2X)/45X,7(9('-''),2X)/1X,'TOTAL DEBT',34X,7(F9.0,
113.      2  2X)/45X,7(9('-''),2X)/1X,'DEFERRED CREDITS'/3X,'ACCUMULATED',
114.      3  ' DEFERRED ITC',
115.      4  18X,7(F9.0,2X)/3X,'ACCUMULATED DEFERRED INCOME TAXES',
116.      5  9X,7(F9.0,2X)/45X,7(9('-''),2X)/8X,'TOTAL DEFERRED CREDITS',
117.      6  15X,7(F9.0,2X)/45X,7(9('-''),2X)/8X,'TOTAL LIABILITIES &',
118.      7  ' OTHER CREDITS',4X,7(F9.0,2X)/45X,7(9('-''),2X)/45X,
119.      8  7(9('-''),2X)/'1'/)
120.      WRITE(6,176) NAME1,NAME2,NAME3,(IYRS(I),I=9,15),
121.      1  ((BALSHT(ITYPE,I,K),I=9,15),K=1,8)
122.      WRITE(6,177) ((BALSHT(ITYPE,I,K),I=9,15),K=9,16)
123.      WRITE(6,178) ((BALSHT(ITYPE,I,K),I=9,15),K=17,23)
124.      WRITE(6,176) NAME1,NAME2,NAME3,(IYRS(I),I=16,22),
125.      1  ((BALSHT(ITYPE,I,K),I=16,22),K=1,8)
126.      WRITE(6,177) ((BALSHT(ITYPE,I,K),I=16,22),K=9,16)
127.      WRITE(6,178) ((BALSHT(ITYPE,I,K),I=16,22),K=17,23)
128.      IF (NYRS.LT.22) GO TO 138
129.      WRITE(6,176) NAME1,NAME2,NAME3,(IYRS(I),I=23,29),
130.      1  ((BALSHT(ITYPE,I,K),I=23,29),K=1,8)
131.      WRITE(6,177) ((BALSHT(ITYPE,I,K),I=23,29),K=9,16)
132.      WRITE(6,178) ((BALSHT(ITYPE,I,K),I=23,29),K=17,23)
133.      IF (NYRS.LT.29) GO TO 138
134.      WRITE(6,176) NAME1,NAME2,NAME3,(IYRS(I),I=30,36),
135.      1  ((BALSHT(ITYPE,I,K),I=30,36),K=1,8)
136.      WRITE(6,177) ((BALSHT(ITYPE,I,K),I=30,36),K=9,16)
137.      WRITE(6,178) ((BALSHT(ITYPE,I,K),I=30,36),K=17,23)
138.      IF (NYRS.LT.36) GO TO 138
139.      WRITE(6,176) NAME1,NAME2,NAME3,(IYRS(I),I=37,43),
140.      1  ((BALSHT(ITYPE,I,K),I=37,43),K=1,8)
141.      WRITE(6,177) ((BALSHT(ITYPE,I,K),I=37,43),K=9,16)
142.      WRITE(6,178) ((BALSHT(ITYPE,I,K),I=37,43),K=17,23)
143.      C
144.      C *****PRINT NOTES TO FINANCIAL STATEMENTS*****
145.      C (ONLY PRINTED FOR PRIVATE SECTOR)
146.      C
147.      138 IF (ITYPE.NE.1) GO TO 909
148.      DO 10 I=2,NYRS
149.      10  TOTAL(I)=XAFDCD(I)-AAFDC(I)+EXTBD(I)
150.      WRITE(6,139)
151.      139 FORMAT(' ',20X,'***** PRO FORMA NOTES TO FINANCIAL',
152.      1  ' STATEMENTS *****',5(/),1X,'INCOME TAX EXPENSE'//5X,
153.      2  'INCOME TAX EXPENSE OF COMPRISED OF THE FOLLOWING',
154.      3  ' COMPONENTS:'//75X,'YEAR ENDED DECEMBER 31'/45X,75('-''))
155.      WRITE(6,140) NAME1,NAME2,NAME3,(IYRS(I),I=2,8),(CIT(I),
156.      1  I=2,8),(EXTBD(I),I=2,8),(XAFDCD(I),I=2,8),(AAFDC(I),I=2,8),
157.      2  (TOTAL(I),I=2,8)
158.      140 FORMAT(3(/1X,6A4)/50X,7(14,7X)/45X,7(9('-''),2X)/75X,'(DOL',
159.      1  'LARS IN MILLIONS)'/3X,'CURRENT',
160.      2  ' INCOME TAXES',22X,7(F9.0,2X)/45X,7(9('-''),2X)/3X,
161.      3  'DEFERRED TAXES, NET'/5X,'ACCELERATED',
162.      4  ' DEPRECIATION',16X,7(F9.0,2X)
163.      5  /5X,'TAX SAVINGS FROM AFUDC-DEBT PORTION'
164.      6  /7X,'DEFERRED',31X,7(F9.0,2X)
165.      7  /7X,'AMORTIZATION (CREDIT)',17X,7('(',F9.0,')')
166.      8  /45X,7(9('-''),2X)/45X,7(F9.0,2X))
167.      WRITE(6,141) (XITCD(I),I=2,8),(AITC(I),I=2,8),

```

```

168.      1 (ADJITC(I),I=2,8),
169.      2 (TTXEX(I),I=2,8)
170.  141  FORMAT(3X,'INVESTMENT TAX CREDIT, NET'/5X,'DEFERRED',
171.      1 32X,7(F9.0,2X)/5X,'AMORTIZATION (CRE',
172.      2 'DIT)',18X,7('(',F9.0,')')/45X,7(9('-',2X)/45X,7(F9.0,
173.      3 2X)/45X,7(9('-',2X))
174.      4 //11X,'TOTAL',
175.      5 ' INCOME TAX EXPENSE',10X,7(F9.0,2X)/45X,7(9('-',2X)/
176.      6 45X,7(9('-',2X)//)
177.      WRITE(6,140) NAME1,NAME2,NAME3,(IYRS(I),I=9,15),(CIT(I),
178.      1 I=9,15),(EXTBD(I),I=9,15),(XAFDCD(I),I=9,15),(AAFDC(I),I=9,15),
179.      2 (TOTAL(I),I=9,15)
180.      WRITE(6,141) (XITCD(I),I=9,15),(AITC(I),I=9,15),
181.      1 (ADJITC(I),I=9,15),
182.      2 (TTXEX(I),I=9,15)
183.      WRITE(6,140) NAME1,NAME2,NAME3,(IYRS(I),I=16,22),(CIT(I),
184.      1 I=16,22),(EXTBD(I),I=16,22),(XAFDCD(I),I=16,22),
185.      2 (AAFDC(I),I=16,22),(TOTAL(I),I=16,22)
186.      WRITE(6,141) (XITCD(I),I=16,22),(AITC(I),I=16,
187.      1 22),(ADJITC(I),I=16,22),
188.      2 (TTXEX(I),I=16,22)
189.      IF (NYRS.LT.22) GO TO 143
190.      WRITE(6,140) NAME1,NAME2,NAME3,(IYRS(I),I=23,29),(CIT(I),
191.      1 I=23,29),(EXTBD(I),I=23,29),(XAFDCD(I),I=23,29),
192.      2 (AAFDC(I),I=23,29),(TOTAL(I),I=23,29)
193.      WRITE(6,141) (XITCD(I),I=23,29),(AITC(I),I=23,
194.      1 29),(ADJITC(I),I=23,29),
195.      2 (TTXEX(I),I=23,29)
196.      IF (NYRS.LT.29) GO TO 143
197.      WRITE(6,140) NAME1,NAME2,NAME3,(IYRS(I),I=30,36),(CIT(I),
198.      1 I=30,36),(EXTBD(I),I=30,36),(XAFDCD(I),I=30,36),
199.      2 (AAFDC(I),I=30,36),(TOTAL(I),I=30,36)
200.      WRITE(6,141) (XITCD(I),I=30,36),(AITC(I),I=30,
201.      1 36),(ADJITC(I),I=30,36),
202.      2 (TTXEX(I),I=30,36)
203.      IF (NYRS.LT.36) GO TO 143
204.      WRITE(6,140) NAME1,NAME2,NAME3,(IYRS(I),I=37,43),(CIT(I),
205.      1 I=37,43),(EXTBD(I),I=37,43),(XAFDCD(I),I=37,43),
206.      2 (AAFDC(I),I=37,43),(TOTAL(I),I=37,43)
207.      WRITE(6,141) (XITCD(I),I=37,43),(AITC(I),I=37,
208.      1 43),(ADJITC(I),I=37,43),
209.      2 (TTXEX(I),I=37,43)
210.  C
211.  143  WRITE(6,145)
212.  145  FORMAT('1',20X,'***** PRO FORMA NOTES TO FINANCIAL STATE',
213.      1 'MENTS, CONT *****',5(/),3X,'INCOME TAXES DIFFER FROM',
214.      2 ' AMOUNTS COMPUTED BY APPLYING THE STATUTORY'/5X,'TAX',
215.      3 ' RATE TO PRE-TAX INCOME AS FOLLOWS:'/75X,'YEAR ENDED DEC',
216.      4 'EMBER 31'/45X,75('-',))
217.      WRITE(6,146) NAME1,NAME2,NAME3,(IYRS(I),I=2,8),(STATX(I),
218.      1 I=2,8),(ADJBDE(I),I=2,8),(ADJADC(I),I=2,8),
219.      2 (XITCF(I),I=2,8),(AITC(I),I=2,8),
220.      3 (XAFDCF(I),I=2,8),(AAFDC(I),I=2,8),
221.      4 (EXTBF(I),I=2,8),(STATET(I),I=2,8),
222.      5 (TTXEX(I),I=2,8)
223.  146  FORMAT(3(/1X,6A4)/50X,7(I4,7X)/45X,7(9('-',2X)/75X,'DO',

```

```

224.      1 'LLARS IN MILLIONS')//1X,'INCOME TAXES ON PRE-TAX INCOME',
225.      2 ' AT THE'/3X,'STATUTORY FEDERAL RATE OF 46%',13X,7(F9.0,
226.      3 2X)//1X,'ADJUSTMENTS TO ABOVE AT 46%'/5X,'UNALLOWABLE ',
227.      4 'BOOK DEPRECIATION',11X,7(F9.0,2X)
228.      5 /5X,'AFUDC-EQUITY FUNDS',21X,7('(',F9.0,')')// 'ADJUSTMENTS',
229.      6 ' DUE TO TAX SAVINGS'/5X,'INVESTMENT TAX CREDIT'
230.      7 /7X,'NOT DEFERRED',26X,7('(',F9.0,')')/7X,'AMORTIZATION',26X,
231.      8 7('(',F9.0,')')/5X,'TAX SAVINGS FROM AFUDC-DEBT PORTION'
232.      9 /7X,'NOT DEFERRED',26X,7('(',F9.0,')')
233.      * /7X,'AMORTIZATION',26X,7('(',F9.0,')')
234.      1 /5X,'TAX SAVINGS FROM ACCELERATED'/7X,'DEPRECIATION',
235.      2 '-NOT DEFERRED',12X,7('(',F9.0,')')
236.      3 //1X,'STATE INCOME TAXES, NET OF FEDERAL'/3X,'INCOME TAX',
237.      4 ' BENEFIT',24X,7(F9.0,2X)/45X,7(9('-'),2X)/1X,'TOTAL',
238.      5 ' INCOME TAX EXPENSE',20X,7(F9.0,2X)/45X,7(9('-'),2X)/45X,
239.      6 7(9('-'),2X)//)
240.      WRITE(6,146) NAME1,NAME2,NAME3,(IYRS(I),I=9,15),(STATX(I),
241.      1 I=9,15),(ADJBDE(I),I=9,15),(ADJADC(I),I=9,15),
242.      2 (XITCF(I),I=9,15),(AITC(I),I=9,15),
243.      3 (XAFDCF(I),I=9,15),(AAFDC(I),I=9,15),
244.      4 (EXTBF(I),I=9,15),(STATET(I),I=9,15),
245.      5 (TTXEX(I),I=9,15)
246.      WRITE(6,146) NAME1,NAME2,NAME3,(IYRS(I),I=16,22),(STATX(I),
247.      1 I=16,22),(ADJBDE(I),I=16,22),(ADJADC(I),I=16,22),
248.      2 (XITCF(I),I=16,22),(AITC(I),I=16,22),
249.      3 (XAFDCF(I),I=16,22),(AAFDC(I),I=16,22),
250.      4 (EXTBF(I),I=16,22),(STATET(I),I=16,22),
251.      5 (TTXEX(I),I=16,22)
252.      IF (NYRS.LT.22) GO TO 909
253.      WRITE(6,146) NAME1,NAME2,NAME3,(IYRS(I),I=23,29),(STATX(I),
254.      1 I=23,29),(ADJBDE(I),I=23,29),(ADJADC(I),I=23,29),
255.      2 (XITCF(I),I=23,29),(AITC(I),I=23,29),
256.      3 (XAFDCF(I),I=23,29),(AAFDC(I),I=23,29),
257.      4 (EXTBF(I),I=23,29),(STATET(I),I=23,29),
258.      5 (TTXEX(I),I=23,29)
259.      IF (NYRS.LT.29) GO TO 909
260.      WRITE(6,146) NAME1,NAME2,NAME3,(IYRS(I),I=30,36),(STATX(I),
261.      1 I=30,36),(ADJBDE(I),I=30,36),(ADJADC(I),I=30,36),
262.      2 (XITCF(I),I=30,36),(AITC(I),I=30,36),
263.      3 (XAFDCF(I),I=30,36),(AAFDC(I),I=30,36),
264.      4 (EXTBF(I),I=30,36),(STATET(I),I=30,36),
265.      5 (TTXEX(I),I=30,36)
266.      IF (NYRS.LT.36) GO TO 909
267.      WRITE(6,146) NAME1,NAME2,NAME3,(IYRS(I),I=37,43),(STATX(I),
268.      1 I=37,43),(ADJBDE(I),I=37,43),(ADJADC(I),I=37,43),
269.      2 (XITCF(I),I=37,43),(AITC(I),I=37,43),
270.      3 (XAFDCF(I),I=37,43),(AAFDC(I),I=37,43),
271.      4 (EXTBF(I),I=37,43),(STATET(I),I=37,43),
272.      5 (TTXEX(I),I=37,43)
273.      C
274.      C *****PRINT SOURCES AND USES STATEMENT*****
275.      C
276.      909 WRITE(6,910)
277.      910 FORMAT('1'//20X,'***** PRO FORMA SOURCES AND USES OF FUNDS',
278.      1 ' *****',3(/),75X,'DECEMBER 31'//45X,75('-')//)
279.      WRITE(6,911) NAME1,NAME2,NAME3,(IYRS(I),I=2,8),((FUNDS(ITYPE,I,K),

```

```

280.      1 I=2,8),K=1,12)
281. 911  FORMAT(3(/1X,6A4)/50X,7(I4,7X)/45X,7(9('-',),2X)/75X,'(DO',
282.      1 'LLARS IN MILLIONS)'/1X,'SOURCES OF FUNDS: '/3X,'NET IN',
283.      2 'COME',32X,7(F9.0,2X)/3X,'DEPRECIATION',30X,7(F9.0,2X)/3X,
284.      3 'DEFERRED ITC-NET',26X,7(F9.0,2X)/3X,'DEFERRED INCOME',
285.      4 ' TAXES',21X,7(F9.0,2X)/3X,'AFUDC',36X,7('(',F9.0,
286.      5 ')')/45X,7(9('-',),2X)/5X,'TOTAL INTERNAL FUNDS',20X,
287.      6 7(F9.0,2X)/3X,'INCREASE IN SHORT-TERM DEBT',15X,7(F9.0,2X)/
288.      7 3X,'INCREASE IN LONG-TERM DEBT',16X,7(F9.0,2X)/3X,
289.      8 'ISSUANCE OF COMMON STOCK',18X,7(F9.0,2X)/3X,'ISSUANCE OF',
290.      9 ' PREFERRED STOCK',15X,7(F9.0,2X)/45X,7(9('-',),2X)/5X,
291.      1 'TOTAL EXTERNAL FUNDS',20X,7(F9.0,2X)/45X,7(9('-',),2X)/8X,
292.      2 'TOTAL SOURCES',24X,7(F9.0,2X)/45X,7(9('-',),2X)/)
293.      WRITE(6,912) ((FUNDS(ITYPE,I,K),I=2,8),K=13,19)
294. 912  FORMAT(45X,7(9('-',),2X)/1X,'APPLICATION OF FUNDS: '/3X,
295.      1 'CONSTRUCTION EXPENDITURES'/5X,'(EXCLUDING AFUDC)',23X,
296.      2 7(F9.0,2X)/3X,'COMMON STOCK DIVIDENDS',20X,7(F9.0,2X)/3X,
297.      3 'PREFERRED STOCK DIVIDENDS',17X,7(F9.0,2X)/3X,'CHANGE IN ',
298.      4 'WORKING CAPITAL',17X,7(F9.0,2X)
299.      4 /3X,'CHANGE IN NUCLEAR FUEL STOCK',14X,7(F9.0,2X)
300.      4 /45X,7(9('-',),2X)/5X,'TOTAL',
301.      5 ' APPLICATIONS',22X,7(F9.0,2X)/45X,7(9('-',),2X)/45X,
302.      6 7(9('-',),2X)///1X,'EXTERNAL FINANCING AS A PERCENT'/3X,
303.      7 'OF TOTAL APPLICATIONS',21X,7(F9.2,2X)///1'/)
304.      WRITE(6,911)NAME1,NAME2,NAME3,(IYRS(I),I=9,15),((FUNDS(ITYPE,I,K),
305.      1 I=9,15),K=1,12)
306.      WRITE(6,912) ((FUNDS(ITYPE,I,K),I=9,15),K=13,19)
307.      WRITE(6,911) NAME1,NAME2,NAME3,(IYRS(I),I=16,22),
308.      1 ((FUNDS(ITYPE,I,K),I=16,22),K=1,12)
309.      WRITE(6,912) ((FUNDS(ITYPE,I,K),I=16,22),K=13,19)
310.      IF (NYRS.LT.22) GO TO 180
311.      WRITE(6,911) NAME1,NAME2,NAME3,(IYRS(I),I=23,29),
312.      1 ((FUNDS(ITYPE,I,K),I=23,29),K=1,12)
313.      WRITE(6,912) ((FUNDS(ITYPE,I,K),I=23,29),K=13,19)
314.      IF (NYRS.LT.29) GO TO 180
315.      WRITE(6,911) NAME1,NAME2,NAME3,(IYRS(I),I=30,36),
316.      1 ((FUNDS(ITYPE,I,K),I=30,36),K=1,12)
317.      WRITE(6,912) ((FUNDS(ITYPE,I,K),I=30,36),K=13,19)
318.      IF (NYRS.LT.36) GO TO 180
319.      WRITE(6,911) NAME1,NAME2,NAME3,(IYRS(I),I=37,43),
320.      1 ((FUNDS(ITYPE,I,K),I=37,43),K=1,12)
321.      WRITE(6,912) ((FUNDS(ITYPE,I,K),I=37,43),K=13,19)
322.  C
323.  C *****PRINT REVENUE REQUIREMENT*****
324. 180  IF (OPT1 .EQ. 1) GO TO 185
325.      DO 181 I=1,5
326.      181 NAME4(I)=NAME5(I)
327.      185 WRITE(6,201)
328.      201 FORMAT(' ',20X,'***** PRO FORMA REVENUE REQUIREMENT *****',
329.      1 5(/),75X,'YEAR ENDED DECEMBER 31'/45X,75('-',)/)
330.      WRITE(6,202)NAME1,NAME2,NAME3,(IYRS(I),I=2,8),((REVREQ(ITYPE,I,K),
331.      1 I=2,8),K=1,7)
332. 202  FORMAT(3(/1X,6A4)/50X,7(I4,7X)/45X,7(9('-',),2X)/75X,'(DO',
333.      1 'LLARS IN MILLIONS)'/1X,'UTILITY PLANT IN SERVICE',20X,
334.      1 7(F9.0,2X)/1X,
335.      2 'ADDITIONS: '/5X,'CONSTRUCTION WORK IN PROGRESS',11X,7(F9.0,2X)

```

```

336.      2 /5X,'NUCLEAR FUEL',28X,7(F9.0,
337.      3 2X)/5X,'WORKING CAPITAL',25X,7(F9.0,2X)/1X,'DEDUCTIONS: '/5X,
338.      4 'ACCUMULATED DEPRECIATION',15X,7('(',F9.0,')')/5X,'ACCU',
339.      5 'MULATED DEFERRED TAXES',13X,7('(',F9.0,')')/45X,7(9('-',
340.      6 2X)/8X,'YEAR END RATE BASE',19X,7(F9.0,2X)/45X,7(9('-',2X)/45X,
341.      7 7(9('-',2X)/)
342.      WRITE(6,203) NAME4,((REVREQ(ITYPE,I,K),I=2,8),K=8,23)
343. 203   FORMAT(1X,'REVENUE REQUIREMENT'//1X,5A4,24X,7(F9.0,
344.      1 2X)/3X,'RATE OF RETURN ON RATE BASE',15X,7(F9.3,2X)/45X,
345.      2 7(9('-',2X)/1X,'RETURN ON RATE BASE',25X,7(F9.0,2X)/45X,
346.      3 7(9('-',2X)/1X,'EXPENSES: '/3X,'OPERATION & MAINTENANCE',
347.      4 19X,7(F9.0,2X)/3X,'FUEL',38X,7(F9.0,2X)/3X,'DEPRECIATION',
348.      5 30X,7(F9.0,2X)/3X,'GENERAL TAXES',
349.      5 29X,7(F9.0,2X)/3X,'STATE TAXES',31X,7(F9.0,2X)/3X,'FEDER',
350.      6 'AL INCOME TAXES (NET OF ITC)',9X,7(F9.0,2X)/3X,'ITC-FLOWED',
351.      7 ' THROUGH',23X,7('(',F9.0,')')/3X,'ITC-AMORTIZED',28X,
352.      7 7('(',F9.0,')')/3X,'TAX SAVINGS FROM AFUDCD-FLOWED THROUGH',3X,
353.      7 7('(',F9.0,')')/3X,'TAX SAVINGS FROM AFUDCD-',
354.      8 'AMORTIZED',8X,7('(',F9.0,')')/3X,
355.      8 'AFUDC OFFSET',29X,7('(',F9.0,')')/
356.      9 45X,7(9('-',2X)/10X,'TOTAL EXPENSES',21X,7(F9.0,2X)/
357.      1 45X,7(9('-',2X)/1X,'REVENUE REQUIREMENTS',24X,7(F9.0,
358.      2 2X)/45X,7(9('-',2X)/45X,7(9('-',2X)/'1'/)
359.      WRITE(6,202) NAME1,NAME2,NAME3,(IYRS(I),I=9,15),
360.      1 ((REVREQ(ITYPE,I,K),I=9,15),K=1,7)
361.      WRITE(6,203) NAME4,((REVREQ(ITYPE,I,K),I=9,15),K=8,23)
362.      WRITE(6,202) NAME1,NAME2,NAME3,(IYRS(I),I=16,22),
363.      1 ((REVREQ(ITYPE,I,K),I=16,22),K=1,7)
364.      WRITE(6,203) NAME4,((REVREQ(ITYPE,I,K),I=16,22),K=8,23)
365.      IF (NYRS.LT.22) GO TO 960
366.      WRITE(6,202) NAME1,NAME2,NAME3,(IYRS(I),I=23,29),
367.      1 ((REVREQ(ITYPE,I,K),I=23,29),K=1,7)
368.      WRITE(6,203) NAME4,((REVREQ(ITYPE,I,K),I=23,29),K=8,23)
369.      IF (NYRS.LT.29) GO TO 960
370.      WRITE(6,202) NAME1,NAME2,NAME3,(IYRS(I),I=30,36),
371.      1 ((REVREQ(ITYPE,I,K),I=30,36),K=1,7)
372.      WRITE(6,203) NAME4,((REVREQ(ITYPE,I,K),I=30,36),K=8,23)
373.      IF (NYRS.LT.36) GO TO 960
374.      WRITE(6,202) NAME1,NAME2,NAME3,(IYRS(I),I=37,43),
375.      1 ((REVREQ(ITYPE,I,K),I=37,43),K=1,7)
376.      WRITE(6,203) NAME4,((REVREQ(ITYPE,I,K),I=37,43),K=8,23)
377.      C
378. 960   RETURN
379.      END
380.      SUBROUTINE SHTRPT(NYRS,IYRS,NAME1,NAME3,AFCPER,ROE,COVRAT,FUNDS,
381.      1 ITYPE)
382.      C
383.      C*****
384.      C*
385.      C*      SUBROUTINE SHTRPT
386.      C*      THIS SUBROUTINE WRITES OUT A SHORT REPORT OF THE KEY
387.      C*      FINANCIAL PARAMETERS FORECAST FOR THIS RUN. THE RESULTS
388.      C*      ARE PRODUCED FOR EACH RUN, AND ARE WRITTEN TO UNIT 9.
389.      C*
390.      C*****
391.      DIMENSION IYRS(50),NAME1(6),NAME3(6),AFCPER(50),ROE(50),

```

```
392.      1 COVRAT(50),FUNDS(3,50,19)
393.      WRITE(8,100) NAME1,NAME3
394.  100  FORMAT(6A4/6A4/25X,'SUMMARY REPORT'/
395.      1 23X,'AFUDC AS A',8X,'EARNED',8X,'EXTERNAL'/
396.      2 9X,'INTEREST',5X,'PERCENTAGE OF',6X,'RETURN',7X,'FINANCING'/
397.      3'YEAR',5X,'COVERAGE',7X,'EARNINGS',8X,'ON EQUITY',5X,'PERCENTAGE')
398.      DO 10 I=2,NYRS
399.      WRITE(8,110) IYRS(I),COVRAT(I),AFCPER(I),ROE(I),
400.      1 FUNDS(ITYPE,I,19)
401.  110  FORMAT(I4,5X,F8.4,6X,F10.2,7X,F8.5,4X,F10.2)
402.  10   CONTINUE
403.      RETURN
404.      END
405.      SUBROUTINE REPR1(NYRS,REVLAG,DISCR,TOTSAL,
406.      1 BALSHT,XINCST,IYRS,COVRAT,AFCPER,ROE,
407.      2 AFUDC,COMDIV,NAME1,XNSHRS,P,
408.      3 NAME2,NAME3,ITYPE,RTCEQ,OPT8)
409.      C
410.      C*****
411.      C*
412.      C*      SUBROUTINE REPR1
413.      C*      THIS SUBROUTINE CALCULATES AND PRINTS VARIOUS
414.      C*      FINANCIAL RATIOS.
415.      C*
416.      C*      VARIABLES FOR SUBROUTINE REPR1
417.      C*
418.      C*      PVREV = YEARLY PRESENT VALUE OF REVENUE REQUIREMENTS*
419.      C*      CUMPVR = CUMULATIVE PRESENT VALUE OF REVENUE
420.      C*      REQUIREMENTS
421.      C*      PRICE = AVERAGE PRICE OF ELECTRICITY ($/KWH)
422.      C*      AFCPER = AFUDC AS A % OF INCOME
423.      C*
424.      C*****
425.      C
426.      DIMENSION PVREV(50),REVLAG(3,50,5),CUMPVR(50),TOTSAL(3,50),
427.      1 COVRAT(50),PRICE(50),
428.      2 BALSHT(3,50,23),XINCST(3,50,18),
429.      3 IYRS(50),AFUDC(3,50),AFCPER(50),BOOKVL(50),
430.      4 PBRAT(50),XNSHRS(50),DPS(50),COMDIV(50),
431.      5 ROE(50),NAME1(6),NAME2(6),NAME3(6),E(50),
432.      6 D(50),G(50),GE(50),ABUS(50),APS(50),P(50),
433.      7 RTCEQ(50),BUS(50)
434.      C
435.      C      *****CALCULATE AND PRINT PV OF ELECTRIC REVENUES*****
436.      C
437.  340  WRITE(6,350) NAME1,NAME2,NAME3,DISCR
438.  350  FORMAT(' ',3(/1X,6A4)/5(/),1X,'YEAR',6X,'ELEC REVS',5X,
439.      1 'ELEC REVS',5X,'ELEC REVS',5X,'DISCOUNT RATE USED=',F6.4/14X,
440.      2 'NOM',12X,'PV',10X,'CUMPV'/)
441.      C
442.      CUMPVR(1)=0.0
443.      DO 352 I=2,NYRS
444.      PVREV(I)=REVLAG(ITYPE,I,5)/((1.0+DISCR)**(I-2))
445.      CUMPVR(I)=CUMPVR(I-1)+PVREV(I)
446.  349  WRITE(6,351) IYRS(I),REVLAG(ITYPE,I,5),PVREV(I),
447.      1 CUMPVR(I)
```

```

448. 351      FORMAT(1X,I4,5X,F9.0,5X,F9.0,4X,F10.0)
449. 352      CONTINUE
450.      IF (OPT8 .NE. 1) RETURN
451. C
452. C      *****CALCULATIONS OF INTEREST COVERAGE (SEC METHOD)*****
453. C
454.      WRITE(6,961) NAME1,NAME2,NAME3
455. 961      FORMAT(3(/1X,6A4),5(/),1X,'INTEREST COVERAGE CALCULATIONS'/
456. 1 6X,'YEAR',5X,'SEC FORMULA'/)
457. C
458.      DO 938 I=2,NYRS
459.          COVRAT(I)=(XINCST(ITYPE,I,8)
460. 1          +XINCST(ITYPE,I,6)
461. 2          +.10*AFUDC(ITYPE,I))/XINCST(ITYPE,I,12)
462.          WRITE(6,962) IYRS(I),COVRAT(I)
463. 962      FORMAT(6X,I4,7X,F6.4)
464. 938      CONTINUE
465. C
466. C      *****CALCULATE ELECTRICITY PRICES*****
467. C
468. 300      WRITE(6,212) NAME1,NAME2,NAME3
469. 212      FORMAT('1',3(/1X,6A4),4(/),1X,'YEAR',8X,
470. 1 'ELEC REVS',5X,'TOTAL SALES',6X,'AVERAGE PRICE OF ELECT',
471. 2 'RICITY'/14X,'($10**6)',9X,'(GWH)',15X,'(MILLS/KWH)')
472.      DO 214 I=2,NYRS
473.          PRICE(I)=REVLG(ITYPE,I,5)/TOTSAL(ITYPE,I)
474.          WRITE(6,213) IYRS(I),REVLG(ITYPE,I,5),TOTSAL(ITYPE,I),
475. 1 PRICE(I)
476. 213      FORMAT(1X,I4,5X,F12.3,2X,F13.0,12X,F10.5)
477. 214      CONTINUE
478. C
479. C      *****COMMON EQUITY INFORMATION*****
480. C
481. C***(ONLY APPLICABLE TO THE PRIVATE SECTOR.)
482. C
483. C      MARKET PRICE TO BOOK FORECASTING ALGORITHM. BASED ON
484. C      ESTIMATING THE COST OF EQUITY CAPITAL FOR ELECTRIC
485. C      UTILITIES: 1958-1976, HOWARD E. THOMPSON, BELL
486. C      JOURNAL OF ECONOMICS
487. C
488. C          PBRAT(I)=(G(I)-ROE(I))/(G(I)-RTCEQ(I))
489. C
490. C          BOOKVL(I) = YEAR END TOTAL BOOK VALUE (SUM OF COMMON
491. C          STOCK AND RETAINED EARNINGS)
492. C          D(I) = COMMON DIVIDENDS
493. C          ROE(I) = CALCULATED RATE OF RETURN ON COMMON EQUITY
494. C          PBRAT(I) = MARKET TO BOOK RATIO
495. C          G(I) = GROWTH RATE IN TOTAL BOOK VALUE
496. C          GE(I) = GROWTH RATE IN COMMON EQUITY
497. C          XNSHRS(I) = NUMBER OF SHARES OF COMMON EQUITY
498. C          DPS(I) = DIVIDENDS PER SHARE
499. C          BUS(I) = END OF YEAR BOOK VALUE PER SHARE
500. C          ABUS(I) = AVERAGE BOOK VALUE PER SHARE
501. C          APS(I) = AVERAGE PRICE PER SHARE
502. C          P(I) = PRICE BASED ON YEAR END SHARES
503. C          E(I) = EARNINGS FOR COMMON EQUITY

```



```

504. C
505. IF (ITYPE .NE. 1) GO TO 363
506. BOOKVL(1)=BALSHT(ITYPE,1,13)+BALSHT(ITYPE,1,15)
507. C PBRAT(1)=P(1)/(BOOKVL(1)/XNSHRS(1))
508. C ROE(1)=XINCST(ITYPE,1,18)/(BALSHT(ITYPE,1,13)+
509. C 1 BALSHT(ITYPE,1,15))
510. C WRITE(4,8900) BOOKVL(1),PBRAT(1),ROE(1)
511. C8900 FORMAT(' 1ST YEAR VALUES: BOOK VALUE:',F10.4,'; MARKET TO BOOK',
512. C * ' RATIO:',F7.4,'; RETURN ON EQUITY:',F6.4)
513. DO 360 I=2,NYRS
514. BOOKVL(I)=BALSHT(ITYPE,I,13)+BALSHT(ITYPE,I,15)
515. C E(I)=XINCST(ITYPE,I,18)
516. C D(I)=COMDIV(I)
517. ROE(I)=XINCST(ITYPE,I,18)/((BOOKVL(I)+BOOKVL(I-1))/2.0)
518. C G(I)=(BOOKVL(I)-BOOKVL(I-1))/
519. C 1 ((BOOKVL(I)+BOOKVL(I-1))/2)
520. C PBRAT(I)=PBRAT(I-1)+((ROE(I)-ROE(
521. C 1 I-1))/(RTCE(I)-G(I)))
522. C GE(I)=(G(I)-((2.0*(E(I)-D(I)))/
523. C 1 (BOOKVL(I)+BOOKVL(I-1))))/
524. C 2 PBRAT(I)
525. C XNSHRS(I)=XNSHRS(I-1)*((2.0+GE(I))/
526. C 1 (2.0-GE(I)))
527. C DPS(I)=D(I)/XNSHRS(I)
528. C ABUS(I)=(BOOKVL(I)+BOOKVL(I-1))/
529. C 1 (XNSHRS(I)+XNSHRS(I-1))
530. C BUS(I)=BOOKVL(I)/XNSHRS(I)
531. C APS(I)=PBRAT(I)*ABUS(I)
532. C P(I)=(BOOKVL(I)/XNSHRS(I))*
533. C 1 PBRAT(I)
534. C WRITE(4,369) E(I),D(I),G(I),GE(I)
535. C 369 FORMAT(' EARNINGS:',F7.0,2X,'DIVIDENDS:',F7.0,2X,'GROWTH RATE IN',
536. C 1 'TOTAL BOOK VALUE:',F7.4,2X,'GROWTH RATE IN COMMON EQUITY:',F7.4)
537. 360 CONTINUE
538. 359 WRITE(6,361) NAME1,NAME2,NAME3
539. C361 FORMAT(' ',3(/1X,6A4)/,4(/),1X,'COMMON STOCK INFORMATION'/
540. C 1 45X,'BOOK VALUE',5X,'STOCK PRICE'/1X,'YEAR',1X,'# OF SHARES',
541. C 2 2X,'DIV/SHARE',2X,'M/B RATIO',3X,'END YEAR AVER',2X,
542. C 3 'END YEAR AVER',3X,'ROE'/)
543. 361 FORMAT(' ',3(/1X,6A4)/,4(/),' COMMON STOCK INFORMATION'/'YEAR',
544. 1 5X,'END BOOK VALUE',5X,'AVERAGE EARNED ROE')
545. DO 363 I=2,NYRS
546. C WRITE(6,362) IYRS(I),XNSHRS(I),DPS(I),
547. C 1 PBRAT(I),BUS(I),ABUS(I),P(I),
548. C 2 APS(I),ROE(I)
549. C362 FORMAT(1X,I4,4X,F7.0,6X,F4.2,7X,F5.3,6X,F5.2,3X,F5.2,2X,
550. C 1 F5.2,3X,F5.2,2X,F5.3)
551. WRITE(6,362) IYRS(I),BOOKVL(I),ROE(I)
552. 362 FORMAT(1X,I4,7X,F9.0,14X,F6.4)
553. 363 CONTINUE
554. C
555. C ***AFUDC AS A PERCENTAGE OF NET INCOME AVAILABLE FOR COMMON***
556. C
557. WRITE(6,950) NAME1,NAME2,NAME3
558. 950 FORMAT('1',3(/1X,6A4)/,4(/),1X,'AFUDC AS A % OF EARNINGS',
559. 1 ' AVAILABLE FOR COMMON'/1X,'YEAR',4X,'AFUDC',4X,'AVAILAB',

```

```

560.      2 'LE EARNINGS',4X,'%')
561.      DO 952 I=2,NYRS
562.          AFCPER(I)=AFUDC(ITYPE,I)/XINCST(ITYPE,I,18)*100.0
563.      949      WRITE(6,951) IYRS(I),AFUDC(ITYPE,I),XINCST(ITYPE,I,18),
564.      1      AFCPER(I)
565.      951      FORMAT(1X,I4,2X,F8.0,7X,F10.0,5X,F6.2)
566.      952      CONTINUE
567.      C
568.      C      ***REGULATORY LAG INFORMATION***
569.      C
570.      WRITE(6,920) NAME1,NAME2,NAME3
571.      920      FORMAT(' ',3(/1X,6A4)/,4(/),1X,'REGULATORY LAG IN',
572.      1      'FORMATION'//1X,'YEAR',4X,'NO LAG',3X,'1 YEAR LAG',4X,
573.      2      '1 & 1/2 YEARS LAG',4X,'TWO YEARS LAG')
574.      DO 922 I=2,NYRS
575.      924      WRITE(6,921) IYRS(I),(REVLAG(ITYPE,I,J),J=1,4)
576.      921      FORMAT(1X,I4,2X,F9.0,2X,F9.0,8X,F9.0,10X,F9.0)
577.      922      CONTINUE
578.      C
579.      RETURN
580.      END
581.      SUBROUTINE REPR2(NYRS,NAME1,NAME2,NAME3,DISCR,IYRS,REVLAG,OPT8,
582.      1      GINF,GINFX,JYR,JPLANT,PLTCAP,CAPCST,SY2,IPRIYR,BLDTOT,ITYPE)
583.      C*****
584.      C*
585.      C*      SUBROUTINE REPR2
586.      C*      THIS SUBROUTINE CALCULATES AND PRINTS EITHER THE
587.      C*      FIXED CHARGE RATE, OR THE PRICING COEFFICIENT.
588.      C*
589.      C*****
590.      DIMENSION NAME1(6),NAME2(6),NAME3(6),IYRS(50),REVLAG(3,50,5),
591.      1      GINF(50),JYR(300),JPLANT(300),PLTCAP(3,300),CAPCST(300),
592.      2      GINF(70),SY2(20),IPRIYR(3),BLDTOT(3),PVREV(50),CUMPVR(50)
593.      REAL NCRF,NFC,NFCR
594.      WRITE(7,100) NAME1,NAME2,NAME3
595.      100      FORMAT(3(1X,6A4))
596.      IF (OPT8 .EQ. 3) GO TO 200
597.      IF (OPT8 .EQ. 4) GO TO 300
598.      C
599.      C      *****CALCULATE PV OF ELECTRIC REVENUES*****
600.      C
601.      CUMPVR(1)=0.0
602.      DO 352 I=2,NYRS
603.          PVREV(I)=REVLAG(ITYPE,I,5)/((1.0+DISCR)**(I-2))
604.      352      CUMPVR(I)=CUMPVR(I-1)+PVREV(I)
605.      C
606.      C      *****FIXED CHARGE RATE CALCULATIONS*****
607.      C
608.      C***PV=CUMULATIVE PV OF REV REQS DISCOUNTED TO THE YEAR PLANT COMES ON
609.      C***RCRF=REAL CAPITAL RECOVERY FACTOR
610.      C***NCRF=NOMINAL CAPITAL RECOVERY FACTOR
611.      C***RFC= REAL FIXED CHARGE (REAL ANNUITY)
612.      C***NFC=NOMINAL FIXED CHARGE (NOMINAL ANNUITY)
613.      C***RFCR=REAL FIXED CHARGE RATE
614.      C***NFCR=NOMINAL FIXED CHARGE RATE
615.      PV=CUMPVR(NYRS)*(1.0+DISCR)**(JYR(1)-2)

```

```

616. C***CALCULATE CAPITAL RECOVERY FACTORS
617.     NCRF=0.0
618.     RCRF=0.0
619.     ISY2=IFIX(SY2(JPLANT(1)))
620.     DO 360 I=1,ISY2
621.         NCRF=NCRF+1.0/(1.0+DISCR)**(I)
622.     360 RCRF=RCRF+((1.0+GINF(I+JYR(1)-1))/(1.0+DISCR)**(I))
623.     NCRF=1.0/NCRF
624.     RCRF=1.0/RCRF
625. C***NOW CALCULATE AND PRINT RESULTS.
626.     RFC=PV*RCRF
627.     NFC=PV*NCRF
628.     COST=PLTCAP(ITYPE,1)*CAPCST(1)
629.     RFCR=RFC/COST
630.     NFCR=NFC/COST
631.     WRITE(7,400)
632.     400 FORMAT(' *****FIXED CHARGE RATE CALCULATIONS*****')
633.     WRITE(7,410) JPLANT(1),SY2(JPLANT(1))
634.     410 FORMAT(' PLANT TYPE:',I2,' PLANT LIFE:',F4.0)
635.     WRITE(7,420) IYRS(JYR(1))
636.     420 FORMAT(' YEAR PLANT COMES ON LINE:',I4)
637.     WRITE(7,430) IYRS(JYR(1)),PV
638.     430 FORMAT(' PV OF REV REQS DISCOUNTED TO ',I4,' DOLLARS:',F10.0)
639.     WRITE(7,440) RCRF
640.     440 FORMAT(' REAL CAPITAL RECOVERY FACTOR:',F6.4)
641.     WRITE(7,445) NCRF
642.     445 FORMAT(' NOMINAL CAPITAL RECOVERY FACTOR:',F6.4)
643.     WRITE(7,450) RFC
644.     450 FORMAT(' REAL ANNUITY:',F10.2)
645.     WRITE(7,455) NFC
646.     455 FORMAT(' NOMINAL ANNUITY:',F10.2)
647.     WRITE(7,460) COST
648.     460 FORMAT(' COST OF PLANT:',F10.0)
649.     WRITE(7,470) RFCR
650.     470 FORMAT(' ***REAL FIXED CHARGE RATE:',F7.5)
651.     WRITE(7,475) NFCR
652.     475 FORMAT(' ***NOMINAL FIXED CHARGE RATE:',F7.5)
653.     RETURN
654. C
655. C *****PRICING COEFFICIENT CALCULATIONS
656. C
657. C***PC=PRICING COEFFICIENT FOR IRUN
658.     200 WRITE(7,520) JPLANT(1)
659.     520 FORMAT(' *****PRICING COEFFICIENT CALCULATIONS FOR PLANT TYPE:',
660.         1 I2,' *****'/24X,' TOTAL VALUE',5X,' PRICING',
661.         2 '/YEAR',5X,' REVENUES',5X,' OF BUILDS ',5X,' COEFFICIENT')
662.     DO 500 IRUN=1,3
663.         PC=REVLAG(ITYPE,IPRIYR(IRUN),5)/BLDTOT(IRUN)
664.         WRITE(7,510) IYRS(IPRIYR(IRUN)),REVLAG(ITYPE,IPRIYR(IRUN),5),
665.         1 BLDTOT(IRUN),PC
666.     510 FORMAT(I4,5X,F9.0,6X,F9.0,7X,F6.4)
667.     500 CONTINUE
668.     RETURN
669. C
670. C *****OLD MONEY CALCULATIONS
671. C

```

```

672. 300 WRITE(7,10)
673. 10  FORMAT(' ****CONSTANT BASE YEAR DOLLAR REVENUE REQUIREMENTS FOR ',
674.      * 'EXISTING ASSETS****'/32X,'(OLD MONEY)''' YEAR',5X,'REVENUE ',
675.      * 'REQUIREMENTS')
676.      DO 20 I=2,NYRS
677.          CONRR=REVLAG(ITYPE,I,5)/GINFX(I)
678. 20    WRITE(7,30) IYRS(I),CONRR
679. 30    FORMAT(I4,10X,F9.0)
680.      RETURN
681.      END
682.      SUBROUTINE ADDUP(XINCST,BALSHT,FUNDS,REVREQ,REVLAG,AFUDC,NYRS,
683. 1 OPT8)
684.      C
685.      C*****
686.      C*
687.      C*          SUBROUTINE ADDUP
688.      C*          THIS SUBROUTINE ADDS UP PUBLIC AND PRIVATE
689.      C*          FINANCIAL STATEMENTS TO GET STATEMENTS
690.      C*          FOR THE ENTIRE NATION
691.      C*
692.      C*****
693.      C
694.      C
695.          DIMENSION XINCST(3,50,18),BALSHT(3,50,23),FUNDS(3,50,19),
696. 1 REVREQ(3,50,23),REVLAG(3,50,5),AFUDC(3,50)
697.      C
698.      DO 8000 I=2,NYRS
699.      C
700.          DO 8001 J=1,18
701.              XINCST(3,I,J)=XINCST(1,I,J)+XINCST(2,I,J)
702. 8001      CONTINUE
703.      C
704.          DO 8002 JJ=1,23
705.              BALSHT(3,I,JJ)=BALSHT(1,I,JJ)+BALSHT(2,I,JJ)
706. 8002      CONTINUE
707.      C
708.          DO 8003 JJJ=1,19
709.              FUNDS(3,I,JJJ)=FUNDS(1,I,JJJ)+FUNDS(2,I,JJJ)
710. 8003      CONTINUE
711.          IF ((OPT8.NE.1).AND.(FUNDS(3,I,18).EQ.0.0)) GO TO 8005
712.              FUNDS(3,I,19)=FUNDS(3,I,11)/FUNDS(3,I,18)*100.0
713.      C
714. 8005      DO 8004 JJJJ=1,23
715.              REVREQ(3,I,JJJJ)=REVREQ(1,I,JJJJ)+REVREQ(2,I,JJJJ)
716. 8004      CONTINUE
717.          IF ((OPT8.NE.1).AND.(REVREQ(3,I,8).EQ.0.0)) GO TO 8006
718.              REVREQ(3,I,9)=REVREQ(3,I,10)/REVREQ(3,I,8)
719.      C
720. 8006      AFUDC(3,I)=AFUDC(1,I)+AFUDC(2,I)
721.          DO 8007 JJJJJ=1,5
722.              REVLAG(3,I,JJJJJ)=REVLAG(1,I,JJJJJ)+REVLAG(2,I,JJJJJ)
723. 8007      CONTINUE
724.      C
725. 8000      CONTINUE
726.      C
727.          WRITE(6,8008)

```

```

728. 8008 FORMAT('1',10('/),1X,'THE FOLLOWING TABLES ARE CUMULATIVE FOR',
729. 1 ' PUBLIC AND PRIVATE UTILITIES:'''1')
730. RETURN
731. END
732. SUBROUTINE COMENT(NN)
733. C*****
734. C*
735. C* SUBROUTINE COMENT
736. C**THIS SUBROUTINE IS CALLED EACH TIME THE FINANCIAL INPUT FILE IS
737. C***READ. ITS PURPOSE IS TO STRIP OFF COMMENTS. A COMMENT IS ANY LINE
738. C***BEGINING WITH A * IN COLUMN 1.
739. C* NN = FILE NUMBER TO BE STRIPED
740. C*****
741. DATA STAR/'*/
742. 10 READ(NN,20) WORD
743. 20 FORMAT(A1)
744. IF (WORD .EQ. STAR) GO TO 10
745. BACKSPACE NN
746. RETURN
747. END
748. SUBROUTINE FORM1(NNN,BA1,A1,EMDT,GTXT,OLDLT,REVLG,
749. 1 DITC,PRODEF,TXFUEL,OLDPS,EMPS,
750. 2 BALSHT,WC,XINCST,TOTSAL,OPT8)
751. C*****
752. C* *
753. C* SUBROUTINE FORM1 *
754. C* THIS SUBROUTINE IS DESIGNED TO READ DATA IN FROM *
755. C* FILES CREATED BY NUFS.FORM1.FORT. THESE DATA ARE *
756. C* BASE YEAR VALUES FROM THE FORM 1. *
757. C* VARIABLES FOR SUBROUTINE FORM1 *
758. C* BA1 = BOOK VALUE OF ASSETS EXISTING IN BASE YEAR
759. C* A1 = BA1, NET OF CAPITALIZED AFUDC
760. C*****
761. DIMENSION BA1(15),A1(15),OLDLT(50),REVLG(3,50,5),
762. 1 DITC(50),PRODEF(50),TXFUEL(50),
763. 2 BALSHT(3,50,23),WC(50),EMDT(50),OLDPS(50),EMPS(50),
764. 3 XINCST(3,50,18),TOTSAL(3,50)
765. IF ((OPT8.NE.1).AND.(OPT8.NE.4)) GO TO 30
766. C***READ IN BASE YEAR SALES
767. READ(NNN,*) TOTSAL(1,1)
768. C***READ IN BASE YEAR ASSET BASE
769. DO 10 I=1,10
770. 10 READ(NNN,*)BA1(I)
771. DO 20 I=1,6
772. 20 A1(I)=.85*BA1(I)
773. DO 25 I=7,10
774. 25 A1(I)=BA1(I)
775. C***READ IN BASE YEAR STOCK OF BONDS
776. READ(NNN,*) OLDLT(1)
777. C***READ IN BASE YEAR SHORT TERM DEBT
778. READ(NNN,*) BALSHT(1,1,18)
779. C***READ IN BASE YEAR COMMON STOCK
780. READ(NNN,*) BALSHT(1,1,13)
781. C***READ IN BASE YEAR PREFERRED STOCK
782. READ(NNN,*) OLDPS(1)
783. C***READ IN BASE YEAR RETAINED EARNINGS

```

```

784.      READ(NNN,*) BALSHT(1,1,15)
785.      C***READ IN BASE YEAR EMBEDDED COST OF LONG TERM DEBT
786.      READ(NNN,*) EMDT(1)
787.      C***READ IN BASE YEAR EMBEDDED COST OF PREFERED STOCK
788.      READ(NNN,*) EMPS(1)
789.      C***READ IN BASE YEAR TOTAL ELECTRIC REVENUES
790.      READ(NNN,*) REVLG(1,1,1)
791.      C***READ IN BASE YEAR NET INCOME
792.      READ(NNN,*) XINCST(1,1,18)
793.      C***READ IN BASE YEAR WORKING CAPITAL
794.      READ(NNN,*) WC(1)
795.      C***READ IN BASE YEAR ACCUMULATED DEFERRED ITC
796.      READ(NNN,*) DITC(1)
797.      C***READ IN BASE YEAR ACCUMULATED DEFERRED INCOME TAXES
798.      READ(NNN,*) PRODEF(1)
799.      C***READ IN BASE YEAR FUEL AND O&M COSTS ADJUSTED FOR TAXES
800.      READ(NNN,*) TXFUEL(1)
801.      BALSHT(1,1,11)=WC(1)
802.      BALSHT(1,1,14)=OLDPS(1)
803.      BALSHT(1,1,17)=OLDLT(1)
804.      BALSHT(1,1,20)=DITC(1)
805.      BALSHT(1,1,21)=PRODEF(1)
806.      GO TO 60
807.      30  DO 40 I=1,24
808.      40  READ(NNN,50) A
809.      50  FORMAT(A4)
810.      C***READ IN BASE YEAR GENERAL TAX RATE
811.      60  READ(NNN,*) GTXRT
812.      RETURN
813.      END
814.      SUBROUTINE FORM1M(NNN,BA1,A1,EMDT,GTXRT,OLDLT,REVLG,
815.      1 TXFUEL,
816.      2 BALSHT,WC,XINCST,TOTSAL,OPT8)
817.      C*****
818.      C*
819.      C*      SUBROUTINE FORM1M
820.      C*      THIS SUBROUTINE IS DESIGNED TO READ DATA IN FROM *
821.      C*      FILES CREATED BY NUFFS.FORM1M.FORT.  THESE DATA ARE*
822.      C*      BASE YEAR VALUES FROM THE FORM 1M.
823.      C*      VARIABLES FOR SUBROUTINE FORM1M
824.      C*      BA1 = BOOK VALUE OF ASSETS EXISTING IN BASE YEAR
825.      C*      A1 = BA1, NET OF CAPITALIZED AFUDC
826.      C*****
827.      DIMENSION BA1(15),A1(15),OLDLT(50),REVLG(3,50,5),
828.      1 TXFUEL(50),EMDT(50),
829.      2 BALSHT(3,50,23),WC(50),
830.      3 XINCST(3,50,18),TOTSAL(3,50)
831.      IF ((OPT8.NE.1).AND.(OPT8.NE.4)) GO TO 30
832.      C***READ IN BASE YEAR SALES
833.      READ(NNN,*) TOTSAL(2,1)
834.      C***READ IN BASE YEAR ASSETS
835.      DO 10 I=1,10
836.      10  READ(NNN,*)BA1(I)
837.      DO 20 I=1,6
838.      20  A1(I)=.85*BA1(I)
839.      DO 25 I=7,10

```

```

840. 25 A1(I)=BA1(I)
841. C***READ IN BASE YEAR STOCK OF BONDS
842. READ(NNN,*) OLDLT(1)
843. C***READ IN BASE YEAR COMMON STOCK
844. READ(NNN,*) BALSHT(2,1,13)
845. C***READ IN BASE YEAR RETAINED EARNINGS
846. READ(NNN,*) BALSHT(2,1,15)
847. C***READ IN BASE YEAR EMBEDDED COST OF LONG TERM DEBT
848. READ(NNN,*) EMDT(1)
849. C***READ IN BASE YEAR TOTAL ELECTRIC REVENUES
850. READ(NNN,*) REVLG(2,1,1)
851. C***READ IN BASE YEAR NET INCOME
852. READ(NNN,*) XINCST(2,1,18)
853. C***READ IN BASE YEAR WORKING CAPITAL
854. READ(NNN,*) WC(1)
855. C***READ IN BASE YEAR FUEL AND O&M COSTS ADJUSTED FOR TAXES
856. READ(NNN,*) TXFUEL(1)
857. BALSHT(2,1,17)=OLDLT(1)
858. BALSHT(2,1,11)=WC(1)
859. GO TO 60
860. 30 DO 40 I=1,19
861. 40 READ(NNN,50) A
862. 50 FORMAT(A4)
863. C***READ IN BASE YEAR GENERAL TAX RATE
864. 60 READ(NNN,*) GTXRT
865. RETURN
866. END
867. SUBROUTINE MEFS(NNN,NPLNTS,JYR,JPLANT,PLTCAP,CAPCST,
868. 1 NYRS,GWHTOT,GWNUC,AVFLCS,FCNUC,
869. 2 AVGOM,TOTSAL,IYRS,GINFX,NIT,IFCRYR,IFCRBD,IPRIYR,IPRIBD,BLDTOT,
870. 3 UP18)
871. C
872. C*****
873. C* *
874. C* SUBROUTINE MEFS *
875. C* THIS SUBROUTINE PUTS OUTPUTS FROM THE MEFS *
876. C* INTO THE APPROPRIATE FINANCIAL MODEL VARIABLES *
877. C* *
878. C* GWHTOT = TOTAL GENERATION (INCLUDING NUC) (1000*GWH)
879. C* GWNUC = NUCLEAR GENERATION (1000*GWH)
880. C* PRIFAC = PRIVATE UTILITY FRACTION OF GENERATION *
881. C* PUBFAC = PUBLIC UTILITY FRACTION OF GENERATION *
882. C* JYR = YEAR BUILD COMES ON LINE
883. C* JPLANT = TYPE OF PLANT
884. C* PLTCAP = CAPACITY OF BUILD (MW)
885. C* CAPCST = CAPACITY BUILD COST ($/KW - BASE YEAR $)
886. C* NPLNTS = NUMBER OF BUILDS *
887. C* AVFLCS = AVERAGE FUEL COST (MILLS/GWH - BASE YEAR $)
888. C* FCNUC = AVER NUCLEAR FUEL COSTS
889. C* AVGOM = AVERAGE O & M COSTS
890. C* XLF1,XLF2,XLF3 = LOSS FACTORS BY TIME PERIOD
891. C* TOTSAL = SYSTEM SALES IN GWH *
892. C*****
893. C
894. C
895. DIMENSION JYR(300),JPLANT(300),PLTCAP(3,300),CAPCST(300),

```

```

896.      1  GWHTOT(3,50),GWHNUC(3,50),AVFLCS(50),FCNUC(50),
897.      2  AVGOM(50),IYRS(50),
898.      3  GINFX(70),TOTSAL(3,50),BLDTOT(3),IPRIYR(3),ISAVE(28)
899.  C***READ IN BASE YEAR GENERATIONFOR PRIVATE SECTOR
900.      NNN=12
901.      READ(NNN,*) GWHTOT(1,1)
902.  C***READ IN BASE YEAR GENERATION FOR PUBLIC SECTOR
903.      NNN=13
904.      READ(NNN,*) GWHTOT(2,1)
905.  C
906.  C***BUILDS AND GENERATION TOTALS FROM MEFS MUST BE ALLOCATED ***
907.  C***TO PUBLIC AND PRIVATE OWNERSHIP. THIS IS DONE ON THE BASIS
908.  C***OF BASE YEAR GENERATION
909.      PRIFAC=GWHTOT(1,1)/(GWHTOT(1,1)+GWHTOT(2,1))
910.      PUBFAC=1.0-PRIFAC
911.      WRITE(4,2070) GWHTOT(1,1),PRIFAC,GWHTOT(2,1),PUBFAC
912.  2070  FORMAT(' YEAR 1 GENERATION (%):  PRIVATE=',F14.3,'(',F6.4,')',5X,
913.      * 'PUBLIC=',F14.3,'(',F6.4,')')
914.      IF (OPT8 .EQ. 2) GO TO 100
915.      IF (OPT8 .EQ. 4) GO TO 300
916.  C***READ IN BUILDS AND CAPACITY COSTS. THE COSTS READ IN ARE REAL DOL-
917.  C***LARS AND THUS MUST BE INFLATED TO NOMINAL.
918.      NPLNTS=0
919.      DO 2050 K=1,300
920.          READ(NNNN,*) JYR(K),JPLANT(K),PLTCAP(3,K),CAPCST(K)
921.          IF (JYR(K).EQ.0) GO TO 2057
922.          NPLNTS=NPLNTS+1
923.          PLTCAP(1,K)=PLTCAP(3,K)*PRIFAC
924.          PLTCAP(2,K)=PLTCAP(3,K)*PUBFAC
925.          CAPCST(K)=CAPCST(K)/1000.0*GINFX(JYR(K))
926.  2050  CONTINUE
927.  2057  IF (OPT8 .EQ. 3) GO TO 200
928.  C***NOW READ IN GENERATION, TOTAL AND NUCLEAR
929.  C***ALSO, WRITE OUT GENERATION FOR DEBUGGING PURPOSES
930.      WRITE(4,10)
931.  10    FORMAT(45X,'MEFS GENERATION'/' ', 'YEAR',7X,'TOTAL',10X,'PRIVATE',
932.      * 9X,'PUBLIC',9X,'NUC-TOTAL',6X,'NUC-PRIVATE',5X,'NUC-PUBLIC')
933.      DO 2051 I=2,NYRS
934.          READ(NNNN,*) GWHTOT(3,I),GWHNUC(3,I)
935.          GWHTOT(1,I)=GWHTOT(3,I)*PRIFAC
936.          GWHTOT(2,I)=GWHTOT(3,I)*PUBFAC
937.          GWHNUC(1,I)=GWHNUC(3,I)*PRIFAC
938.          GWHNUC(2,I)=GWHNUC(3,I)*PUBFAC
939.          WRITE(4,2059) I,GWHTOT(3,I),GWHTOT(1,I),GWHTOT(2,I),GWHNUC(3,I),
940.      * GWHNUC(1,I),GWHNUC(2,I)
941.  2059  FORMAT(2X,I2,4X,(6(F14.3,2X)))
942.  2051  CONTINUE
943.  C***NOW READ AVERAGE FUEL , NUCLEAR FUEL, AND AVERAGE OM COSTS PER KWH.
944.  C***THESE ARE IN REAL DOLLARS, SO THEY ARE INFLATED TO NOMINAL.
945.      DO 2054 I=2,NYRS
946.          READ(NNNN,*) AVFLCS(I),FCNUC(I),AVGOM(I)
947.          AVFLCS(I)=AVFLCS(I)*GINFX(I)
948.          FCNUC(I)=FCNUC(I)*GINFX(I)
949.          AVGOM(I)=AVGOM(I)*GINFX(I)
950.  2054  CONTINUE
951.  C***READ IN LOSS FACTORS.

```



```

952.      READ(NNNN,*) XLF1,XLF2,XLF3
953.      C
954.      C***ADJUST GENERATION BY APPROPRIATE LOSS FACTOR TO GET SALES
955.      C      XLF1=FIRST YEAR - 1985
956.      C      XLF2=1986-1990
957.      C      XLF3=1991-END
958.      C
959.      DO 551 I=2,NYRS
960.          IF (IYRS(I).GT.1985) GO TO 552
961.          TOTSAL(1,I)=GWHTOT(1,I)*(1.0-XLF1)
962.          TOTSAL(2,I)=GWHTOT(2,I)*(1.0-XLF1)
963.          GO TO 550
964.      552  IF (IYRS(I).GT.1990) GO TO 553
965.          TOTSAL(1,I)=GWHTOT(1,I)*(1.0-XLF2)
966.          TOTSAL(2,I)=GWHTOT(2,I)*(1.0-XLF2)
967.          GO TO 550
968.      553  TOTSAL(1,I)=GWHTOT(1,I)*(1.0-XLF3)
969.          TOTSAL(2,I)=GWHTOT(2,I)*(1.0-XLF3)
970.      550  TOTSAL(3,I)=TOTSAL(1,I)+TOTSAL(2,I)
971.      551  CONTINUE
972.      RETURN
973.      C*****
974.      C***THIS SECTION IS CALLED WHEN FIXED CHARGE RATES
975.      C***ARE TO BE CALCULATED. IT SETS UP THE NECESSARY BUILD.
976.      C*****
977.      C*
978.      C*      IFCRBD = THE PLANT TYPE FOR WHICH THE FIXED CHARGE RATE
979.      C*      IS TO BE CALCULATED.
980.      C*      IFCRYR = THE YEAR THAT THE BUILD IS TO COME ON LINE
981.      C*
982.      C***IF NEC, FIRST READ IFCRYR AND IFCRBD (OR USE VALUES ALREADY READ)
983.      100  IF (NIT .NE. 1) GO TO 110
984.          CALL COMENT(9)
985.          READ(9,*) IFCRYR,IFCRBD
986.      C***NOW SET UP BUILD FOR FCR ANALYSIS.
987.      110  NPLNTS=1
988.          JYR(1)=IFCRYR
989.          JPLANT(1)=IFCRBD
990.          CAPCST(1)=1000.0
991.          PLTCAP(1,1)=1000.0*PRIFAC
992.          PLTCAP(2,1)=1000.0*PUBFAC
993.          PLTCAP(3,1)=1000.0
994.          RETURN
995.      C*****
996.      C***THIS SECTION IS CALLED WHEN PRICING COEFFICIENTS
997.      C***ARE TO BE CALCULATED. IT MAKES THE NECESSARY CHANGES
998.      C***TO THE VALUES READ IN SUBROUTINE MEFS SO THAT THE PRICING
999.      C***COEFFICIENTS CAN BE CALCULATED.
1000.      C*
1001.      C*      IPRIBD = THE PLANT TYPE FOR WHICH THE PRICING COEFFICIENT
1002.      C*      IS TO BE CALCULATED.
1003.      C*      BLDTOT = TOTAL 'VALUE' OF PLANT TYPE BUILT IN MEFS RUN *
1004.      C*
1005.      C*      IRUN (SEE BELOW) *
1006.      C*      IRUN = 1  MEFS BUILD SOLUTION FOR 1979-1984
1007.      C*      2      1979-1989

```

```

1008. C*          3          1979-1995
1009. C*          ISAVE = ARRAY OF BUILD NUMBERS WHICH ARE NOT TO BE ZEROED
1010. C*****
1011. C***IF NEC, FIRST READ IPRIBD AND IPRIYR (OR USE VALUES ALREADY READ)
1012. 200 IF (NIT .NE. 1) GO TO 210
1013.      CALL COMENT(9)
1014.      READ(9,*) IPRIBD,(IPRIYR(I),I=1,3)
1015. C***NOW CALCULATE BLDTOT AND ISAVE AND MAKE RELEVANT CAPCST REAL $'S
1016. 210 ICOUNT=0
1017.      DO 20 M=1,NPLNTS
1018.          IF (JPLANT(M).NE.IPRIBD .OR. JYR(M).GE.IPRIYR(3))
1019.              *          GO TO 20
1020.              CAPCST(M)=CAPCST(M)/GINFX(JYR(M))
1021.              VALUE=PLTCAP(3,M)*CAPCST(M)
1022.              IF (JYR(M).LE.IPRIYR(1)) BLDTOT(1)=BLDTOT(1)+VALUE
1023.              IF (JYR(M).LE.IPRIYR(2)) BLDTOT(2)=BLDTOT(2)+VALUE
1024.              BLDTOT(3)=BLDTOT(3)+VALUE
1025.              ICOUNT=ICOUNT+1
1026.              ISAVE(ICOUNT)=M
1027. 20 CONTINUE
1028. C***NOW ZERO OUT ALL OTHER BUILDS.
1029.      NPLNTS=ICOUNT
1030.      DO 30 M=1,NPLNTS
1031.          JYR(M)=JYR(ISAVE(M))
1032.          JPLANT(M)=JPLANT(ISAVE(M))
1033.          CAPCST(M)=CAPCST(ISAVE(M))
1034.      DO 40 I=1,3
1035. 40      PLTCAP(I,M)=PLTCAP(I,ISAVE(M))
1036. 30 CONTINUE
1037. C***READ REST OF MEFS INTERFACE FILE BUT IGNORE IT.
1038.      NUM=(NYRS-1)*2+1
1039.      DO 50 I=1,NUM
1040. 50      READ(NNN,60) TEMP
1041. 60      FORMAT(A1)
1042.      RETURN
1043. C*****
1044. C***THIS SECTION IS CALLED WHEN RUNNING OLD MONEY CALCULATIONS (IE. ES-
1045. C***TIMATING REVENUE REQUIREMENTS FOR EXISTING ASSETS). A DUMMY BUILD
1046. C***IS SET UP WITH ZERO COSTS.
1047. C*****
1048. 300 NPLNTS=1
1049.      JPLANT(1)=1
1050.      JYR(1)=2
1051.      RETURN
1052.      END
1053.      SUBROUTINE READ(NN,NAME1,NAME2,OPT1,OPT2,OPT3,
1054.          1 OPT4,OPT5,OPT6,NPTYPE,TXRT,STXRT,
1055.          2 PRTXRT,SATXRT,RITC,DISCR,IYRS,AFDCR,
1056.          4 P,XNSHRS,FLPR1,FLPR2,DIVRAT,
1057.          5 PERLT,PERST,PERCE,PERPS,RTNWLT,RTST,RTCEQ,RTNWPS,GINF,GINFX,
1058.          6 DR1,T,SY1,TSY1,SALPR1,
1059.          7 SY2,TSY2,DR2,DEPFAC,SALPR2,
1060.          8 CWPPE,RBAFDC,CONPRO,
1061.          9 ITYPE,NYRS)
1062. C
1063. C*****

```

```

1064. C*
1065. C*      SUBROUTINE READ
1066. C*      THIS SUBROUTINE READS IN DATA FROM THE
1067. C*      PRICING DATA FILE.
1068. C*
1069. C*      VARIABLES IN SUBROUTINE READ
1070. C*
1071. C*      NAME1 - PAGE HEADING
1072. C*      NAME2 - PAGE HEADING
1073. C*      NPTYPE - NUMBER OF NEW ASSET TYPES
1074. C*      CONSTRUCTION PERIOD
1075. C*      TXRT - FEDERAL STATUTORY TAX RATE
1076. C*      STXRT - STATE INCOME TAX RATE
1077. C*      PRTXRT - PROPERTY TAX RATE
1078. C*      SATXRT - SALES TAX RATE
1079. C*      RITC - INVESTMENT TAX CREDIT RATE
1080. C*      DISCR - DISCOUNT RATE
1081. C*      IYRS - ACTUAL YEAR (I.E. 1985,1990,ETC.)
1082. C*      AFDCR - AFUDC RATE
1083. C*      P - AVERAGE PRICE/SHARE OF STOCK
1084. C*      XNSHRS - NUMBER OF COMMON STOCK SHARES AT YEAR END
1085. C*      FLPR1 - % FLOW-THROUGH OF ADR AND ITC
1086. C*      FLPR2 - % FLOW-THROUGH OF DEBT PORTION OF AFUDC
1087. C*      DIVRAT - DIVIDEND PAYOUT RATIO
1088. C*      PERLT - % CAPITAL LONG-TERM DEBT
1089. C*      PERST - % CAPITAL SHORT-TERM DEBT
1090. C*      PERCE - % CAPITAL COMMON EQUITY
1091. C*      PERPS - % CAPITAL PREFERRED STOCK
1092. C*      RTNWL - COST OF NEW LONG-TERM DEBT
1093. C*      RTST - COST OF SHORT-TERM DEBT
1094. C*      RTCEQ - COST OF COMMON EQUITY
1095. C*      RTNWPS - COST OF NEW PREFERRED STOCK
1096. C*      GINF - GENERAL INFLATION RATE
1097. C*      GINF - CUMULATIVE GENERAL INFLATION INDEX
1098. C*      DR1,DR2 - DEPRECIATION RATE (FOR BOOK PURPOSES)
1099. C*      T - CURRENT YEAR OF SERVICE
1100. C*      SY1,SY2 - BOOK LIFE
1101. C*      TXY1,TXY2 - TAX LIFE (ADR METHOD)
1102. C*      SALPR1,SALPR2 - SALVAGE PERCENTAGE
1103. C*      DEPFAC - FOR DECLINING BALANCE DEPRECIATION - FACTOR
1104. C*      (E.G. IF 2 THEN DOUBLE DECLINING)
1105. C*      CWPPER - % CWIP IN RATE BASE FOR NEW PLANT
1106. C*      RBAFDC - 1 = AFUDC OFFSET
1107. C*      2 = AFUDC CAPITALIZED
1108. C*      CONPRO - CONSTRUCTION PROFILE (% EACH YEAR, FIRST ENTRY IS
1109. C*      YEAR PLANT COMES IN-SERVICE
1110. C*
1111. C*****
1112. C***SUBROUTINE COMENT IS CALLED EACH TIME THE DATA FILE IS TO BE READ.
1113. C***ITS PURPOSE IS TO STRIP OFF ANY COMMENTS FROM THE DATA FILE.
1114. C*****
1115. C      DIMENSION NAME1(6),NAME2(6),IYRS(50),
1116. C      1 P(50),XNSHRS(50),
1117. C      3 DIVRAT(50),PERLT(50),PERST(50),PERCE(50),
1118. C      4 PERPS(50),RTNWL(50),RTST(50),RTCEQ(50),RTNWPS(50),GINF(50),
1119. C      5 GINF(70),DR1(15),T(15),

```

```

1120.      6 SY1(15),TXSY1(15),SALPR1(15),
1121.      7 SY2(20),TXSY2(20),DR2(20),DEPFAC(20),
1122.      8 SALPR2(20),
1123.      9 CWPPER(300),
1124.      1 RBAFDC(300),CONPRO(15,20)
1125.      C
1126.      CALL COMENT(NN)
1127.      READ(NN,435) NAME1
1128.      CALL COMENT(NN)
1129.      READ(NN,435) NAME2
1130.      435 FORMAT(6A4)
1131.      CALL COMENT(NN)
1132.      READ(NN,*) OPT1,OPT2,OPT3,OPT4,OPT5,OPT6,DISCR,IYRS(2)
1133.      WRITE(4,9002)
1134.      9002 FORMAT(' PLACEC')
1135.      CALL COMENT(NN)
1136.      READ(NN,*) CWPPER(1),RBAFDC(1),AFDCR,FLPR1,FLPR2
1137.      CALL COMENT(NN)
1138.      READ(NN,*) TXRT,STXRT,PTXRT,SATXRT,RITC
1139.      CALL COMENT(NN)
1140.      READ(NN,*) P(1),XNSHRS(1)
1141.      WRITE(4,8900)
1142.      8900 FORMAT(' PLACED')
1143.      CALL COMENT(NN)
1144.      READ(NN,*) (DIVRAT(I),I=2,NYRS)
1145.      CALL COMENT(NN)
1146.      READ(NN,*) (PERLT(I),I=2,NYRS)
1147.      CALL COMENT(NN)
1148.      READ(NN,*) (PERST(I),I=2,NYRS)
1149.      CALL COMENT(NN)
1150.      READ(NN,*) (PERCE(I),I=2,NYRS)
1151.      CALL COMENT(NN)
1152.      READ(NN,*) (PERPS(I),I=2,NYRS)
1153.      CALL COMENT(NN)
1154.      READ(NN,*) (RTNWLT(I),I=2,NYRS)
1155.      CALL COMENT(NN)
1156.      READ(NN,*) (RTST(I),I=2,NYRS)
1157.      CALL COMENT(NN)
1158.      READ(NN,*) (RTCEQ(I),I=2,NYRS)
1159.      CALL COMENT(NN)
1160.      READ(NN,*) (RTNWPS(I),I=2,NYRS)
1161.      CALL COMENT(NN)
1162.      READ(NN,*) (GINF(I),I=2,NYRS)
1163.      GINF(1)=1.0
1164.      DO 50 I=2,NYRS
1165.      50 GINF(I)=GINF(I-1)*(1.0+GINF(I))
1166.      NYRS1=NYRS+1
1167.      NYRS2=NYRS+20
1168.      DO 55 I=NYRS1,NYRS2
1169.      55 GINF(I)=GINF(I-1)*(1.0+GINF(NYRS))
1170.      WRITE(4,9004)
1171.      9004 FORMAT(' PLACEE')
1172.      DO 100 J=1,10
1173.      CALL COMENT(NN)
1174.      100 READ(NN,*) DR1(J),T(J),SY1(J),TXSY1(J),SALPR1(J)
1175.      DO 110 J=1,NPTYPE

```

```

1176.      CALL COMENT(NN)
1177. 110 READ(NN,*) SY2(J),TXSY2(J),DR2(J),DEPFAC(J),SALPR2(J)
1178.      DO 10 J=1,NPTYPE
1179.      CALL COMENT(NN)
1180.      READ(NN,*) (CONPRO(K,J),K=1,15)
1181. 10 CONTINUE
1182.      DO 9 I=3,NYRS
1183.      IYRS(I)=IYRS(I-1)+1
1184. 9 CONTINUE
1185. C*** WRITE OUT KEY INPUT ASSUMPTIONS.
1186.      WRITE(4,200) AFDCR,FLPR1,FLPR2,CWPPER(1)
1187. 200 FORMAT(' **KEY PARAMETER INPUTS**/' AFUDC RATE:',F6.4,/' FLOW',
1188.      * 'THROUGH % OF ADR:',F6.4,/' FLOW THROUGH % OF AFUDC TAX SAVINGS:',
1189.      * F6.4,/' % OF CWIP IN RATE BASE:',F6.4)
1190.      RETURN
1191.      END
1192.      SUBROUTINE SUPPLY(NNN,NPLNTS,JYR,JPLANT,PLTCAP,CAPCST,
1193.      1 NYRS,GWHTOT,GWNUC,AVFLCS,FCNUC,GINFX,AVGOM,
1194.      2 SERPER,CWPPER,RBAFDC,TOTSAL,NIT,IFCRYR,IFCRBD,OPT2,OPT8)
1195. C
1196. C*****
1197. C*
1198. C*      SUBROUTINE SUPPLY
1199. C*      THIS SUBROUTINE PUTS OUTPUTS FROM THE ICF SUPPLY MODEL *
1200. C*      INTO THE APPROPRIATE FINANCIAL MODEL VARIABLES
1201. C*      UNITS FOR INPUTS
1202. C*      GENERATION          GWH
1203. C*      PEAK                MW
1204. C*      TOTAL CAPA         MW
1205. C*      NEW CAPACITY       MW
1206. C*      CAPACITY COST      $/KW - NOMINAL $
1207. C*      FUEL COSTS         MILLS/KWH - NOMINAL $
1208. C*
1209. C*      XSALES = BASE YEAR SALES (SCH 9,SCH 4)
1210. C*      XGENER = BASE YEAR GENERATION (SCH 12,SCH 7)
1211. C*
1212. C*****
1213. C
1214. C
1215.      DIMENSION JYR(300),JPLANT(300),PLTCAP(3,300),CAPCST(300),
1216.      1 GWHTOT(3,50),GWNUC(3,50),PEAK(50),AVFLCS(50),FCNUC(50),
1217.      2 TOTSAL(3,50),GINFX(70),AVGOM(50),CUST(50),GCUST(50),
1218.      3 RATIMP(50),CWPPER(300),SERPER(300),RBAFDC(300)
1219. C***READ IN BASE YEAR GENERATION FOR UTILITY.
1220.      NNN=12
1221.      READ(NNN,*) GWHTOT(1,1)
1222.      IF (OPT8.EQ. 2) GO TO 100
1223. C***READ IN BUILDS AND CAPACITY COSTS. THE COSTS READ IN ARE NOMINAL
1224. C***DOLLARS.
1225.      NPLNTS=0
1226.      CALL COMENT(NNN)
1227.      DO 2050 K=1,300
1228.      READ(NNN,*) JYR(K),JPLANT(K),PLTCAP(1,K),CAPCST(K)
1229.      IF (JYR(K).EQ.0) GO TO 2057
1230.      NPLNTS=NPLNTS+1
1231. 2050 CONTINUE

```

```

1232. 2057 WRITE(4,2064) NPLNTS
1233. 2064 FORMAT(1X,'NUMBER OF NEW BUILDS:',2X,I5)
1234. C***NOW READ IN GENERATION, TOTAL AND NUCLEAR, FOLLOWED BY PEAK DEMAND
1235. WRITE(4,15)
1236. 15 FORMAT(' TOTAL GENERATION',5X,'NUC GENERATION',10X,'PEAK')
1237. CALL COMENT(NNNN)
1238. DO 2051 I=2,NYRS
1239. READ(NNNN,*) GWHTOT(1,I),GWHNUC(1,I),PEAK(I)
1240. WRITE(4,2059) GWHTOT(1,I),GWHNUC(1,I),PEAK(I)
1241. 2059 FORMAT(2X,F14.3,6X,F14.3,5X,F14.3)
1242. 2051 CONTINUE
1243. C
1244. C***NOW READ AVERAGE FUEL AND NUCLEAR FUEL COSTS PER KWH. THESE COSTS
1245. C***ARE IN NOMONAL MILLS.
1246. CALL COMENT(NNNN)
1247. DO 2054 I=2,NYRS
1248. READ(NNNN,*) AVFLCS(I),FCNUC(I)
1249. 2054 CONTINUE
1250. C
1251. C***READ IN LOSS FACTOR. ADJUST GENERATION BY LOSS FACTOR TO GET SALES
1252. C
1253. CALL COMENT(NNNN)
1254. READ(NNNN,*) ALOSS
1255. DO 551 I=2,NYRS
1256. TOTSAL(1,I)=(1.0-ALOSS)*GWHTOT(1,I)
1257. 551 CONTINUE
1258. C***INPUT PLANT SPECIFIC CWIP RATE TREATMENT, IF DESIRED. THIS IS READ
1259. C***FROM UNIT 16.
1260. C* SERPER = % OF YEAR NEW PLANT IN-SERVICE FOR 1ST YEAR *
1261. SERPER(1)=1.0
1262. IF (OPT2 .NE. 1) GO TO 5
1263. DO 80 ICOUNT=1,NPLNTS
1264. 80 READ(16,*) SERPER(ICOUNT),CWPPER(ICOUNT),RBAFDC(ICOUNT)
1265. ICOUNT=NPLNTS+1
1266. 5 NN=10
1267. C***NOW CALCULATE FIXED O AND M EXPENSES NOT CAPTURED IN THE SUPPLY MODE
1268. C***FUEL COSTS (FUEL COSTS ONLY INCLUDE VARIABLE O AND M). THE INPUT IS
1269. C***ADJUSTED FOR INFLATION. DATA IS READ FROM USER FILE.
1270. CALL COMENT(NN)
1271. READ(NN,*) (AVGOM(I),I=2,NYRS)
1272. DO 10 I=2,NYRS
1273. 10 AVGOM(I)=AVGOM(I)*GINFX(I)
1274. C***NOW T&D BUILDS ARE CALCULATED FROM INPUTS. TRANSMISSION BUILDS ARE
1275. C***A FUNCTION OF GROWTH IN PEAK DEMAND. DISTRIBUTION AR A FUNCTION OF
1276. C***OF GROWTH IN NUMBER OF CUSTOMERS AND PEAK DEMAND.
1277. C*** CUST - TOTAL NUMBER OF CUSTOMERS
1278. C*** GWTRA - TRANSMISSION BUILDS PER GW COST
1279. C*** CUSDIS - DISTRIBUTION BUILDS PER CUSTOMER COST
1280. C*** GWDIS - DISTRIBUTION BUILDS PER GW COST
1281. C*** GCUST - GROWTH RATE IN NUMBER OF CUSTOMERS
1282. CALL COMENT(NN)
1283. READ(NN,*) PEAK(1),GWTRA,CUSDIS,GWDIS
1284. CALL COMENT(NN)
1285. READ(NN,*) CUST(1),(GCUST(I),I=2,NYRS)
1286. DO 20 I=2,NYRS
1287. 20 CUST(I)=CUST(I-1)*(1.0+GCUST(I))

```

```

1288.      IF (GWTRA .EQ. 0.0) GO TO 40
1289.      DO 30 I=2,NYRS
1290.          NPLNTS=NPLNTS+1
1291.          CAPCST(NPLNTS)=1.0
1292.          PLTCAP(1,NPLNTS)=(PEAK(I)-PEAK(I-1))*GWTRA*GINFX(I)
1293.          JYR(NPLNTS)=I
1294.          JPLANT(NPLNTS)=13
1295.      30  CONTINUE
1296.      C***INPUT TRANSMISSION CHIP RATE TREATMENT, IF NECESSARY
1297.          IF (OPT2 .NE. 1.0) GO TO 40
1298.          READ(16,*) SER,CWP,RBA
1299.          DO 35 I=ICOUNT,NPLNTS
1300.              SERPER(I)=SER
1301.              CWPPI(I)=CWP
1302.      35  RBAFDC(I)=RBA
1303.          ICOUNT=NPLNTS+1
1304.      40  IF ((GWDIS .EQ. 0.0).AND.(CUSDIS .EQ.0.0)) GO TO 60
1305.          DO 50 I=2,NYRS
1306.              NPLNTS=NPLNTS+1
1307.              CAPCST(NPLNTS)=1.0
1308.              PLTCAP(1,NPLNTS)=((PEAK(I)-PEAK(I-1))*GWDIS+
1309.              *              (CUST(I)-CUST(I-1))*CUSDIS/1000.0)*GINFX(I)
1310.              JYR(NPLNTS)=I
1311.              JPLANT(NPLNTS)=14
1312.      50  CONTINUE
1313.      C***INPUT DISTRIBUTION CHIP RATE TREATMENT, IF NECESSARY
1314.          IF (OPT2 .NE. 1.0) GO TO 60
1315.          READ(16,*) SER,CWP,RBA
1316.          DO 55 I=ICOUNT,NPLNTS
1317.              SERPER(I)=SER
1318.              CWPPI(I)=CWP
1319.      55  RBAFDC(I)=RBA
1320.          ICOUNT=NPLNTS+1
1321.      C***NOW METER COSTS ARE CALCULATED.
1322.      C***      USIME1 - METER COSTS ($/METER)
1323.      C***      RATIMP - IMPLEMENTATION RATE FOR NEW METERS
1324.      60  RATIMP(1)=0.0
1325.          CALL COMENT(NN)
1326.          READ(NN,*) CSTMET,(RATIMP(I),I=2,NYRS)
1327.          IF (CSTMET .EQ.0.0) GO TO 90
1328.          DO 70 I=2,NYRS
1329.              NPLNTS=NPLNTS+1
1330.              CAPCST(NPLNTS)=1.0
1331.              PLTCAP(1,NPLNTS)=(RATIMP(I)*CUST(I)-RATIMP(I-1)*CUST(I-1))
1332.              *              *CSTMET*GINFX(I)/1000.0
1333.              JYR(NPLNTS)=I
1334.              JPLANT(NPLNTS)=15
1335.      70  CONTINUE
1336.      C***INPUT METER CHIP RATE TREATMENT, IF NECESSARY
1337.          IF (OPT2 .NE. 1.0) GO TO 90
1338.          READ(16,*) SER,CWP,RBA
1339.          DO 75 I=ICOUNT,NPLNTS
1340.              SERPER(I)=SER
1341.              CWPPI(I)=CWP
1342.      75  RBAFDC(I)=RBA
1343.      90  CONTINUE

```

```

1344.      RETURN
1345.      C*****
1346.      C***THIS SECTION IS CALLED WHEN FIXED CHARGE RATES
1347.      C***ARE TO BE CALCULATED. IT SETS UP THE NECESSARY BUILD.
1348.      C*****
1349.      C*
1350.      C*      IFCRBD = THE PLANT TYPE FOR WHICH THE FIXED CHARGE RATE
1351.      C*      IS TO BE CALCULATED.
1352.      C*      IFCRYR = THE YEAR THAT THE BUILD IS TO COME ON LINE
1353.      C*
1354.      C***IF NEC, FIRST READ IFCRYR AND IFCRBD (OR USE VALUES ALREADY READ)
1355.      100 IF (NIT .NE. 1) GO TO 110
1356.      CALL COMENT(9)
1357.      READ(9,*) IFCRYR,IFCRBD
1358.      C***NOW SET UP BUILD FOR FCR ANALYSIS.
1359.      110 NPLNTS=1
1360.      JYR(1)=IFCRYR
1361.      JPLANT(1)=IFCRBD
1362.      CAPCST(1)=1000.0
1363.      PLTCAP(1,1)=1000.0
1364.      GWHTOT(1,1)=0.0
1365.      RETURN
1366.      END
1367.      SUBROUTINE ZEROS1(Y1,Y2,Y3)
1368.      C
1369.      C*****
1370.      C*
1371.      C*      SUBROUTINE ZEROS1
1372.      C*      THIS SUBROUTINE ZEROS OUT ALL VARIABLES
1373.      C*      IN THE FIRST FOUR COMMON BLOCKS IN THE MAIN PRO--*
1374.      C*      GRAM. IT IS CALLED FOR EACH OWNERSHIP ITYPE.
1375.      C*
1376.      C*****
1377.      C
1378.      C      DIMENSION Y1(4017),Y2(2670),Y3(2080)
1379.      C
1380.      C      DO 5000 I=1,4017
1381.      C          Y1(I)=Y1(I)-Y1(I)
1382.      5000 CONTINUE
1383.      C
1384.      C      DO 5002 I=1,2670
1385.      C          Y2(I)=Y2(I)-Y2(I)
1386.      5002 CONTINUE
1387.      C
1388.      C      DO 5003 I=1,2080
1389.      C          Y3(I)=Y3(I)-Y3(I)
1390.      5003 CONTINUE
1391.      C
1392.      C      RETURN
1393.      C      END
1394.      C      SUBROUTINE ZEROS2(Y4,Y5)
1395.      C
1396.      C
1397.      C*****
1398.      C*
1399.      C*      SUBROUTINE ZEROS2

```



```
1400. C*          THIS SUBROUTINE ZEROS OUT ALL VARIABLES      *
1401. C*          IN THE LAST THREE COMMON BLOCKS OF THE MAIN PRO-*
1402. C*          GRAM. IT IS CALLED ONCE FOR EACH REGION.
1403. C*
1404. C*****
1405. C
1406.          DIMENSION Y4(12450),Y5(4253)
1407. C
1408.          DO 5019 I=1,12450
1409.              Y4(I)=Y4(I)-Y4(I)
1410.          5019 CONTINUE
1411. C
1412.          DO 5020 I=1,4253
1413.              Y5(I)=Y5(I)-Y5(I)
1414.          5020 CONTINUE
1415. C
1416.          RETURN
1417.          END
1418.          //LKED,SYSMOD DD DSN=CN6376.MI2.NUFS.RPT.LOAD(SUBS),UNIT=DASD,
1419.          // DISP=(,CATLG),SPACE=(TRK,(10,1,1),RLSE)
```

?

APPENDIX A

This appendix relates to the software which reads and aggregates data from the Form 1 and Form 1M data tapes. This process is discussed in Section II of Volume II. This appendix is divided into two sections.

i) A listing of the computer code for NUFS.FORM1.FORT. This includes the input file referred to in Volume II as the Form 1 state code file. This file is input as FTO9FOO1 and contains information regarding which utilities are in each region and which financial accounts are to be aggregated.

The program consists of a main routine and one subroutine entitled FORM1. The main subroutine reads in the data tape and aggregates the financial accounts according to the instructions from the state code file. Subroutine FORM1 then calculates ratios and creates the files containing results for NUFS. The remainder of this module contains a series of job steps which create a partitioned data set whose members are the data sets for NUFS.

ii) A listing of the computer code for NUFS.FORM1M.FORT. This module also contains the Form 1M state code file. Its structure is very similar to NUFS.FORM1.FORT.

i) LISTING OF NUFFS.FORM1.FORT

```

1. //MI2UFORM JOB (6376,D02,,999),'FORM1',TIME=(1,05),MSGCLASS=A
2. //FPC EXEC FORTGCG,REGION=400K
3. //FORT.SYSIN DD *
4.     DIMENSION ISTATE(60),INFO(300)
5.     DOUBLE PRECISION AMNT,AMNT1,AMOUNT(11,300),AMNT2(300)
6.     REAL*4 ACODE,JTYP,KTYP,LTYF,MTYP,NTYP,OTYP,PTYP,QTYP,
7.     * RTYP,RBKTYP,NOTYP,N1TYP,N2TYP,N3TYP,N4TYP,N5TYP,N6TYP,
8.     * N7TYP,N8TYP,N9TYP
9.     DATA JTYP/'J'/
10.    DATA KTYP/'K'/
11.    DATA LTYP/'L'/
12.    DATA MTYP/'M'/
13.    DATA NTYP/'N'/
14.    DATA OTYP/'O'/
15.    DATA PTYP/'P'/
16.    DATA QTYP/'Q'/
17.    DATA RTYP/'R'/
18.    C DATA RBKTYP/'/'/
19.    DATA NOTYP/'O'/
20.    DATA N1TYP/'1'/
21.    DATA N2TYP/'2'/
22.    DATA N3TYP/'3'/
23.    DATA N4TYP/'4'/
24.    DATA N5TYP/'5'/
25.    DATA N6TYP/'6'/
26.    DATA N7TYP/'7'/
27.    DATA N8TYP/'8'/
28.    DATA N9TYP/'9'/
29.    C***NUFS.FORM1.FORT. THIS PROGRAM READS DATA FROM THE FORM 1 TAPE AND
30.    C***WRITES OUT DATA NEEDED FOR THE NATIONAL UTILITY FINANCIAL STATEMENT
31.    C***MODEL (NUFS).
32.    C***
33.    C***WRITTEN BY MARK R. INGLIS, ICF INC. JULY 31,1980
34.    C***
35.    C***FIRST, A USER SUPPLIED INPUT FILE IS READ. THIS FILE CONTAINS STATE
36.    C***AND ACCOUNT CODES.
37.    C***** NSTATE=# OF STATES
38.    C***** NINFO= # OF ACCOUNTS TO BE ADDED OVER REGIONS
39.    C***** ISTATE(I)=DOE REGION THAT STATE I IS IN
40.    C***** INFO(I)= 6 DIGIT ACCOUNT CODE
41.    READ(9,*)NSTATE
42.    DO 77 J=1,NSTATE
43. 77 READ(9,*) I,ISTATE(I)
44.    READ(9,*) NINFO
45.    DO 88 I=1,NINFO
46. 88 READ(9,802) INFO(I)
47.    802 FORMAT(I6)
48.    C***INITIALIZE AMOUNT, THE MATRIX WHICH CONTAINS TOTALS FOR EACH REGION
49.    C***IE., AMOUNT(J,I)=AMOUNT OF ACCOUNT I SUMMED OVER ALL UTILITIES IN
50.    C***REGION J. (REGION 11 = NATION)
51.    DO 91 J=1,11
52.    DO 91 I=1,300
53.    91 AMOUNT(J,I)=0.0
54.    C***NOW THE FORM 1 TAPE IS READ. EACH RECORD CONTAINS THE TOTAL FOR
55.    C***FOR THE GIVEN ACCOUNT FOR THE GIVEN UTILITY. IF THE RECORD CONTAINS

```

```

56. C***AN ACCOUNT WHICH WAS READ INTO INFO(I), THEN THE TOTAL IS ADDED IN
57. C***TO THE PROPER REGION. IF NOT, THE NEXT RECORD IS READ.
58. C***** JSTATE = STATE CODE
59. C*****JUTIL=UTILITY CODE
60. C*****JINFO = ACCOUNT CODE
61. C
62. 3020 READ(8,1001,END=3030) JSTATE,JUTIL,JINFO,JAMNT1,JAMNT,ACODE
63. 1001 FORMAT(I2,I4,I6,1X,I6,I5,A1)
64. 50 DO 60 I=1,NINFO
65. IF(JINFO.EQ.INFO(I)) GO TO 70
66. 60 CONTINUE
67. GO TO 30
68. C***IF A STATES REGION IS 12, THIS MEANS THE STATE IS NOT IN A DOE REGION.
69. C***IN THIS CASE, THE NEXT RECORD IS READ.(EXAMPLES: PUERTO RICO, ETC.)
70. 70 IF (ISTATE(JSTATE) .EQ. 12) GO TO 3020
71. C***THE TOTAL FOR THE ACCOUNT IS WRITTEN IN 'CODE', SOMETIMES CONTAINING
72. C***LETTERS. THIS NEXT SECTION CONVERTS THE ACCOUNT INTO A REAL NUMBER.
73. C
74. IF(ACODE.EQ.JTYP) GO TO 100
75. IF(ACODE.EQ.KTYP) GO TO 101
76. IF(ACODE.EQ.LTYP) GO TO 102
77. IF(ACODE.EQ.MTYP) GO TO 103
78. IF(ACODE.EQ.NTYP) GO TO 104
79. IF(ACODE.EQ.OTYP) GO TO 105
80. IF(ACODE.EQ.PTYP) GO TO 106
81. IF(ACODE.EQ.QTYP) GO TO 107
82. IF(ACODE.EQ.RTYP) GO TO 108
83. C IF(ACODE.EQ.RBKTYT) GO TO 109
84. IF(ACODE.EQ.NOTYP) GO TO 110
85. IF(ACODE.EQ.N1TYP) GO TO 111
86. IF(ACODE.EQ.N2TYP) GO TO 112
87. IF(ACODE.EQ.N3TYP) GO TO 113
88. IF(ACODE.EQ.N4TYP) GO TO 114
89. IF(ACODE.EQ.N5TYP) GO TO 115
90. IF(ACODE.EQ.N6TYP) GO TO 116
91. IF(ACODE.EQ.N7TYP) GO TO 117
92. IF(ACODE.EQ.N8TYP) GO TO 118
93. IF(ACODE.EQ.N9TYP) GO TO 119
94. GO TO 109
95. C
96. 100 JAMNT=-10*JAMNT-1
97. GO TO 900
98. 101 JAMNT=-10*JAMNT-2
99. GO TO 900
100. 102 JAMNT=-10*JAMNT-3
101. GO TO 900
102. 103 JAMNT=-10*JAMNT-4
103. GO TO 900
104. 104 JAMNT=-10*JAMNT-5
105. GO TO 900
106. 105 JAMNT=-10*JAMNT-6
107. GO TO 900
108. 106 JAMNT=-10*JAMNT-7
109. GO TO 900
110. 107 JAMNT=-10*JAMNT-8
111. GO TO 900

```

```

112. 108 JAMNT=-10*JAMNT-9
113.    GO TO 900
114. 109 JAMNT=-10*JAMNT
115.    GO TO 900
116. 110 JAMNT=10*JAMNT
117.    GO TO 900
118. 111 JAMNT=10*JAMNT+1
119.    GO TO 900
120. 112 JAMNT=10*JAMNT+2
121.    GO TO 900
122. 113 JAMNT=10*JAMNT+3
123.    GO TO 900
124. 114 JAMNT=10*JAMNT+4
125.    GO TO 900
126. 115 JAMNT=10*JAMNT+5
127.    GO TO 900
128. 116 JAMNT=10*JAMNT+6
129.    GO TO 900
130. 117 JAMNT=10*JAMNT+7
131.    GO TO 900
132. 118 JAMNT=10*JAMNT+8
133.    GO TO 900
134. 119 JAMNT=10*JAMNT+9
135.    GO TO 900
136.  C
137. 900 AMNT=FLOAT(JAMNT)
138.    IF(JAMNT1.EQ.0) GO TO 6010
139.    AMNT1=FLOAT(JAMNT1)
140.    AMNT=AMNT+AMNT1*10.0**6
141. 6010 CONTINUE
142. C***THE CONVERTED AMOUNT IS NOW ADDED INTO THE PROPER REGION, AND THE
143. C***NEXT RECORD IS READ
144.    AMOUNT(ISTATE(JSTATE),I)=AMNT+AMOUNT(ISTATE(JSTATE),I)
145.    AMOUNT(11,I)=AMNT+AMOUNT(11,I)
146. 30 CONTINUE
147.    GO TO 3020
148. 3030 CONTINUE
149.  C
150. C***AFTER THE WHOLE TAPE HAS BEEN READ, SUBROUTINE FORM1 IS CALLED FOR
151. C***EACH REGION. THIS SUBROUTINE WRITES OUT THE DATA NECESSARY FOR THE
152. C***NUFS MODEL.
153.    DO 950 IREG=1,11
154.    DO 940 I=1,NINFO
155. 940 AMNT2(I)=AMOUNT(IREG,I)
156.    CALL FORM1(IREG,AMNT2)
157. 950 CONTINUE
158. 5000 STOP
159.    END
160.    SUBROUTINE FORM1(IREG,AMOUNT)
161.  C
162. C*****
163. C*
164. C* SUBROUTINE FORM1
165. C* THIS SUBROUTINE TAKES THE TOTALS FROM THE MAIN
166. C* PROGRAM FOR EACH REGION AND WRITES OUT FILES CON-
167. C* TAINING THE DATA NEEDED BY NUFFS. IT ALSO PRINTS *

```

```

168. C*          OUT VARIOUS QUANTITIES OF INTEREST TO UNIT 6.  *
169. C*          (UNIT 6=LINE PRINTER)                          *
170. C*****
171. C
172. C
173.          DOUBLE PRECISION FM1BS(83),FM1IS(56),FM1SU(47)
174.          DOUBLE PRECISION FM1EPS(10),FM1DEP(11),FM1FUL(3)
175.          DOUBLE PRECISION FM1PIS(2),AMOUNT(300)
176.          DOUBLE PRECISION TEMP5,TEMP6,EPPOR
177.          DIMENSION ZA1(10)
178. C***FIRST, PUT DATA INTO MATRICES BY SCHEDULE:
179. C***          FM1BS=BALANCE SHEET
180. C***          FM1IS=INCOME SHEET
181. C***          FM1SU=SOURCES AND USES
182. C***          FM1EPS=ELECTRIC PLANT IN SERVICE
183. C***          ETC.
184.          ICOUNT=0
185.          DO 100 I=1,83
186.          ICOUNT=ICOUNT+1
187. 100    FM1BS(I)=AMOUNT(ICOUNT)/1000000.
188.          DO 110 I=1,56
189.          ICOUNT=ICOUNT+1
190. 110    FM1IS(I)=AMOUNT(ICOUNT)/1000000.
191.          DO 120 I=1,47
192.          ICOUNT=ICOUNT+1
193. 120    FM1SU(I)=AMOUNT(ICOUNT)/1000000.
194.          DO 130 I=1,10
195.          ICOUNT=ICOUNT+1
196. 130    FM1EPS(I)=AMOUNT(ICOUNT)/1000000.
197.          DO 140 I=1,11
198.          ICOUNT=ICOUNT+1
199. 140    FM1DEP(I)=AMOUNT(ICOUNT)/1000000.
200.          DO 150 I=1,3
201.          ICOUNT=ICOUNT+1
202. 150    FM1FUL(I)=AMOUNT(ICOUNT)/1000000.
203.          DO 160 I=1,2
204.          ICOUNT=ICOUNT+1
205. 160    FM1PIS(I)=AMOUNT(ICOUNT)/1000000.
206.          ICOUNT=ICOUNT+1
207.          SALES=AMOUNT(ICOUNT)/1.0E09
208.          ICOUNT=ICOUNT+1
209.          GENER=AMOUNT(ICOUNT)/1.0E09
210. C
211.          WRITE(6,11)IREG
212. 11    FORMAT(' THE TOTALS FOR REGION ',I2,':')
213. C
214. C*****NOW STRIP OFF RELEVANT AMOUNTS FOR NUF5 AND INTEREST. (IE. THIS
215. C*****ROUTINE CAN BE MODIFIED TO PRINT OUT ANY ACCOUNT.)
216. C
217.          WRITE(6,311)
218. 311    FORMAT(' THE VALUES FOR FM1EPS ARE:')
219.          DO 9011 J=1,10
220. 9011    WRITE(6,2015)FM1EPS(J)
221. 2015    FORMAT(5X,F10.2)
222.          ZA1(1)=FM1EPS(1)
223.          ZA1(2)=FM1EPS(2)

```

```

224.      ZA1(3)=FM1EPS(3)
225.      ZA1(4)=FM1EPS(4)
226.      ZA1(5)=0.0
227.      ZA1(6)=FM1EPS(5)
228.      ZA1(7)=FM1EPS(7)
229.      ZA1(8)=FM1EPS(8)
230.      ZA1(9)=FM1EPS(9)
231.      ZA1(10)=0.0
232.      WRITE(6,2014)
233. 2014  FORMAT(' THE VALUES FOR ZA(I) ARE')
234.      DO 9015 J=1,10
235. 9015  WRITE(6,2015) ZA1(J)
236.      C
237.  C***** FIND RELATIVE ERROR FOR ASSET TOTALS.
238.      C          RE1-RE3 = RELATIVE ERRORS
239.      C
240.      TEMP5=0.0
241.      DO 9007 J=1,5
242. 9007  TEMP5=TEMP5+FM1EPS(J)
243.      C
244.      RE1=(FM1EPS(6)-TEMP5)/FM1EPS(6)
245.      WRITE(6,2000)RE1
246. 2000  FORMAT(' R.E. FOR FM1EPS(6) IS: ',F10.8)
247.      C
248.      TEMP6=0.0
249.      DO 9008 J=7,9
250. 9008  TEMP6=TEMP6+FM1EPS(J)
251.      TEMP6=TEMP6+TEMP5
252.      C
253.      RE2=(FM1EPS(10)-TEMP6)/FM1EPS(10)
254.      WRITE(6,2001)RE2
255. 2001  FORMAT(' R.E. FOR FM1EPS(10) IS: ',F10.8)
256.      C
257.      RE3=(FM1PIS(2)-FM1EPS(10))/FM1PIS(2)
258.      WRITE(6,2018)RE3
259. 2018  FORMAT(' R.E. FOR FM1PIS(2) & FM1EPS(10) IS: ',F10.8)
260.      C
261.  C***** COST OF EMBEDDED DEBT
262.      C
263.      RTBSBN=FM1IS(39)/FM1BS(57)
264.      WRITE(6,2002)RTBSBN
265. 2002  FORMAT(' RTBSBN= ',F10.4)
266.      C
267.  C***** GENERAL TAX RATE
268.  C*** (NOTE: THIS INCLUDES STATE INCOME TAXES FOR NUF5 PURPOSES)
269.      GTXRT=(FM1IS(9)+FM1IS(11))/FM1IS(1)
270.      WRITE(6,2003)GTXRT
271. 2003  FORMAT(' GTXRT= ',F10.4)
272.      C
273.  C***** ELECTRIC PORTION OF TOTAL PLANT
274.  C*** (NOTE: MANY OF THESE ACCOUNTS INCLUDE GAS UTILITIES. THIS IS
275.  C*** NETTED OUT BY MULTIPLYING BY THIS RATIO.)
276.      EPPOR=FM1EPS(10)/(FM1BS(1)+FM1BS(17))
277.      WRITE(6,2017)EPPOR
278. 2017  FORMAT(' EPPOR= ',D10.2)
279.      C

```



```

280. C***** RETAINED EARNINGS
281.     RE=FM1BS(53)
282.     WRITE(6,2004)RE
283. 2004 FORMAT(' RETAINED EARNINGS= ',F10.2)
284.     RE=RE*EPPOR
285. C
286. C***** STOCK OF OUTSTANDING BONDS PRE-BASE YEAR
287. C
288.     OB=FM1BS(62)
289.     WRITE(6,2005)OB
290. 2005 FORMAT(' OUTSTANDING BONDS= ',F10.2)
291.     OB=OB*EPPOR
292. C*****SHORT TERM DEBT PRE-BASE YEAR.
293.     STDEBT=FM1BS(63)*EPPOR
294.     WRITE(6,2050) STDEBT
295. 2050 FORMAT(' SHORT TERM DEBT= ',F10.2)
296. C
297. C***** REVENUES PRE-BASE YEAR
298. C
299.     REVS=FM1IS(1)
300.     WRITE(6,2006) REVS
301. 2006 FORMAT(' ELECTRIC REVENUES= ',F10.2)
302. C
303. C***** CAPITAL STRUCTURE
304. C
305.     TEMP7=FM1BS(56)+FM1BS(62)+FM1BS(63)
306.     W1=FM1BS(62)/TEMP7
307.     WRITE(6,2007) W1
308. 2007 FORMAT(' W1= ',F10.4)
309.     W2=FM1BS(63)/TEMP7
310.     WRITE(6,2008) W2
311. 2008 FORMAT(' W2= ',F10.4)
312.     W3=(FM1BS(56)-FM1BS(45))/TEMP7
313.     WRITE(6,2009) W3
314. 2009 FORMAT(' W3= ',F10.4)
315.     W4=FM1BS(45)/TEMP7
316.     WRITE(6,2010) W4
317. 2010 FORMAT(' W4= ',F10.4)
318. C
319. C***** TOTAL ACCUMULATED DEFERRED ITC
320. C
321.     TDITC=FM1BS(76)
322.     WRITE(6,2011)TDITC
323. 2011 FORMAT(' TOTAL DEFERED ITC= ',F10.2)
324.     TDITC=TDITC*EPPOR
325. C
326. C***** ACCUMULATED DEFERRED I/T
327.     PRODEF=FM1BS(80)
328.     WRITE(6,2012)PRODEF
329. 2012 FORMAT(' PROVISION FOR DEFERED INCOME TAXES= ',F10.2)
330.     PRODEF=PRODEF*EPPOR
331. C
332. C***** FUEL & O&M COSTS ADJUSTED FOR TAXES
333.     TXFUEL=(FM1IS(2)+FM1IS(3))
334.     WRITE(6,2013)TXFUEL
335. 2013 FORMAT(' FUEL PLUS O & M= ',F10.2)

```

```

336.      TXFUEL=TXFUEL/(1-GTXRT)
337.      C
338.      C***** DIVIDEND PAYOUT RATIO
339.      C*** (NOTE: THIS IS % OF EARNINGS AVAILABLE TO COMMON, NOT NET INCOME)
340.      DIVRAT=FM1IS(56)/(FM1IS(54)-FM1IS(55))
341.      WRITE(6,2023) DIVRAT
342.      2023 FORMAT(' DIVIDEND PAYOUT RATIO= ',F10.4)
343.      C
344.      C***** WORKING CAPITAL
345.      C
346.      WC=(FM1BS(31)-FM1BS(74))
347.      WRITE(6,2021) WC
348.      2021 FORMAT(' WORKING CAPITAL= ',F10.4)
349.      WC=WC*EPPOR
350.      C
351.      C***** COMMON STOCK PRE-BASE YEAR
352.      C
353.      COMSTK=(FM1BS(56)-FM1BS(53)-FM1BS(45))
354.      WRITE(6,2022) COMSTK
355.      2022 FORMAT(' COMMON STOCK= ',F10.4)
356.      COMSTK=COMSTK*EPPOR
357.      C
358.      C***** PREFERRED STOCK PRE-BASE YEAR
359.      C
360.      PRESTK=FM1BS(45)
361.      WRITE(6,2024) PRESTK
362.      2024 FORMAT(' PREFERRED STOCK= ',F10.4)
363.      PRESTK=PRESTK*EPPOR
364.      C
365.      C***** RETURN ON PREFERRED EQUITY
366.      C
367.      RTPEQ=FM1IS(55)/FM1BS(45)
368.      WRITE(6,2020) RTPEQ
369.      2020 FORMAT(' RETURN ON PREFERRED EQUITY= ',F10.4)
370.      C
371.      C***** CONSTRUCTION WORK IN PROGRESS ACCOUNT
372.      C
373.      WRITE(6,2025) FM1BS(2)
374.      2025 FORMAT(' CWIP ACCOUNT: ',F10.2,'1')
375.      C
376.      C***** NET INCOME
377.      C***** ESHRI=ELECTRIC SHARE OF INCOME
378.      ESHRI=FM1IS(18)/FM1IS(19)
379.      WRITE(6,2026) ESHRI
380.      2026 FORMAT(' ESHRI= ',1X,F10.4)
381.      ENETIN=ESHRI*FM1IS(54)
382.      WRITE(6,2028) ENETIN
383.      2028 FORMAT(' ELECTRIC NET INCOME (NET OF GAS)= ',F10.4)
384.      C***
385.      C*** NOW WRITE RESULTS TO FILES FOR NUFF.
386.      C***
387.      NNN=IREG+19
388.      C*** WRITE OUT BASE YEAR GENERATION
389.      WRITE(NNN,200) GENER
390.      200 FORMAT(E15.9,5X,'GENERATION')
391.      WRITE(6,200) GENER

```

```

392. C***WRITE OUT BASE YEAR SALES
393.     WRITE(NNN,210) SALES
394. 210  FORMAT(E15.9,5X,'SALES TO ULTIMATE')
395.     WRITE(6,210) SALES
396. C***WRITE OUT EXISTING ASSET BASE
397.     WRITE(NNN,220)(ZA1(I),I=1,10)
398. 220  FORMAT(E15.9,5X,'INTANGIBLE (EXISTING)'/E15.9,5X,'STEAM PRODU',
399.      1 'CTION'/E15.9,5X,'NUCLEAR PRODUCTION'/E15.9,5X,'HYDRO CONV'/
400.      2 E15.9,5X,'PUMPED STORAGE'/E15.9,5X,'OTHER PRODUCTION'/E15.9,
401.      3 5X,'TRANSMISSION'/E15.9,5X,'DISTRIBUTION'/E15.9,5X,'GENERAL'/
402.      4 E15.9,5X,'COMMON-ELEC')
403. C***WRITE OUT BASE YEAR STOCK OF BONDS
404.     WRITE(NNN,230) OB
405. 230  FORMAT(E15.9,5X,'OUTSTANDING BONDS')
406. C***WRITE OUT BASE YEAR SHORT TERM DEBT
407.     WRITE(NNN,240) STDEBT
408. 240  FORMAT(E15.9,5X,'SHORT TERM DEBT')
409. C***WRITE OUT BASE YEAR COMMON STOCK
410.     WRITE(NNN,250) COMSTK
411. 250  FORMAT(E15.9,5X,'COMMON STOCK')
412. C***WRITE OUT BASE YEAR REFERED STOCK
413.     WRITE(NNN,260) PRESTK
414. 260  FORMAT(E15.9,5X,'PREFERRED STOCK')
415. C***WRITE OUT BASE YEAR RETAINED EARNINGS
416.     WRITE(NNN,270) RE
417. 270  FORMAT(E15.9,5X,'RETAINED EARNINGS')
418. C***WRITE OUT BASE YEAR EMBEDDED COST OF LONG TERM DEBT
419.     WRITE(NNN,280) RTBSBN
420. 280  FORMAT(E15.9,5X,'EMBEDDED COST OF DEBT')
421. C***WRITE OUT BASE YEAR EMBEDDED COST OF PREFERRED STOCK
422.     WRITE(NNN,285) RTPEQ
423. 285  FORMAT(E15.9,5X,'EMBEDDED COST OF PREFERRED')
424. C***WRITE OUT BASE YEAR TOTAL ELECTRIC REVENUES
425.     WRITE(NNN,290) REVS
426. 290  FORMAT(E15.9,5X,'ELECTRIC REVENUES')
427. C***WRITE OUT BASE YEAR NET INCOME
428.     WRITE(NNN,300) ENETIN
429. 300  FORMAT(E15.9,5X,'NET INCOME')
430. C***WRITE OUT BASE YEAR WORKING CAPITAL
431.     WRITE(NNN,310) WC
432. 310  FORMAT(E15.9,5X,'WORKING CAPITAL')
433. C***WRITE OUT BASE YEAR ACCUMULATED DEFERRED ITC
434.     WRITE(NNN,320) TDITC
435. 320  FORMAT(E15.9,5X,'DEFERRED ITC')
436. C***WRITE OUT BASE YEAR ACCUMULATED DEFERRED INCOME TAXES
437.     WRITE(NNN,330) PRODEF
438. 330  FORMAT(E15.9,5X,'DEFERRED INCOME TAXES')
439. C***WRITE OUT BASE YEAR FUEL AND O&M COSTS ADJUSTED FOR TAXES
440.     WRITE(NNN,340) TXFUEL
441. 340  FORMAT(E15.9,5X,'FUEL ADJUSTED FOR TAXES')
442. C***WRITE OUT BASE YEAR GENERAL TAX RATE
443.     WRITE(NNN,350) GTXRT
444. 350  FORMAT(E15.9,5X,'GENERAL TAX RATE')
445.     RETURN
446.     END
447. //GO.FT08F001 DD DSN=CN8922.PRJ.FM1T778.DATA,DISP=(OLD,KEEP),

```

```
448. // VOL=SER=FE4698,UNIT=TAPE9,LABEL=(,,IN)
449. //GO.FT09F001 DD *
450. 56 NUMBER OF STATES
451. 09 1 STATE NUMBER, DOE REGION NUMBER
452. 23 1
453. 25 1
454. 33 1
455. 44 1
456. 50 1
457. 34 2
458. 36 2
459. 42 3
460. 24 3
461. 10 3
462. 51 3
463. 54 3
464. 11 3
465. 21 4
466. 47 4
467. 37 4
468. 45 4
469. 28 4
470. 01 4
471. 13 4
472. 12 4
473. 55 5
474. 26 5
475. 17 5
476. 18 5
477. 39 5
478. 27 5
479. 35 6
480. 40 6
481. 48 6
482. 05 6
483. 22 6
484. 31 7
485. 20 7
486. 19 7
487. 29 7
488. 30 8
489. 38 8
490. 46 8
491. 56 8
492. 49 8
493. 08 9
494. 06 9
495. 32 9
496. 04 9
497. 53 10
498. 41 10
499. 16 10
500. 02 10
501. 15 10
502. 03 12
503. 07 12
```

504.	14	12
505.	43	12
506.	52	12
507.	214	NUMBER OF ACCOUNTS TO BE ADDED
508.	010021	ACCOUNT NUMBERS WHICH ARE TO BE AGGREGATED
509.	010031	(ENTER CODES FOUND ON TAPE)
510.	010041	
511.	010051	
512.	010061	
513.	010071	
514.	010081	
515.	010091	
516.	010101	
517.	010111	
518.	010121	
519.	010141	
520.	010151	
521.	010161	
522.	010171	
523.	010181	
524.	010191	
525.	010201	
526.	010211	
527.	010221	
528.	010231	
529.	010241	
530.	010251	
531.	010261	
532.	010271	
533.	010281	
534.	010291	
535.	010301	
536.	010311	
537.	010321	
538.	010331	
539.	010341	
540.	010351	
541.	010361	
542.	010371	
543.	010381	
544.	010391	
545.	010401	
546.	010411	
547.	010421	
548.	010431	
549.	010441	
550.	010451	
551.	020011	
552.	020021	
553.	020031	
554.	020041	
555.	020051	
556.	020061	
557.	020071	
558.	020081	
559.	020091	

560.	020101
561.	020111
562.	020121
563.	020131
564.	020141
565.	020151
566.	020161
567.	020171
568.	020181
569.	020191
570.	020201
571.	020211
572.	020221
573.	020231
574.	020241
575.	020251
576.	020261
577.	020271
578.	020281
579.	020291
580.	020301
581.	020311
582.	020321
583.	020331
584.	020341
585.	020351
586.	020361
587.	020371
588.	020381
589.	020391
590.	020401
591.	040021
592.	040041
593.	040051
594.	040061
595.	040071
596.	040081
597.	040091
598.	040101
599.	040111
600.	040121
601.	040131
602.	040141
603.	040151
604.	040161
605.	040171
606.	040181
607.	040191
608.	040211
609.	040219
610.	040269
611.	040279
612.	040289
613.	040299
614.	040309
615.	040319

616.	040339
617.	040349
618.	040359
619.	040369
620.	040389
621.	040399
622.	040409
623.	040419
624.	040429
625.	040439
626.	040449
627.	040459
628.	040469
629.	040489
630.	040499
631.	040509
632.	040519
633.	040529
634.	040539
635.	040549
636.	040569
637.	040579
638.	040589
639.	040609
640.	040619
641.	040629
642.	040639
643.	040649
644.	040659
645.	041299
646.	041369
647.	140021
648.	140041
649.	140051
650.	140061
651.	140071
652.	140081
653.	140091
654.	140101
655.	140121
656.	140141
657.	140151
658.	140161
659.	140171
660.	140181
661.	140191
662.	140201
663.	140211
664.	140221
665.	140231
666.	140241
667.	140251
668.	140261
669.	140271
670.	140281
671.	140321

672.	140331
673.	140341
674.	140351
675.	140361
676.	140371
677.	140381
678.	140391
679.	140401
680.	140421
681.	140431
682.	140441
683.	140451
684.	140461
685.	140471
686.	140481
687.	140491
688.	140501
689.	140511
690.	140521
691.	140531
692.	140541
693.	140551
694.	070056
695.	070156
696.	070236
697.	070326
698.	070416
699.	070426
700.	070536
701.	070696
702.	070836
703.	070886
704.	110011
705.	110021
706.	110031
707.	110041
708.	110051
709.	110061
710.	110071
711.	110081
712.	110091
713.	110101
714.	110111
715.	100051
716.	100241
717.	100621
718.	030039
719.	030031
720.	090102
721.	120091
722.	//GO.FT20F001 DD DSN=&&TEMPU1,DISP=(NEW,PASS),
723.	// UNIT=SYSDA,SPACE=(TRK,(1,1)),
724.	// DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=6160)
725.	//GO.FT21F001 DD DSN=&&TEMPU2,DISP=(NEW,PASS),
726.	// UNIT=SYSDA,SPACE=(TRK,(1,1)),
727.	// DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=6160)


```

728. //GO.FT22F001 DD DSN= &&TEMPU3, DISP=(NEW,PASS),
729. // UNIT=SYSDA, SPACE=(TRK,(1,1)),
730. // DCB=(DSORG=PS, RECFM=FB, LRECL=80, BLKSIZE=6160)
731. //GO.FT23F001 DD DSN= &&TEMPU4, DISP=(NEW,PASS),
732. // UNIT=SYSDA, SPACE=(TRK,(1,1)),
733. // DCB=(DSORG=PS, RECFM=FB, LRECL=80, BLKSIZE=6160)
734. //GO.FT24F001 DD DSN= &&TEMPU5, DISP=(NEW,PASS),
735. // UNIT=SYSDA, SPACE=(TRK,(1,1)),
736. // DCB=(DSORG=PS, RECFM=FB, LRECL=80, BLKSIZE=6160)
737. //GO.FT25F001 DD DSN= &&TEMPU6, DISP=(NEW,PASS),
738. // UNIT=SYSDA, SPACE=(TRK,(1,1)),
739. // DCB=(DSORG=PS, RECFM=FB, LRECL=80, BLKSIZE=6160)
740. //GO.FT26F001 DD DSN= &&TEMPU7, DISP=(NEW,PASS),
741. // UNIT=SYSDA, SPACE=(TRK,(1,1)),
742. // DCB=(DSORG=PS, RECFM=FB, LRECL=80, BLKSIZE=6160)
743. //GO.FT27F001 DD DSN= &&TEMPU8, DISP=(NEW,PASS),
744. // UNIT=SYSDA, SPACE=(TRK,(1,1)),
745. // DCB=(DSORG=PS, RECFM=FB, LRECL=80, BLKSIZE=6160)
746. //GO.FT28F001 DD DSN= &&TEMPU9, DISP=(NEW,PASS),
747. // UNIT=SYSDA, SPACE=(TRK,(1,1)),
748. // DCB=(DSORG=PS, RECFM=FB, LRECL=80, BLKSIZE=6160)
749. //GO.FT29F001 DD DSN= &&TEMPUA, DISP=(NEW,PASS),
750. // UNIT=SYSDA, SPACE=(TRK,(1,1)),
751. // DCB=(DSORG=PS, RECFM=FB, LRECL=80, BLKSIZE=6160)
752. //GO.FT30F001 DD DSN= &&TEMPUT, DISP=(NEW,PASS),
753. // UNIT=SYSDA, SPACE=(TRK,(1,1)),
754. // DCB=(DSORG=PS, RECFM=FB, LRECL=80, BLKSIZE=6160)
755. //DELD EXEC PGM=IEFBR14, REGION=10K
756. //DD1 DD DSN=CN6376.MI2.NUFS.FORM1.PDS78, DISP=(MOD,DELETE),
757. // UNIT=DASD, SPACE=(TRK,(3,1,1))
758. //DELD EXEC PGM=IEFBR14, REGION=10K
759. //DD1 DD DSN=CN6376.MI2.NUFS.FORM1.PDS78, DISP=(NEW,CATLG),
760. // UNIT=DASD,
761. // SPACE=(TRK,(3,1,1)), DCB=(DSORG=PO, RECFM=FB, BLKSIZE=6160, LRECL=80)
762. //GEN01 EXEC PGM=IEBGENER
763. //SYSPRINT DD SYSOUT=A
764. //SYSUT1 DD DISP=(OLD,DELETE), DSN= &&TEMPU1
765. //SYSUT2 DD DISP=(OLD,KEEP), DSN=CN6376.MI2.NUFS.FORM1.PDS78(U1)
766. //SYSIN DD DUMMY
767. //GEN02 EXEC PGM=IEBGENER
768. //SYSPRINT DD SYSOUT=A
769. //SYSUT1 DD DISP=(OLD,DELETE), DSN= &&TEMPU2
770. //SYSUT2 DD DISP=(OLD,KEEP), DSN=CN6376.MI2.NUFS.FORM1.PDS78(U2)
771. //SYSIN DD DUMMY
772. //GEN03 EXEC PGM=IEBGENER
773. //SYSPRINT DD SYSOUT=A
774. //SYSUT1 DD DISP=(OLD,DELETE), DSN= &&TEMPU3
775. //SYSUT2 DD DISP=(OLD,KEEP), DSN=CN6376.MI2.NUFS.FORM1.PDS78(U3)
776. //SYSIN DD DUMMY
777. //GEN04 EXEC PGM=IEBGENER
778. //SYSPRINT DD SYSOUT=A
779. //SYSUT1 DD DISP=(OLD,DELETE), DSN= &&TEMPU4
780. //SYSUT2 DD DISP=(OLD,KEEP), DSN=CN6376.MI2.NUFS.FORM1.PDS78(U4)
781. //SYSIN DD DUMMY
782. //GEN05 EXEC PGM=IEBGENER
783. //SYSPRINT DD SYSOUT=A

```

784. //SYSUT1 DD DISP=(OLD,DELETE),DSN=*&TEMPU5
785. //SYSUT2 DD DISP=(OLD,KEEP),DSN=CN6376.MI2.NUFS.FORM1.PDS78(U5)
786. //SYSIN DD DUMMY
787. //GEN06 EXEC PGM=IEBGENER
788. //SYSPRINT DD SYSOUT=A
789. //SYSUT1 DD DISP=(OLD,DELETE),DSN=*&TEMPU6
790. //SYSUT2 DD DISP=(OLD,KEEP),DSN=CN6376.MI2.NUFS.FORM1.PDS78(U6)
791. //SYSIN DD DUMMY
792. //GEN07 EXEC PGM=IEBGENER
793. //SYSPRINT DD SYSOUT=A
794. //SYSUT1 DD DISP=(OLD,DELETE),DSN=*&TEMPU7
795. //SYSUT2 DD DISP=(OLD,KEEP),DSN=CN6376.MI2.NUFS.FORM1.PDS78(U7)
796. //SYSIN DD DUMMY
797. //GEN08 EXEC PGM=IEBGENER
798. //SYSPRINT DD SYSOUT=A
799. //SYSUT1 DD DISP=(OLD,DELETE),DSN=*&TEMPU8
800. //SYSUT2 DD DISP=(OLD,KEEP),DSN=CN6376.MI2.NUFS.FORM1.PDS78(U8)
801. //SYSIN DD DUMMY
802. //GEN09 EXEC PGM=IEBGENER
803. //SYSPRINT DD SYSOUT=A
804. //SYSUT1 DD DISP=(OLD,DELETE),DSN=*&TEMPU9
805. //SYSUT2 DD DISP=(OLD,KEEP),DSN=CN6376.MI2.NUFS.FORM1.PDS78(U9)
806. //SYSIN DD DUMMY
807. //GEN10 EXEC PGM=IEBGENER
808. //SYSPRINT DD SYSOUT=A
809. //SYSUT1 DD DISP=(OLD,DELETE),DSN=*&TEMPUA
810. //SYSUT2 DD DISP=(OLD,KEEP),DSN=CN6376.MI2.NUFS.FORM1.PDS78(UA)
811. //SYSIN DD DUMMY
812. //GEN11 EXEC PGM=IEBGENER
813. //SYSPRINT DD SYSOUT=A
814. //SYSUT1 DD DISP=(OLD,DELETE),DSN=*&TEMPUT
815. //SYSUT2 DD DISP=(OLD,KEEP),DSN=CN6376.MI2.NUFS.FORM1.PDS78(UT)
816. //SYSIN DD DUMMY

?

ii) LISTING OF NUS.FORM1M.FORT

```

1. //MI2UFORM JOB (6376,D02,,999),'FORM1M',TIME=(,11),MSGCLASS=A
2. //FPC EXEC FORTGCG,REGION=400K
3. //FORT.SYSIN DD *
4. C***NUFS.FORM1M.FORT. THIS PROGRAM READS DATA FROM THE FORM 1M TAPE AND
5. C***WRITES OUT DATA NEEDED FOR THE NATIONAL UTILITY FINANCIAL STATEMENT
6. C***MODEL (NUFS).
7. C***
8. C***WRITTEN BY MARK R. INGLIS, ICF INC. JULY 31,1980
9. C***
10. DIMENSION ISTATE(60),IFED(2,100),INFO(100)
11. DOUBLE PRECISION AMNT,AMNT1,AMOUNT(11,100),AMNT2(100)
12. REAL*4 ACODE,JTYP,KTYP,LTYP,MTYP,NTYP,OTYP,PTYP,QTYP,
13. * RTYP,RBKTYT,NOTYP,N1TYP,N2TYP,N3TYP,N4TYP,N5TYP,N6TYP,
14. * N7TYP,N8TYP,N9TYP
15. DATA JTYP/'J'/
16. DATA KTYP/'K'/
17. DATA LTYP/'L'/
18. DATA MTYP/'M'/
19. DATA NTYP/'N'/
20. DATA OTYP/'O'/
21. DATA PTYP/'P'/
22. DATA QTYP/'Q'/
23. DATA RTYP/'R'/
24. C DATA RBKTYT/'/'/
25. DATA NOTYP/'O'/
26. DATA N1TYP/'1'/
27. DATA N2TYP/'2'/
28. DATA N3TYP/'3'/
29. DATA N4TYP/'4'/
30. DATA N5TYP/'5'/
31. DATA N6TYP/'6'/
32. DATA N7TYP/'7'/
33. DATA N8TYP/'8'/
34. DATA N9TYP/'9'/
35. C***INITIALIZE AMOUNT, THE MATRIX WHICH CONTAINS TOTALS FOR EACH REGION
36. C***IE., AMOUNT(J,I)=AMOUNT OF ACCOUNT I SUMMED OVER ALL UTILITIES IN
37. C***REGION J. (REGION 11 = NATION)
38. DATA AMOUNT/1100*0.0/
39. C***NOW, A USER SUPPLIED INPUT FILE IS READ. THIS FILE CONTAINS STATE,
40. C***FEDERAL PROJECTS, AND ACCOUNT CODES.
41. C***** NSTATE=# OF STATES
42. C***** NINFO= # OF ACCOUNTS TO BE ADDED OVER REGIONS
43. C***** ISTATE(I)=DOE REGION THAT STATE I IS IN
44. C***** INFO(I)= 6 DIGIT ACCOUNT CODE
45. C***** NFED= # OF FEDERAL PROJECT CODES
46. C***** IFED(1,I)=FEDERAL PROJECT CODE
47. C***** IFED(2,I)=DOE REGION THAT FED PROJECT I IS IN
48. READ(9,*)NSTATE
49. DO 77 J=1,NSTATE
50. 77 READ(9,*) I,ISTATE(I)
51. READ(9,*)NFED
52. DO 777 I=1,NFED
53. 777 READ(9,*) IFED(1,I),IFED(2,I)
54. READ(9,*)NINFO
55. DO 88 I=1,NINFO

```

```

56.      88  READ(9,802)INFO(I)
57.      802  FORMAT(I6)
58.  C***NOW THE FORM 1M TAPE IS READ.  EACH RECORD CONTAINS THE TOTAL FOR
59.  C***FOR THE GIVEN ACCOUNT FOR THE GIVEN UTILITY.  IF THE RECORD CONTAINS
60.  C***AN ACCOUNT WHICH WAS READ INTO INFO(I), THEN THE TOTAL IS ADDED IN
61.  C***TO THE PROPER REGION.  IF NOT, THE NEXT RECORD IS READ.
62.  C*****JSTATE = STATE CODE
63.  C*****JUTIL=UTILITY CODE
64.  C*****JINFO = ACCOUNT CODE
65.  3020  READ(8,1001,END=3030) JSTATE,JUTIL,JINFO,JAMNT1,JAMNT,ACODE
66.  1001  FORMAT(I2,I4,I6,1X,I6,I5,A1)
67.      DO 60 I=1,NINFO
68.      IF(JINFO.EQ.INFO(I)) GO TO 50
69.      60  CONTINUE
70.      GO TO 3020
71.  50  IF (JSTATE .EQ. 60) GO TO 120
72.      IREG=ISTATE(JSTATE)
73.  C***IF A STATES REGION IS 12, THIS MEANS THE STATE IS NOT IN A DOE REGION.
74.  C***IN THIS CASE, THE NEXT RECORD IS READ. (EXAMPLES: PUERTO RICO, ETC.)
75.      IF (IREG .EQ. 12) GO TO 3020
76.      GO TO 70
77.  120  DO 40 J=1,NFED
78.      IF (JUTIL .EQ. IFED(1,J)) GO TO 65
79.      40  CONTINUE
80.      GO TO 3020
81.      65  IREG=IFED(2,J)
82.  C***THE TOTAL FOR THE ACCOUNT IS WRITTEN IN 'CODE', SOMETIMES CONTAINING
83.  C***LETTERS.  THIS NEXT SECTION CONVERTS THE ACCOUNT INTO A REAL NUMBER.
84.  C
85.      70  CONTINUE
86.      IF(ACODE.EQ.JTYP) GO TO 100
87.      IF(ACODE.EQ.KTYP) GO TO 101
88.      IF(ACODE.EQ.LTYP) GO TO 102
89.      IF(ACODE.EQ.MTYP) GO TO 103
90.      IF(ACODE.EQ.NTYP) GO TO 104
91.      IF(ACODE.EQ.OTYP) GO TO 105
92.      IF(ACODE.EQ.PTYP) GO TO 106
93.      IF(ACODE.EQ.QTYP) GO TO 107
94.      IF(ACODE.EQ.RTYP) GO TO 108
95.  C      IF(ACODE.EQ.RBKTYT) GO TO 109
96.      IF(ACODE.EQ.NOTYP) GO TO 110
97.      IF(ACODE.EQ.N1TYP) GO TO 111
98.      IF(ACODE.EQ.N2TYP) GO TO 112
99.      IF(ACODE.EQ.N3TYP) GO TO 113
100.     IF(ACODE.EQ.N4TYP) GO TO 114
101.     IF(ACODE.EQ.N5TYP) GO TO 115
102.     IF(ACODE.EQ.N6TYP) GO TO 116
103.     IF(ACODE.EQ.N7TYP) GO TO 117
104.     IF(ACODE.EQ.N8TYP) GO TO 118
105.     IF(ACODE.EQ.N9TYP) GO TO 119
106.     GO TO 109
107.  100  JAMNT=-10*JAMNT-1
108.     GO TO 900
109.  101  JAMNT=-10*JAMNT-2
110.     GO TO 900
111.  102  JAMNT=-10*JAMNT-3

```

```

112.      GO TO 900
113.    103  JAMNT=-10*JAMNT-4
114.      GO TO 900
115.    104  JAMNT=-10*JAMNT-5
116.      GO TO 900
117.    105  JAMNT=-10*JAMNT-6
118.      GO TO 900
119.    106  JAMNT=-10*JAMNT-7
120.      GO TO 900
121.    107  JAMNT=-10*JAMNT-8
122.      GO TO 900
123.    108  JAMNT=-10*JAMNT-9
124.      GO TO 900
125.    109  JAMNT=-10*JAMNT
126.      GO TO 900
127.    110  JAMNT=10*JAMNT
128.      GO TO 900
129.    111  JAMNT=10*JAMNT+1
130.      GO TO 900
131.    112  JAMNT=10*JAMNT+2
132.      GO TO 900
133.    113  JAMNT=10*JAMNT+3
134.      GO TO 900
135.    114  JAMNT=10*JAMNT+4
136.      GO TO 900
137.    115  JAMNT=10*JAMNT+5
138.      GO TO 900
139.    116  JAMNT=10*JAMNT+6
140.      GO TO 900
141.    117  JAMNT=10*JAMNT+7
142.      GO TO 900
143.    118  JAMNT=10*JAMNT+8
144.      GO TO 900
145.    119  JAMNT=10*JAMNT+9
146.      GO TO 900
147.    900  AMNT=FLOAT(JAMNT)
148.      IF(JAMNT1.EQ.0) GO TO 6010
149.      AMNT1=FLOAT(JAMNT1)
150.      AMNT=AMNT+AMNT1*10.0**6
151.    6010  CONTINUE
152.    C***THE CONVERTED AMOUNT IS NOW ADDED INTO THE PROPER REGION, AND THE
153.    C***NEXT RECORD IS READ
154.      AMOUNT(IREG,I)=AMNT+AMOUNT(IREG,I)
155.      AMOUNT(11,I)=AMNT+AMOUNT(11,I)
156.      GO TO 3020
157.    3030  CONTINUE
158.    C
159.    C***AFTER THE WHOLE TAPE HAS BEEN READ, SUBROUTINE FORM1M IS CALLED FOR
160.    C***EACH REGION. THIS SUBROUTINE WRITES OUT THE DATA NECESSARY FOR THE
161.    C***NUFS MODEL.
162.    C
163.      DO 950 IREG=1,11
164.      DO 940 I=1,NINFO
165.    940    AMNT2(I)=AMOUNT(IREG,I)
166.      CALL FORM1M(IREG,AMNT2)
167.    950    CONTINUE

```

```

168.      STOP
169.      END
170.      SUBROUTINE FORM1M(IREG,AMOUNT)
171.      C
172.      C
173.      C*****
174.      C*                                     *
175.      C*      SUBROUTINE FORM1M                                     *
176.      C*      THIS SUBROUTINE TAKES THE TOTALS FROM THE MAIN      *
177.      C*      PROGRAM FOR EACH REGION AND WRITES OUT FILES CON-*
178.      C*      TAINING THE DATA NEEDED BY NUFS. IT ALSO PRINTS *
179.      C*      OUT VARIOUS QUANTITIES OF INTEREST TO UNIT 6.      *
180.      C*      (UNIT 6=LINE PRINTER)                                *
181.      C*****
182.      C
183.      C
184.      DOUBLE PRECISION FM3BS(49),FM3IS(21)
185.      DOUBLE PRECISION FM3EPS(17),FM3DEP(3)
186.      DOUBLE PRECISION TEMP5,TEMP6,EPPOR,AMOUNT(100)
187.      DIMENSION ZA1(10)
188.      C***FIRST, PUT DATA INTO MATRICES BY SCHEDULE:
189.      C***      FM3BS=BALANCE SHEET
190.      C***      FM3IS=INCOME SHEET
191.      C***      FM3EPS=ELECTRIC PLANT IN SERVICE
192.      C***      ETC.
193.      ICOUNT=0
194.      DO 12 I=1,49
195.      ICOUNT=ICOUNT+1
196.      12  FM3BS(I)=AMOUNT(ICOUNT)/1000000.
197.      DO 20 I=1,21
198.      ICOUNT=ICOUNT+1
199.      20  FM3IS(I)=AMOUNT(ICOUNT)/1000000.
200.      DO 30 I=1,17
201.      ICOUNT=ICOUNT+1
202.      30  FM3EPS(I)=AMOUNT(ICOUNT)/1000000.
203.      DO 40 I=1,3
204.      ICOUNT=ICOUNT+1
205.      40  FM3DEP(I)=AMOUNT(ICOUNT/1000000.)
206.      ICOUNT=ICOUNT+1
207.      SALES=AMOUNT(ICOUNT)/1.0E09
208.      GENER=AMOUNT(ICOUNT+1)/1.0E09
209.      C
210.      WRITE(6,11)IREG
211.      11  FORMAT(' THE TOTALS FOR REGION ',I2,':')
212.      C
213.      C*****NOW STRIP OFF RELEVANT AMOUNTS FOR NUFS AND INTEREST. (IE. THIS
214.      C*****ROUTINE CAN BE MODIFIED TO PRINT OUT ANY ACCOUNT.)
215.      C
216.      WRITE(6,311)
217.      311  FORMAT(' THE VALUES FOR FM3EPS ARE:')
218.      DO 9011 J=1,17
219.      9011 WRITE(6,2015)FM3EPS(J)
220.      2015 FORMAT(5X,F10.2)
221.      ZA1(1)=FM3EPS(1)
222.      ZA1(2)=FM3EPS(2)
223.      ZA1(3)=FM3EPS(3)

```

```

224.      ZA1(4)=FM3EPS(4)
225.      ZA1(5)=0.0
226.      ZA1(6)=FM3EPS(5)
227.      ZA1(7)=FM3EPS(7)
228.      ZA1(8)=FM3EPS(8)
229.      ZA1(9)=FM3EPS(9)
230.      ZA1(10)=0.0
231.      WRITE(6,2014)
232. 2014  FORMAT(' THE VALUES FOR ZA(I) ARE')
233.      DO 9015 J=1,10
234. 9015   WRITE(6,2015) ZA1(J)
235.      C
236.  C***** FIND RELATIVE ERROR FOR ASSET TOTALS.
237.      C          RE1-RE2 = RELATIVE ERRORS
238.      C
239.      TEMP5=0.0
240.      DO 9007 J=1,5
241. 9007   TEMP5=TEMP5+FM3EPS(J)
242.      C
243.      RE1=(FM3EPS(6)-TEMP5)/FM3EPS(6)
244.      WRITE(6,2000)RE1
245. 2000  FORMAT(' R.E. FOR FM3EPS(6) IS: ',F10.8)
246.      C
247.      TEMP6=0.0
248.      DO 9008 J=7,9
249. 9008   TEMP6=TEMP6+FM3EPS(J)
250.      TEMP6=TEMP6+TEMP5
251.      C
252.      RE2=(FM3EPS(10)-TEMP6)/FM3EPS(10)
253.      WRITE(6,2001)RE2
254. 2001  FORMAT(' R.E. FOR FM3EPS(10) IS: ',F10.8)
255.      C
256.  C***** COST OF EMBEDDED DEBT
257.      C
258.      RTBSBN=FM3IS(15)/FM3BS(31)
259.      WRITE(6,2002)RTBSBN
260. 2002  FORMAT(' RTBSBN= ',F10.4)
261.      C
262.  C***** GENERAL TAX RATE
263.      C
264.      GTXRT=FM3IS(5)/FM3IS(1)
265.      WRITE(6,2003)GTXRT
266. 2003  FORMAT(' GTXRT= ',F10.4)
267.      C
268.  C***** ELECTRIC PORTION OF TOTAL PLANT
269.  C***(NOTE:  MANY OF THESE ACCOUNTS INCLUDE GAS UTILITIES.  THIS IS
270.  C***NETTED OUT BY MULTIPLYING BY THIS RATIO.)
271.      EPPOR=FM3EPS(10)/(FM3EPS(17)-FM3EPS(12)+FM3BS(7))
272.      WRITE(6,2017)EPPOR
273. 2017  FORMAT(' EPPOR= ',D10.2)
274.      C
275.  C***** RETAINED EARNINGS
276.      RE=FM3BS(24)
277.      WRITE(6,2004)RE
278. 2004  FORMAT(' RETAINED EARNINGS= ',F10.2)
279.      RE=RE+EPPOR

```



```

280. C
281. C***** STOCK OF OUTSTANDING BONDS PRE-BASE YEAR
282. C
283.      OB=FM3BS(31)
284.      WRITE(6,2005)OB
285. 2005 FORMAT(' OUTSTANDING BONDS= ',F10.2)
286.      OB=OB*EPPOR
287. C
288. C***** REVENUES PRE-BASE YEAR
289. C
290.      REVS=FM3IS(1)
291.      WRITE(6,2006) REVS
292. 2006 FORMAT(' ELECTRIC REVENUES= ',F10.2)
293. C
294. C***** CAPITAL STRUCTURE
295. C
296.      TEMP7=FM3BS(25)+FM3BS(31)+FM3BS(33)
297.      W1=FM3BS(31)/TEMP7
298.      WRITE(6,2007) W1
299. 2007 FORMAT(' W1= ',F10.4)
300.      W2=FM3BS(33)/TEMP7
301.      WRITE(6,2008) W2
302. 2008 FORMAT(' W2= ',F10.4)
303.      W3=FM3BS(25)/TEMP7
304.      WRITE(6,2009) W3
305. 2009 FORMAT(' W3= ',F10.4)
306.      W4=0.0
307.      WRITE(6,2010) W4
308. 2010 FORMAT(' W4= ',F10.4)
309. C
310. C***** FUEL & O&M COSTS ADJUSTED FOR TAXES
311.      TXFUEL=(FM3IS(2)+FM3IS(3))
312.      WRITE(6,2013)TXFUEL
313. 2013 FORMAT(' FUEL PLUS O & M= ',F10.2)
314.      TXFUEL=TXFUEL/(1-GTXRT)
315. C
316. C***** WORKING CAPITAL
317. C
318.      WC=(FM3BS(15)-FM3BS(39))
319.      WRITE(6,2021) WC
320. 2021 FORMAT(' WORKING CAPITAL= ',F10.4)
321.      WC=WC*EPPOR
322. C
323. C***** COMMON STOCK PRE-BASE YEAR
324. C
325.      COMSTK=(FM3BS(25)-FM3BS(24))
326.      WRITE(6,2022) COMSTK
327. 2022 FORMAT(' COMMON STOCK= ',F10.4)
328.      COMSTK=COMSTK*EPPOR
329. C
330. C***** CONSTRUCTION WORK IN PROGRESS ACCOUNT
331. C
332.      WRITE(6,2025) FM3EPS(12)
333. 2025 FORMAT(' CWIP ACCOUNT:',F10.2,'1')
334. C
335. C*****NET INCOME

```

```

336. C
337. ENETIN=FM3IS(21)
338. WRITE(6,2028) ENETIN
339. 2028 FORMAT(' NET INCOME=',F10.4)
340. C***
341. C***NOW WRITE RESULTS TO FILES FOR NUF5.
342. C***
343. NNN=IREG+19
344. C***WRITE OUT BASE YEAR GENERATION
345. WRITE(NNN,200) GENER
346. 200 FORMAT(E15.9,5X,'GENERATION')
347. WRITE(6,200) GENER
348. C***WRITE OUT BASE YEAR SALES
349. WRITE(NNN,210) SALES
350. 210 FORMAT(E15.9,5X,'SALES TO ULTIMATE')
351. WRITE(6,210) SALES
352. C***WRITE OUT EXISTING ASSET BASE
353. WRITE(NNN,220)(ZAI(I),I=1,10)
354. 220 FORMAT(E15.9,5X,'INTANGIBLE (EXISTING)'/E15.9,5X,'STEAM PRODU'
355. 1 'CTION'/E15.9,5X,'NUCLEAR PRODUCTION'/E15.9,5X,'HYDRO CONV'/
356. 2 E15.9,5X,'PUMPED STORAGE'/E15.9,5X,'OTHER PRODUCTION'/E15.9,
357. 3 5X,'TRANSMISSION'/E15.9,5X,'DISTRIBUTION'/E15.9,5X,'GENERAL'/
358. 4 E15.9,5X,'COMMON-ELEC')
359. C***WRITE OUT BASE YEAR STOCK OF BONDS
360. WRITE(NNN,230) OB
361. 230 FORMAT(E15.9,5X,'OUTSTANDING BONDS')
362. C***WRITE OUT BASE YEAR COMMON STOCK
363. WRITE(NNN,250) COMSTK
364. 250 FORMAT(E15.9,5X,'COMMON STOCK')
365. C***WRITE OUT BASE YEAR RETAINED EARNINGS
366. WRITE(NNN,270) RE
367. 270 FORMAT(E15.9,5X,'RETAINED EARNINGS')
368. C***WRITE OUT BASE YEAR EMBEDDED COST OF LONG TERM DEBT
369. WRITE(NNN,280) RTBSBN
370. 280 FORMAT(E15.9,5X,'COST OF EMBEDDED DEBT')
371. C***WRITE OUT BASE YEAR TOTAL ELECTRIC REVENUES
372. WRITE(NNN,290) REVS
373. 290 FORMAT(E15.9,5X,'ELECTRIC REVENUES')
374. C***WRITE OUT BASE YEAR NET INCOME
375. WRITE(NNN,300) ENETIN
376. 300 FORMAT(E15.9,5X,'NET INCOME')
377. C***WRITE OUT BASE YEAR WORKING CAPITAL
378. WRITE(NNN,310) WC
379. 310 FORMAT(E15.9,5X,'WORKING CAPITAL')
380. C***WRITE OUT BASE YEAR FUEL AND O&M COSTS ADJUSTED FOR TAXES
381. WRITE(NNN,340) TXFUEL
382. 340 FORMAT(E15.9,5X,'FUEL ADJUSTED FOR TAXES')
383. C***WRITE OUT BASE YEAR GENERAL TAX RATE
384. WRITE(NNN,350) GTXRT
385. 350 FORMAT(E15.9,5X,'GENERAL TAX RATE')
386. RETURN
387. END
388. //GO.FT08F001 DD DSN=CN9922.PRJ.FM3T778.DATA,DISP=(OLD,KEEP),
389. // VOL=SER=F14225,UNIT=TAPE9,LABEL=(...,IN)
390. //GO.FT09F001 DD *
391. 56 NUMBER OF STATES (INCLUDING DC)

```

	STATE NUMBER,	DOE REGION NUMBER
392.	09	1
393.	23	1
394.	25	1
395.	33	1
396.	44	1
397.	50	1
398.	34	2
399.	36	2
400.	42	3
401.	24	3
402.	10	3
403.	51	3
404.	54	3
405.	11	3
406.	21	4
407.	47	4
408.	37	4
409.	45	4
410.	28	4
411.	01	4
412.	13	4
413.	12	4
414.	55	5
415.	26	5
416.	17	5
417.	18	5
418.	39	5
419.	27	5
420.	35	6
421.	40	6
422.	48	6
423.	05	6
424.	22	6
425.	31	7
426.	20	7
427.	19	7
428.	29	7
429.	30	8
430.	38	8
431.	46	8
432.	56	8
433.	49	8
434.	08	9
435.	06	9
436.	32	9
437.	04	9
438.	53	10
439.	41	10
440.	16	10
441.	02	10
442.	15	10
443.	03	12
444.	07	12
445.	14	12
446.	43	12
447.	52	12

448.	91	NUMBER OF FEDERAL PROJECTS
449.	0360 3	FEDERAL PROJECT NUMBER, DOE REGION NUMBER
450.	0380 3	
451.	0085 4	
452.	0325 4	
453.	0395 4	
454.	0270 4	
455.	0280 4	
456.	0285 4	
457.	0290 4	
458.	0300 4	
459.	0310 4	
460.	0320 4	
461.	0330 4	
462.	0340 4	
463.	0350 4	
464.	0365 4	
465.	0370 4	
466.	0385 4	
467.	0390 4	
468.	0400 4	
469.	0580 4	
470.	0576 5	
471.	0436 6	
472.	0461 6	
473.	0462 6	
474.	0480 6	
475.	0483 6	
476.	0505 6	
477.	0410 6	
478.	0415 6	
479.	0420 6	
480.	0425 6	
481.	0430 6	
482.	0435 6	
483.	0440 6	
484.	0445 6	
485.	0450 6	
486.	0460 6	
487.	0465 6	
488.	0470 6	
489.	0485 6	
490.	0500 6	
491.	0510 6	
492.	0488 7	
493.	0490 7	
494.	0090 8	
495.	0102 8	
496.	0170 8	
497.	0175 8	
498.	0190 8	
499.	0200 8	
500.	0210 8	
501.	0240 8	
502.	0550 8	
503.	0565 8	

504. 0180 8
 505. 0530 9
 506. 0540 9
 507. 0072 10
 508. 0100 10
 509. 0112 10
 510. 0114 10
 511. 0010 10
 512. 0020 10
 513. 0025 10
 514. 0030 10
 515. 0040 10
 516. 0050 10
 517. 0060 10
 518. 0065 10
 519. 0070 10
 520. 0075 10
 521. 0080 10
 522. 0105 10
 523. 0110 10
 524. 0115 10
 525. 0120 10
 526. 0130 10
 527. 0560 10
 528. 0072 10
 529. 0085 4
 530. 0102 8
 531. 0112 10
 532. 0325 4
 533. 0395 4
 534. 0436 6
 535. 0461 6
 536. 0462 4
 537. 0483 6
 538. 0488 7
 539. 0505 6
 540. 92
 541. 010021
 542. 010041
 543. 010051
 544. 010091
 545. 010101
 546. 010111
 547. 010121
 548. 010141
 549. 010151
 550. 010181
 551. 010191
 552. 010201
 553. 010211
 554. 010221
 555. 010231
 556. 010251
 557. 010261
 558. 010271
 559. 010291

NUMBER OF ACCOUNTS

ACCOUNT NUMBERS WHICH ARE TO AGGREGATED

560.	010301
561.	010371
562.	020021
563.	020031
564.	020041
565.	020051
566.	020071
567.	020081
568.	020091
569.	020101
570.	020111
571.	020121
572.	020141
573.	020151
574.	020161
575.	020171
576.	020181
577.	020191
578.	020201
579.	020211
580.	020231
581.	020241
582.	020251
583.	020261
584.	020281
585.	020291
586.	020301
587.	020311
588.	020321
589.	020371
590.	030011
591.	030021
592.	030031
593.	030041
594.	030051
595.	030061
596.	030071
597.	030081
598.	030091
599.	030101
600.	030121
601.	030131
602.	030141
603.	030151
604.	030161
605.	030171
606.	030181
607.	030191
608.	030201
609.	030211
610.	030221
611.	060011
612.	060021
613.	060031
614.	060041
615.	060051

```

616. 060061
617. 060071
618. 060081
619. 060091
620. 060101
621. 060111
622. 060121
623. 060131
624. 060141
625. 060151
626. 060161
627. 060181
628. 070211
629. 070221
630. 070251
631. 040462
632. 090051
633. //GO.FT20F001 DD DSN=&&TEMPU1,DISP=(NEW,PASS),
634. // UNIT=SYSDA,SPACE=(TRK,(1,1)),
635. // DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=6160)
636. //GO.FT21F001 DD DSN=&&TEMPU2,DISP=(NEW,PASS),
637. // UNIT=SYSDA,SPACE=(TRK,(1,1)),
638. // DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=6160)
639. //GO.FT22F001 DD DSN=&&TEMPU3,DISP=(NEW,PASS),
640. // UNIT=SYSDA,SPACE=(TRK,(1,1)),
641. // DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=6160)
642. //GO.FT23F001 DD DSN=&&TEMPU4,DISP=(NEW,PASS),
643. // UNIT=SYSDA,SPACE=(TRK,(1,1)),
644. // DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=6160)
645. //GO.FT24F001 DD DSN=&&TEMPU5,DISP=(NEW,PASS),
646. // UNIT=SYSDA,SPACE=(TRK,(1,1)),
647. // DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=6160)
648. //GO.FT25F001 DD DSN=&&TEMPU6,DISP=(NEW,PASS),
649. // UNIT=SYSDA,SPACE=(TRK,(1,1)),
650. // DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=6160)
651. //GO.FT26F001 DD DSN=&&TEMPU7,DISP=(NEW,PASS),
652. // UNIT=SYSDA,SPACE=(TRK,(1,1)),
653. // DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=6160)
654. //GO.FT27F001 DD DSN=&&TEMPU8,DISP=(NEW,PASS),
655. // UNIT=SYSDA,SPACE=(TRK,(1,1)),
656. // DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=6160)
657. //GO.FT28F001 DD DSN=&&TEMPU9,DISP=(NEW,PASS),
658. // UNIT=SYSDA,SPACE=(TRK,(1,1)),
659. // DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=6160)
660. //GO.FT29F001 DD DSN=&&TEMPUA,DISP=(NEW,PASS),
661. // UNIT=SYSDA,SPACE=(TRK,(1,1)),
662. // DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=6160)
663. //GO.FT30F001 DD DSN=&&TEMPUT,DISP=(NEW,PASS),
664. // UNIT=SYSDA,SPACE=(TRK,(1,1)),
665. // DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=6160)
666. //DELD EXEC PGM=IEFBR14,REGION=10K
667. //DD1 DD DSN=CN6376.MI2.NUFS.FORM1M.PDS78,DISP=(MOD,DELETE),
668. // UNIT=DASD,SPACE=(TRK,(3,1,1))
669. //DELD EXEC PGM=IEFBR14,REGION=10K
670. //DD1 DD DSN=CN6376.MI2.NUFS.FORM1M.PDS78,DISP=(NEW,CATLG),
671. // UNIT=DASD,

```

```
672. // SPACE=(TRK,(3,1,1)),DCB=(DSORG=PO,RECFM=FB,BLKSIZE=6160,LRECL=80)
673. //GEN01 EXEC PGM=IEBGENER
674. //SYSPRINT DD SYSOUT=A
675. //SYSUT1 DD DISP=(OLD,DELETE),DSN=*&TEMPU1
676. //SYSUT2 DD DISP=(OLD,KEEP),DSN=CN6376.MI2.NUFS.FORM1M.PDS78(U1)
677. //SYSIN DD DUMMY
678. //GEN02 EXEC PGM=IEBGENER
679. //SYSPRINT DD SYSOUT=A
680. //SYSUT1 DD DISP=(OLD,DELETE),DSN=*&TEMPU2
681. //SYSUT2 DD DISP=(OLD,KEEP),DSN=CN6376.MI2.NUFS.FORM1M.PDS78(U2)
682. //SYSIN DD DUMMY
683. //GEN03 EXEC PGM=IEBGENER
684. //SYSPRINT DD SYSOUT=A
685. //SYSUT1 DD DISP=(OLD,DELETE),DSN=*&TEMPU3
686. //SYSUT2 DD DISP=(OLD,KEEP),DSN=CN6376.MI2.NUFS.FORM1M.PDS78(U3)
687. //SYSIN DD DUMMY
688. //GEN04 EXEC PGM=IEBGENER
689. //SYSPRINT DD SYSOUT=A
690. //SYSUT1 DD DISP=(OLD,DELETE),DSN=*&TEMPU4
691. //SYSUT2 DD DISP=(OLD,KEEP),DSN=CN6376.MI2.NUFS.FORM1M.PDS78(U4)
692. //SYSIN DD DUMMY
693. //GEN05 EXEC PGM=IEBGENER
694. //SYSPRINT DD SYSOUT=A
695. //SYSUT1 DD DISP=(OLD,DELETE),DSN=*&TEMPU5
696. //SYSUT2 DD DISP=(OLD,KEEP),DSN=CN6376.MI2.NUFS.FORM1M.PDS78(U5)
697. //SYSIN DD DUMMY
698. //GEN06 EXEC PGM=IEBGENER
699. //SYSPRINT DD SYSOUT=A
700. //SYSUT1 DD DISP=(OLD,DELETE),DSN=*&TEMPU6
701. //SYSUT2 DD DISP=(OLD,KEEP),DSN=CN6376.MI2.NUFS.FORM1M.PDS78(U6)
702. //SYSIN DD DUMMY
703. //GEN07 EXEC PGM=IEBGENER
704. //SYSPRINT DD SYSOUT=A
705. //SYSUT1 DD DISP=(OLD,DELETE),DSN=*&TEMPU7
706. //SYSUT2 DD DISP=(OLD,KEEP),DSN=CN6376.MI2.NUFS.FORM1M.PDS78(U7)
707. //SYSIN DD DUMMY
708. //GEN08 EXEC PGM=IEBGENER
709. //SYSPRINT DD SYSOUT=A
710. //SYSUT1 DD DISP=(OLD,DELETE),DSN=*&TEMPU8
711. //SYSUT2 DD DISP=(OLD,KEEP),DSN=CN6376.MI2.NUFS.FORM1M.PDS78(U8)
712. //SYSIN DD DUMMY
713. //GEN09 EXEC PGM=IEBGENER
714. //SYSPRINT DD SYSOUT=A
715. //SYSUT1 DD DISP=(OLD,DELETE),DSN=*&TEMPU9
716. //SYSUT2 DD DISP=(OLD,KEEP),DSN=CN6376.MI2.NUFS.FORM1M.PDS78(U9)
717. //SYSIN DD DUMMY
718. //GEN10 EXEC PGM=IEBGENER
719. //SYSPRINT DD SYSOUT=A
720. //SYSUT1 DD DISP=(OLD,DELETE),DSN=*&TEMPUA
721. //SYSUT2 DD DISP=(OLD,KEEP),DSN=CN6376.MI2.NUFS.FORM1M.PDS78(UA)
722. //SYSIN DD DUMMY
723. //GEN11 EXEC PGM=IEBGENER
724. //SYSPRINT DD SYSOUT=A
725. //SYSUT1 DD DISP=(OLD,DELETE),DSN=*&TEMPUT
726. //SYSUT2 DD DISP=(OLD,KEEP),DSN=CN6376.MI2.NUFS.FORM1M.PDS78(UT)
727. //SYSIN DD DUMMY
```


APPENDIX B

This appendix relates to the software which interfaces the solutions of MEFS with NUFS. This process is discussed in Section III of Volume II. This appendix is divided into three sections:

i) Listing of NUFS.MEFS.FORT. This is the FORTRAN source module which creates the MEFS interface files. This module contains a main controlling routine and four major (as well as several minor) subroutines. First, Subroutine INITL2 is called. This subroutine reads in the base data file (see below). Second, subroutine STRIP is called. This subroutine strips off irrelevant information from the PERUSE solution files from MEFS. Third, subroutine EXTRAC is called and extracts the relevant information from the stripped peruse files. This information is the solution of MEFS for the years 1985, 1990, and 1995. Finally, subroutine COMPUT is called. This subroutine interpolates and extrapolates the solutions from EXTRAC to yield solutions for each year of the forecast period. In addition, it writes out this information to data files for NUFS.

The remainder of this module contains a series of job steps which create a partitioned data set whose members are the data sets for NUFS.

ii) Listing of NUFS.MEFS.BASEDATA. This file contains two types of data. First, it contains information on how MEFS categories are to be aggregated into NUFS categories. This information need never be altered except when the structure of NUFS or MEFS is changed. Second, it contains some of the assumptions that went into MEFS but could not be stripped from the solution. The primary items in this category are the new asset capital costs. This information needs to be updated as the assumptions in MEFS change (probably every calendar year). The file is self documenting.

iii) Listing of NUFS.CREMOD.DATA. MEFS solves for cumulative new generating capacity builds for the years 1985, 1990, and 1995. NUFS needs year by year builds. The purpose of this file is to determine how the cumulative builds from MEFS are to be allocated to the individual years needed by NUFS. The approach is to actually declare the year by year builds. The columns of the file are for each of the eleven DOE regions and the rows are for each combination of forecast years and plant types. The source of this file is from software entitled "A Capital Requirements Estimating Model" (CREMOD). The creation of this file is outside the scope of this manual.

i) LISTING OF NUFFS.MEFS.FORT

```

1. //IS3UMEFC JOB (6376,M20),'SHAVEL',TIME=(0,6)
2. //DELD EXEC PGM=IEFBR14,REGION=10K
3. //DD1 DD DSN=CN6376.IS3.EYNON.GG.FINANCE,DISP=(MOD,DELETE),
4. // UNIT=3350,SPACE=(TRK,(10,1,1))
5. //FPC EXEC FORTGCL,REGION=200K
6. //FORT.SYSPRINT DD SYSOUT=A
7. //FORT.SYSIN DD *
8. COMMON/SHARE/TRASH(20000)
9. INTEGER ISOLN/25/,NUMRUN/3/
10. CALL INITLZ
11. DO 10 IRUN=1,NUMRUN
12. CALL STRIP(ISOLN)
13. CALL EXTRAC(IRUN)
14. ISOLN=ISOLN+1
15. 10 CONTINUE
16. CALL COMPUT
17. STOP
18. END
19. SUBROUTINE INITLZ
20. COMMON /CENSUS/BUILDS(11,14,2,3),OPER8S(11,14,3),FUEL(11,3)
21. X ,NUCFL(11,3),OANDM(11,3)
22. REAL*4 NUCFL
23. REAL NAMREG,NWLOAD
24. INTEGER IBUFF(10)
25. COMMON/WORKSP/ NUMREG,NAMREG(10),NFUE,FUENAM(12),ZUENAM(12)
26. X ,EXLOAD(4),NWLOAD(4),NSEG,NATION,DEF79
27. X ,NBILD,BILDNM(30),NDEFER,DEFNM(3),KODE(30)
28. X ,BLDFAC(2,3),CAPITL(10,30,3,3),OFIX(30)
29. X ,DCF(30,3),NCONV,CONNAM(40),CONVTP(40),SLOSS(11,3)
30. X ,NTRANS,TRANS(5),NMISC,MISCGN(20),MISCB(20),MKODE(20)
31. X ,DCFMIS(20),TDCAP(10,6),TDOM(10,6)
32. REAL*8 TNAME
33. REAL YEARS(3)/'85','90','95'/
34. COMMON/SHARE/TNAME,ARRAY(50,99),ROWNAM(50),COLNAM(99)
35. CALL RDATA(DEF79,1,NROWS,NCOLS,ROWNAM,COLNAM,TNAME)
36. CALL PDATA(DEF79,1,NROWS,NCOLS,ROWNAM,COLNAM,TNAME)
37. CALL RDATA(ARRAY,50,NROWS,NUMREG,ROWNAM,NAMREG,TNAME)
38. C CALL PDATA(ARRAY,50,NROWS,NUMREG,ROWNAM,NAMREG,TNAME)
39. CALL RDATA(ARRAY,50,NROWS,NFUE,ROWNAM,FUENAM,TNAME)
40. C CALL PDATA(ARRAY,50,NROWS,NFUE,ROWNAM,FUENAM,TNAME)
41. CALL RDATA(ARRAY,50,NROWS,NCOLS,ROWNAM,ZUENAM,TNAME)
42. C CALL PDATA(ARRAY,50,NROWS,NCOLS,ROWNAM,ZUENAM,TNAME)
43. CALL RDATA(ARRAY,50,NROWS,NSEG,ROWNAM,EXLOAD,TNAME)
44. C CALL PDATA(ARRAY,50,NROWS,NSEG,ROWNAM,EXLOAD,TNAME)
45. CALL RDATA(ARRAY,50,NROWS,NSEG,ROWNAM,NWLOAD,TNAME)
46. C CALL PDATA(ARRAY,50,NROWS,NSEG,ROWNAM,NWLOAD,TNAME)
47. C***READ TRANSMISSION COST CODES
48. CALL RDATA(ARRAY,50,NROWS,NTRANS,ROWNAM,TRANS,TNAME)
49. C***READ MISC GEN & BLD CATEGORIES.
50. CALL RDATA(ARRAY,50,NROWS,NMISC,ROWNAM,MISCGN,TNAME)
51. CALL RDATA(ARRAY,50,NROWS,NMISC,ROWNAM,MISCB,TNAME)
52. DO 30 I=1,NMISC
53. MKODE(I)=ARRAY(1,I)+0.1
54. 30 DCFMIS(I)=ARRAY(2,I)
55. CALL RDATA(ARRAY,50,NBILD,NCOLS,BILDNM,COLNAM,TNAME)

```

```

56.      CALL PDATA(ARRAY,50,NBILD,NCOLS,BILDNM,COLNAM,TNAME)
57.      DO 40 I=1,NBILD
58.      40 KODE(I)=ARRAY(I,1)+0.1
59.      CALL RDATA(BLDFAC,2,NROWS,NDEFER,ROWNAM,DEFRNM,TNAME)
60.      CALL PDATA(BLDFAC,2,NROWS,NDEFER,ROWNAM,DEFRNM,TNAME)
61.      DO 50 I=1,9
62.      CALL RDATA(ARRAY,50,NROWS,NCOLS,ROWNAM,COLNAM,TNAME)
63.      C    CALL PDATA(ARRAY,50,NROWS,NCOLS,ROWNAM,COLNAM,TNAME)
64.      CALL LOOKUP(TNAME,DEFRNM,1,5,1,NDEFER,JDEF,&999)
65.      CALL LOOKUP(TNAME,YEARS,2,6,1,3,NYR,&999)
66.      DO 60 JROW=1,NROWS
67.      CALL LOOKUP(ROWNAM(JROW),BILDNM,1,1,1,NBILD,NPLANT,&60)
68.      DO 70 IREG=1,NCOLS
69.      CAPITL(IREG,NPLANT,JDEF,NYR)=ARRAY(JROW,IREG)
70.      70 CONTINUE
71.      60 CONTINUE
72.      50 CONTINUE
73.      CALL LOOKUP('N',BILDNM,1,1,1,NBILD,NUC,&999)
74.      DO 80 IYR=1,3
75.      CALL RDATA(ARRAY,50,NROWS,NCOLS,ROWNAM,COLNAM,TNAME)
76.      C    CALL PDATA(ARRAY,50,NROWS,NCOLS,ROWNAM,COLNAM,TNAME)
77.      DO 90 JROW=1,NROWS
78.      CALL LOOKUP(ROWNAM(JROW),DEFRNM,1,2,1,NDEFER,JDEF,&90)
79.      DO 95 IREG=1,NCOLS
80.      CAPITL(IREG,NUC,JDEF,IYR)=ARRAY(JROW,IREG)
81.      95 CONTINUE
82.      90 CONTINUE
83.      80 CONTINUE
84.      CALL RDATA(ARRAY,50,NROWS,NCOLS,ROWNAM,COLNAM,TNAME)
85.      C    CALL PDATA(ARRAY,50,NROWS,NCOLS,ROWNAM,COLNAM,TNAME)
86.      DO 100 JROW=1,NROWS
87.      CALL LOOKUP(ROWNAM(JROW),BILDNM,1,1,1,NBILD,NPLANT,&100)
88.      QFIX(NPLANT)=ARRAY(JROW,1)
89.      100 CONTINUE
90.      C****READ DCF TABLE
91.      CALL RDATA(ARRAY,50,NROWS,NCOLS,ROWNAM,COLNAM,TNAME)
92.      DO 110 JCOL=1,NCOLS
93.      CALL LOOKUP(COLNAM(JCOL),DEFRNM,1,1,1,NDEFER,JDEF,&999)
94.      110 IBUFF(JCOL)=JDEF
95.      DO 120 JROW=1,NROWS
96.      CALL LOOKUP(ROWNAM(JROW),BILDNM,1,1,1,NBILD,NPLANT,&120)
97.      DO 130 JCOL=1,NCOLS
98.      DCF(NPLANT,IBUFF(JCOL))=ARRAY(JROW,JCOL)
99.      130 CONTINUE
100.     120 CONTINUE
101.     CALL PDATA(DCF,30,NBILD,NDEFER,BILDNM,DEFRNM,TNAME)
102.     C****READ RETROFIT AND CONVERSION INFORMATION.
103.     CALL RDATA(ARRAY,50,NROWS,NCONV,ROWNAM,CONNAM,TNAME)
104.     DO 140 I=1,NCONV
105.     140 CONVTP(I)=ARRAY(1,I)
106.     C****READ T&D CAPITAL AND O&M COSTS.
107.     CALL RDATA(ARRAY,50,NROWS,NCOLS,ROWNAM,COLNAM,TNAME)
108.     DO 150 I=1,NROWS
109.     DO 160 J=1,NCOLS
110.     160 TDCAP(J,I)=ARRAY(I,J)
111.     150 CONTINUE

```

```

112.      CALL RDATA(ARRAY,50,NROWS,NCOLS,ROWNAM,COLNAM,TNAME)
113.      DO 170 I=1,NROWS
114.      DO 180 J=1,NCOLS
115.      180  TDOM(J,I)=ARRAY(I,J)
116.      170  CONTINUE
117.      PRINT 997
118.      997  FORMAT(' GOT THRU READ')
119.      CALL PDATA(OFIX,30,NBILD,1,BILDNM,'DATA','OFIX')
120.      CALL PDATA(CAPITL(1,1,2,2),10,NUMREG,NBILD,NAMREG,BILDNM
121.      X , 'CPCSZ90')
122.      RETURN
123.      999  PRINT 998,TNAME
124.      998  FORMAT(' LOOKUP ERROR ',A8)
125.      STOP
126.      END
127.      SUBROUTINE STRIP(ISOLN)
128.      C*** SUBROUTINE STRIPS IRRELEVANT INFORMATION FROM
129.      C PERUSE FILES. THE WORD 'ENDSEC' IS PLACED
130.      C AFTER THE 'M','L' ROWS AND 'R','B' COLUMNS,
131.      C RESPECTIVELY. THE RESULT IS STORED IN A TEMPORARY
132.      C FILE 'ITEMP'
133.      REAL*8 ALPHA
134.      COMMON /SHARE/ALPHA(10)
135.      REAL*8 ENDSEC/'ENDSEC'/
136.      ITEMP=20
137.      REWIND ITEMP
138.      1  FORMAT(10A8)
139.      100  READ(ISOLN,1,END=200) ALPHA
140.      C IF THE THIRD CHARACTER IN A LINE IS 'U', SAVE.
141.      CALL LOOKUP(ALPHA(1),'U',1,3,1,1,KPTR,&105)
142.      WRITE(ITEMP,1) ALPHA
143.      GO TO 100
144.      C IF THE FIFTH CHARACTER IS 'U', SAVE
145.      105  CALL LOOKUP(ALPHA(1),'U',1,5,1,1,KPTR,&130)
146.      WRITE(ITEMP,1)ALPHA
147.      GO TO 100
148.      C***IF THE STRING 'UE' BEGINS IN THE 2ND COLUMN SAVE (IMPORTS)
149.      130  CALL LOOKUP(ALPHA(1),'UE',2,2,1,1,KPTR,&150)
150.      WRITE(ITEMP,1)ALPHA
151.      GO TO 100
152.      C IF THE STRING 'NAME' STARTS IN COLUMN 4,WRITE ENDSEC,ELSE SKIP
153.      150  CALL LOOKUP(ALPHA(1),'NAME',4,4,1,1,KPTR,&100)
154.      WRITE (ITEMP,1) ENDSEC
155.      GO TO 100
156.      200  WRITE (ITEMP,1) ENDSEC
157.      C CLOSE FILE WITH ENDSEC
158.      RETURN
159.      END
160.      SUBROUTINE EXTRAC(IRUN)
161.      COMMON /OPS/ AMT(14,26,10),RATE1(14,10)
162.      COMMON /CENSUS/BUILDS(11,14,2,3),OPER8S(11,14,3),FUEL(11,3)
163.      X ,NUCFL(11,3),OANDM(11,3)
164.      REAL*4 NUCFL
165.      REAL NAMREG,NWLOAD,TDLOSS(10,3),TRNSFR(10,3)
166.      COMMON/WORKSP/ NUMREG,NAMREG(10),NFUE,FUENAM(12),ZUENAM(12)
167.      X ,EXLOAD(4),NWLOAD(4),NSEG,NATION,DEF79

```

```

168.      X      ,NBILD,BILDNM(30),NDEFER,DEFNM(3),KODE(30)
169.      X      ,BLDFAC(2,3),CAPITL(10,30,3,3),OFIX(30)
170.      X      ,DCF(30,3),NCONV,CONNAM(40),CONVTP(40),SLOSS(11,3)
171.      X      ,NTRANS,TRANS(5),NMISC,MISCGN(20),MISCBD(20),MKODE(20)
172.      X      ,DCFMS(20),TDCAP(10,6),TDOM(10,6)
173.      COMMON /SHARE/FUELP(10,12),BUFFLD(400,2,10),NBUFF(10)
174.      REAL*8 NAME,STATUS,LOLIM,UPLIM,DUAL
175.      REAL BLANK4/' '/
176.      CALL LOOKUP('O',DEFNM,1,1,1,NDEFER,NEW,&999)
177.      ITEMP=20
178.      REWIND ITEMP
179.      1 FORMAT(3X,A8,5X,A8,F15.5)
180.      C*****
181.      C *****ZERO OUT SOME ARRAYS
182.      C*****
183.      DO 10 JREG=1,NUMREG
184.      DO 13 JAY=1,3
185.      13 TRNSFR(JREG,JAY)=0.0
186.      10 NBUFF(JREG)=0
187.      NATION=NUMREG+1
188.      DO 15 JREG=1,NATION
189.      FUEL(JREG,IRUN)=0.
190.      OANDM(JREG,IRUN)=0.
191.      DO 14 NPLANT=1,14
192.      BUILDS(JREG,NPLANT,1,IRUN)=0.
193.      BUILDS(JREG,NPLANT,2,IRUN)=0.
194.      14 OPER8S(JREG,NPLANT,IRUN)=0.
195.      15 CONTINUE
196.      C*****
197.      C READ 'L *,MU*E' SECTION. THIS GIVES NAMES AND CAPACITY FACTORS FOR
198.      C ELECTRICAL GENERATORS.
199.      C*****
200.      20 READ (ITEMP,1) NAME,DUAL,ACTIVE
201.      CALL LOOKUP(NAME,'SEC',3,1,1,1,KPTR,&30)
202.      GO TO 100
203.      30 IF (ACTIVE.LE.0.) GO TO 80
204.      CALL LOOKUP(NAME,NAMREG,2,2,1,NUMREG,KPTR,&999)
205.      CALL LOOKUP(NAME,'U1',2,4,1,1,LPTR,&40)
206.      TDLOSS(KPTR,IRUN)=ACTIVE
207.      GO TO 20
208.      40 KBUFF=NBUFF(KPTR)+1
209.      NBUFF(KPTR)=KBUFF
210.      CALL MVECH(NAME,BUFFLD(KBUFF,1,KPTR),2,4,1)
211.      CALL MVECH(DUAL,BUFFLD(KBUFF,1,KPTR),2,4,3)
212.      BUFFLD(KBUFF,2,KPTR)=ACTIVE
213.      GO TO 20
214.      80 GO TO 20
215.      C*****
216.      C READ 'DISP ROW MU*' SECTION. THIS GIVES FUEL PRICES.
217.      C*****
218.      100 READ (ITEMP,2) NAME,STATUS,ACTIVE,SLACK,LOLIM,UPLIM,DUAL
219.      2 FORMAT (1X,A8,1X,A2,1X,5F12.4)
220.      CALL LOOKUP(NAME,'NDSE',4,1,1,1,KPTR,&110)
221.      GO TO 300
222.      C***** EXTRACT FUEL SHADOW PRICES FROM MATERIAL BALANCE ROWS
223.      110 CALL LOOKUP(NAME,FUEENAM,2,4,1,NFUE,KPTR,&100)

```

```

224.      CALL LOOKUP(NAME,NAMREG,2,2,1,NUMREG,JREG,&999)
225.      C      PRINT 2,NAME,STATUS,DUAL
226.      FUELP(JREG,KPTR)=-DUAL
227.      GO TO 100
228.      C      PKINT 3,(NBUFF(I),I=1,10)
229.      C      3 FORMAT(10I5)
230.      C      KBUFF=NBUFF(1)
231.      C      DO 202 I=1,KBUFF
232.      C      PRINT 2,BUFFLD(I,1,1)
233.      C      202 CONTINUE
234.      C*****
235.      C READ 'DISP COL RU*' SECTION. THIS GIVES OPERATE VALUES WHICH GIVE:
236.      C      (1)FUEL CONSUMPTION QUANTITIES, (2)ELECTRICITY CONVERSION ACTIVITY,
237.      C      (3)GENERATION TOTALS, (4)FUEL CONVERSIONS & RETROFITS, OR (5)TRANS-
238.      C      MISSION COSTS.
239.      C*****
240.      300 READ (ITEMP,2) NAME,STATUS,ACTIVE,COST,LOLIM,UPLIM,DUAL
241.      ACTIVE=ACTIVE*0.365
242.      CALL LOOKUP(NAME,'NDSE',4,1,1,1,KPTR,&310)
243.      GO TO 400
244.      310 CALL LOOKUP(NAME,NAMREG,2,2,1,NUMREG,JREG,&999)
245.      C*****(1)IS IT FUEL QUANTITY SUM? IF SO, TAB TOTAL FUELCOST
246.      CALL LOOKUP(NAME,ZUENAM,2,4,1,NFUE,KPTR,&350)
247.      FUEL(JREG,IRUN)=FUEL(JREG,IRUN)+FUELP(JREG,KPTR)*ACTIVE
248.      C***IS IT NUCLEAR FUEL? (RECORD SEPARATELY)
249.      CALL LOOKUP(NAME,'ZN',2,4,1,1,NPTR,&300)
250.      NUCFL(JREG,IRUN)=NUCFL(JREG,IRUN)+ACTIVE*FUELP(JREG,KPTR)
251.      GO TO 300
252.      C*****(2)IS IT THE ELECTRICITY CONVERSION ACTIVITY? IF SO,
253.      C      TAKE CARE OF TRANSMISSION TDLOSSES
254.      350 CALL LOOKUP(NAME,'U1',2,4,1,1,KPTR,&360)
255.      PRINT 9,JREG,ACTIVE
256.      9      FORMAT(' LP DERIVED GENERATION FOR REGION ',I2,':',F15.4)
257.      GO TO 300
258.      C *****(3)DOES IT GENERATE ELECTRICITY?
259.      360 CALL LOOKUP(NAME,BUFFLD(1,1,JREG),2,4,1,NBUFF(JREG),KPTR,&380)
260.      C      PRINT 5,NAME,BUFFLD(KPTR,1,JREG),BUFFLD(KPTR,2,JREG)
261.      C****IS IT NEW OR EXISTING?
262.      NPLANT=3
263.      CALL LOOKUP(NAME,EXLOAD,1,5,1,NSEG,LPTR,&365)
264.      NPLANT=1
265.      C****IS IT NUCLEAR OR NOT?
266.      365 CALL LOOKUP(NAME,'N',1,4,1,1,NPTR,&370)
267.      NPLANT=NPLANT+1
268.      C****(1=EXIST NON-NUC, 2=EXIST NUC, 3=NEW NON-NUC, 4=NEW NUC)
269.      370 OPERSS(JREG,NPLANT,IRUN)=OPERSS(JREG,NPLANT,IRUN)
270.      X      +ACTIVE*BUFFLD(KPTR,2,JREG)
271.      OANDM(JREG,IRUN)=OANDM(JREG,IRUN)+COST*ACTIVE
272.      GO TO 300
273.      C *****(4)IS IT A FUEL CONVERSION OR RETROFIT?
274.      380 CALL LOOKUP(NAME,CONNAM,2,4,1,NCONV,KPTR,&385)
275.      CALL LOOKUP(CONNAM(KPTR),BILDM,1,4,1,NBILD,LPTR,&999)
276.      NPLANT=CONVTP(KPTR)
277.      BUILDS(JREG,NPLANT,1,IRUN)=BUILDS(JREG,NPLANT,1,IRUN)
278.      X      +ACTIVE/.365/24.0
279.      BUILDS(JREG,NPLANT,2,IRUN)=BUILDS(JREG,NPLANT,2,IRUN)

```

```

280.      X                      +ACTIVE/.365/24.0*COST*8.76/DCF(LPTR,NEW)
281.  C****DEBUG
282.      BUFF2=ACTIVE/.365/24.0
283.      BUFF3=BUFF2*COST*8.76/DCF(LPTR,NEW)
284.      PRINT 402,JREG,CONVTP(KPTR),ACTIVE,COST,BUFF2,BUFF3
285.  402  FORMAT(' ','R:',I2,1X,F3.1,' RET',4(3X,F8.4))
286.      GO TO 300
287.  C****(5) IS IT TRANSMISSION COSTS? IF SO, FIND OUT IF ITS NEW OR OLD.
288.  385  CALL LOOKUP(NAME,TRANS,2,4,1,NTRANS,KPTR,&395)
289.      CALL LOOKUP(NAME,'N',1,5,1,1,NPTR,&390)
290.  C**NEW T&D. *****GET CONSTANT BUILD # OUT
291.      NPLANT=13
292.      ICOL=IRUN*2
293.      BUILDS(JREG,NPLANT,1,IRUN)=BUILDS(JREG,NPLANT,1,IRUN)
294.      X                      +ACTIVE/.365/24.0
295.      BUILDS(JREG,NPLANT,2,IRUN)=BUILDS(JREG,NPLANT,2,IRUN)
296.      X                      +ACTIVE/.365/24.0*TDCAP(JREG,ICOL)
297.      OANDM(JREG,IRUN)=OANDM(JREG,IRUN)+TDOM(JREG,ICOL)*ACTIVE/.365/24.0
298.  C**DEBUG
299.      COST1=.0643*ACTIVE/.365/24.0*TDCAP(JREG,ICOL)
300.      X                      +ACTIVE/.365/24.0*TDOM(JREG,ICOL)
301.      COST2=COST*ACTIVE
302.      PRINT 401,JREG,COST1,COST2
303.  401  FORMAT(' ','REGION:',I2,5X,'CALC. COST:',F15.2,5X,'LP COST:',
304.      X F15.2)
305.      GO TO 300
306.  C**OLD T&D. CURRENTLY IGNORE.*****CHANGE THIS*****
307.  C390  OANDM(JREG,IRUN)=OANDM(JREG,IRUN)+COST*ACTIVE
308.  390  GO TO 300
309.      5 FORMAT(' GENERATE,NAMES,LOAD ',2(A8,1X),F10.5)
310.      395 PRINT 6,NAME,COST,ACTIVE
311.      6 FORMAT(' MYSTERY:COST,ACTIV ',A8,2F12.4)
312.      GO TO 300
313.  C*****
314.  C READ 'DISP COL BU*' SECTION. THESE ARE BUILD ACTIVITIES.
315.  C*****
316.      400 READ (ITEMP,2) NAME,STATUS,ACTIVE,COST,LOLIM,UP LIM,DUAL
317.      ACTIVE=ACTIVE/24.
318.      CALL LOOKUP(NAME,'NDSE',4,1,1,1,KPTR,&410)
319.      GO TO 500
320.      410 CALL LOOKUP(NAME,NAMREG,2,2,1,NUMREG,JREG,&999)
321.      CALL LOOKUP(NAME,DILDNM,1,4,1,NBILD,JBILD,&440)
322.      NCODE=KODE(JBILD)
323.      CALL LOOKUP(NAME,DEFRNM,1,5,1,NDEFER,JDEFR,&499)
324.      DEFFAC=BLDFAC(1,JDEFR)
325.      CALL LOOKUP(NAME,'N',1,4,1,1,KPTR,&430)
326.      DEFFAC=BLDFAC(2,JDEFR)
327.      430 BUILDS(JREG,NCODE,1,IRUN)=BUILDS(JREG,NCODE,1,IRUN)+ACTIVE
328.      BUILDS(JREG,NCODE,2,IRUN)=BUILDS(JREG,NCODE,2,IRUN)
329.      X                      +ACTIVE*DEFFAC*CAPITL(JREG,JBILD,JDEFR,IRUN)
330.      OANDM(JREG,IRUN)=OANDM(JREG,IRUN)+ACTIVE*QFIX(JBILD)
331.      GO TO 400
332.  C***IS IT A MISC BLD CATEGORY? *****CHANGE .876 IN BUILDS2***
333.  440  CALL LOOKUP(NAME,MISCBD,2,4,1,NMISC,JBILD,&499)
334.      NCODE=MKODE(JBILD)
335.      BUILDS(JREG,NCODE,1,IRUN)=BUILDS(JREG,NCODE,1,IRUN)+ACTIVE

```



```

336.      BUILDS(JREG,NCODE,2,IRUN)=BUILDS(JREG,NCODE,2,IRUN)
337.      X   +ACTIVE*COST*8.76/DCFMIS(JBILD)
338.      GO TO 400
339.      499 PRINT 2, NAME,STATUS,ACTIVE,COST
340.      GO TO 400
341.      C*****
342.      C READ 'DISP COL UE***U' SECTION. THESE ARE IMPORTS AND INTERREGIONAL
343.      C TRANSFERS.
344.      C*****
345.      500 READ (ITEMP,2) NAME,STATUS,ACTIVE,COST,LOLIM,UPLIM,DUAL
346.      ACTIVE=ACTIVE*.365
347.      CALL LOOKUP(NAME,'NDSE',4,1,1,1,KPTR,&510)
348.      GO TO 600
349.      510 CALL LOOKUP(NAME,NAMREG,2,6,1,NUMREG,IBREG,&999)
350.      C***CHECK FOR IMPORT.
351.      CALL LOOKUP(NAME,'FC',2,4,1,1,NPTR,&550)
352.      C**IMPORT TRANSFER TO REGION IBREG.
353.      FUEL(IBREG,IRUN)=FUEL(IBREG,IRUN)+COST*ACTIVE
354.      TRNSFR(IBREG,IRUN)=TRNSFR(IBREG,IRUN)+ACTIVE
355.      GO TO 500
356.      550 CALL LOOKUP(NAME,NAMREG,2,4,1,NUMREG,ISREG,&999)
357.      C**INTER-REGIONAL TRANSFER FROM REGION ISREG TO REGION IBREG.
358.      CALL LOOKUP('E1',FUENAM,2,1,1,NFUE,IELEC,&999)
359.      CHARGE=(FUELP(IBREG,IELEC)+FUELP(ISREG,IELEC))/2.0
360.      FUEL(ISREG,IRUN)=FUEL(ISREG,IRUN)-CHARGE*ACTIVE
361.      FUEL(IBREG,IRUN)=FUEL(IBREG,IRUN)+CHARGE*ACTIVE
362.      TRNSFR(ISREG,IRUN)=TRNSFR(ISREG,IRUN)-ACTIVE
363.      TRNSFR(IBREG,IRUN)=TRNSFR(IBREG,IRUN)+ACTIVE
364.      GO TO 500
365.      C*****
366.      C DO NATIONAL AGGREGATIONS AND CALCULATE THE LOSSES WHICH OCCUR FROM
367.      C GENERATION TO SALES, SLOSS. (IE. TO LOSSES AND TRANSFERS)
368.      C*****
369.      600 SALESN=0.0
370.      DO 615 JREG=1,NUMREG
371.      GENER=0.0
372.      FUEL(NATION,IRUN)=FUEL(NATION,IRUN)+FUEL(JREG,IRUN)
373.      NUCFL(NATION,IRUN)=NUCFL(NATION,IRUN)+NUCFL(JREG,IRUN)
374.      OANDM(NATION,IRUN)=OANDM(NATION,IRUN)+OANDM(JREG,IRUN)
375.      DO 614 NPLANT=1,14
376.      BUILDS(NATION,NPLANT,1,IRUN)=BUILDS(NATION,NPLANT,1,IRUN)
377.      X   +BUILDS(JREG,NPLANT,1,IRUN)
378.      BUILDS(NATION,NPLANT,2,IRUN)=BUILDS(NATION,NPLANT,2,IRUN)
379.      X   +BUILDS(JREG,NPLANT,2,IRUN)
380.      OPER8S(NATION,NPLANT,IRUN)=OPER8S(NATION,NPLANT,IRUN)
381.      X   +OPER8S(JREG,NPLANT,IRUN)
382.      614 GENER=GENER+OPER8S(JREG,NPLANT,IRUN)
383.      SALES=TDLOSS(JREG,IRUN)*(GENER+TRNSFR(JREG,IRUN))
384.      SLOSS(JREG,IRUN)=SALES/GENER
385.      ADJGEN=GENER+TRNSFR(JREG,IRUN)
386.      PRINT 8,JREG,ADJGEN
387.      8   FORMAT(' CALCULATED ADJUSTED GENERATION FOR REGION ',I2,' ',F15.4)
388.      SALESN=SALESN+SALES
389.      615 CONTINUE
390.      GENER8=0.0
391.      DO 620 NPLANT=1,14

```

```

392. 620 GENER8=GENER8+OPER8S(NATION,NPLANT,IRUN)
393. SLOSS(NATION,IRUN)=SALESN/GENER8
394. DO 625 JREG=1,NATION
395. 625 SLOSS(JREG,IRUN)=1.0-SLOSS(JREG,IRUN)
396. DO 700 JREG=1,10
397. PRINT 11,JREG,TRNSFR(JREG,IRUN)
398. 11 FORMAT(' NET TRANSFERS FOR REGION ',I2,':','IS',F10.4)
399. 700 CONTINUE
400. RETURN
401. 999 PRINT 998,NAME
402. 998 FORMAT(1X,'LOOKUP ERROR ',A8)
403. STOP
404. END
405. SUBROUTINE COMPUT
406. COMMON /CENSUS/BUILDS(11,14,2,3),OPER8S(11,14,3),FUEL(11,3)
407. X ,NUCFL(11,3),OANDM(11,3)
408. REAL*4 NUCFL
409. REAL NAMREG,NWLOAD
410. COMMON/WORKSP/ NUMREG,NAMREG(10),NFUE,FUENAM(12),ZUENAM(12)
411. X ,EXLOAD(4),NWLOAD(4),NSEQ,NATION,DEF79
412. X ,NBILD,BILDNM(30),NDEFER,DEFNM(3),KODE(30)
413. X ,BLDFAC(2,3),CAPITL(10,30,3,3),OFIX(30)
414. X ,DCF(30,3),NCONV,CONNAM(40),CONVTP(40),SLOSS(11,3)
415. X ,NTRANS,TRANS(5),NMISC,MISCGN(20),MISCB(20),MKODE(20)
416. X ,DCFNIS(20),TDCAP(10,6),TDOM(10,6)
417. INTEGER ISPAN(4)/7,5,5,10/
418. INTEGER KSPAN(4)/10,5,5,10/
419. INTEGER LSPAN(4)/6,5,5,10/
420. REAL DELTA(2,3)
421. REAL SHARES(28),TOTNEW(18),GENNEW(10)
422. REAL POINTS(4,3),GROW(4,3)
423. REAL GENER8(18,2),TOTGEN(18)
424. REAL*4 BUFF(14),BUFF1(5),RATE(4,4)
425. COMMON /OPS/ AMT(14,26,10),RATE1(14,10)
426. REAL USGEN(18,14),USBILD(28,14),USTOTG(18)
427. C*****CALL RDCREM TO READ THE CREMOD DATA
428. C PRINT 9997
429. 9997 FORMAT('READ CREMOD DATA')
430. CALL RDCREM
431. C PRINT 9995
432. 9995 FORMAT('RETURN FROM RDCREM')
433. CALL DEBUG(BUILDS,AMT)
434. DO 9998 L=1,10
435. WRITE(50,9996) L
436. 9996 FORMAT('//REGION=',I3)
437. DO 9998 IPLT=1,14
438. 9998 WRITE(50,9999) (BUILDS(L,IPLT,1,NUM),NUM=1,3)
439. 9999 FORMAT(3(F10.5,4X))
440. C*****
441. C INITIALIZE MATRICES TO ZERO.
442. C*****
443. DO 790 J=1,18
444. TOTNEW(J)=0.0
445. SHARES(J)=0.0
446. USTOTG(J)=0.
447. DO 791 KAY=1,14

```

```

448.      USGEN(J,KAY)=0.
449.      791 USBILD(J,KAY)=0.
450.      790 CONTINUE
451.      DO 793 J=19,28
452.      DO 794 KAY=1,14
453.      794 USBILD(J,KAY)=0.0
454.      793 CONTINUE
455.      C*****PRINT DEBUG CHECKS.
456.      DO 50 I=1,NATION
457.      PRINT 104,I
458.      104 FORMAT(' REGION ',I5)
459.      DO 792 NPLANT=1,14
460.      792 PRINT 101, ((BUILDS(I,NPLANT,K,IRUN),K=1,2),IRUN=1,3)
461.      101 FORMAT(' BUILDS',6F14.4)
462.      DO 51 J=1,4
463.      PRINT 102, (OPER8S(I,J,IRUN),IRUN=1,3)
464.      102 FORMAT(' OPERATES',6F14.4)
465.      51 CONTINUE
466.      PRINT 103, (FUEL(I,IRUN),IRUN=1,3),(OANDM(I,IRUN),IRUN=1,3)
467.      103 FORMAT(' FUEL COST ',3F20.4/' OANDM COST',3F20.4)
468.      50 CONTINUE
469.      C*****
470.      C INTERPOLATE EXISTING (NUC & NON-NUC) AND TOTAL NEW GENERATION BY YEAR.
471.      C THIS WILL BE FIRST USED TO INTERPOLATE BUILDS BY YEAR, THEN TO INTER-
472.      C POLATE FINER CATEGORIES OF NEW GENERATION.
473.      C *****RULES FIRST COMPUTE TOTAL GENERATION USING
474.      C EXPONENTIAL GROWTH RATE
475.      C ***** FOR OPERATE TYPES 1 AND 2, USE LINEAR INTERPOLATION
476.      C ***** COMPUTE NEW GENERATION BY SUBTRACTING EXISTING
477.      C FROM THE TOTAL
478.      C** LOOP ON REGIONS
479.      C*****
480.      DO 200 IREG=1,NATION
481.      KFILE=IREG+29
482.      C** FIRST COMPUTE TOTAL GENERATION
483.      DO 202 IRUN=2,4
484.      DO 201 KAY=1,2
485.      POINTS(IRUN,KAY)=OPER8S(IREG,KAY,IRUN-1)
486.      201 CONTINUE
487.      POINTS(IRUN,3)=0.
488.      DO 202 J=1,14
489.      202 POINTS(IRUN,3)=POINTS(IRUN,3)+OPER8S(IREG,J,IRUN-1)
490.      C**** SET BASE YEAR GENERATION EQUAL TO 1985 EXISTING GENERATION.
491.      POINTS(1,1)=POINTS(2,1)
492.      POINTS(1,2)=POINTS(2,2)
493.      POINTS(1,3)=POINTS(1,1)+POINTS(1,2)
494.      IF (IREG.EQ.NATION) GO TO 720
495.      DO 204 K=1,3
496.      DO 203 J=1,2
497.      203 GROW(K,J)=(POINTS(K+1,J)-POINTS(K,J))/KSPAN(K)
498.      204 GROW(K,3)=ALOG(POINTS(K+1,3)/POINTS(K,3))/KSPAN(K)
499.      WRITE(65,999) (POINTS(L,3),L=1,4)
500.      999 FORMAT(4(F10.2,2X))
501.      JAY=1
502.      DO 205 IRUN=1,3
503.      SUM=0.

```

```

504.      IYR=ISPAN(IRUN)
505.      JYR=KSPAN(IRUN)-ISPAN(IRUN)
506.      IBEGYR=JAY
507.      DO 206 KYR=1,IYR
508.      SUMEX=0.
509.      JYR=JYR+1
510.      JAY=JAY+1
511.      DO 208 J=1,2
512.      GENER8(JAY,J)=POINTS(IRUN,J)+GROW(IRUN,J)*JYR
513.      208 SUMEX=SUMEX+GENER8(JAY,J)
514.      TOTNEW(JAY)=POINTS(IRUN,3)*EXP(GROW(IRUN,3)*JYR)-SUMEX
515.      TOTGEN(JAY)=TOTNEW(JAY)+SUMEX
516.      USTOTG(JAY)=USTOTG(JAY)+TOTGEN(JAY)
517.      IF (TOTNEW(JAY).LE.0.) TOTNEW(JAY)=0.
518.      206 CONTINUE
519.      205 CONTINUE
520.      C*****
521.      C NOW, BASED ON NEW GENERATION BY YEAR, INTERPOLATE BUILDS BY YEAR.
522.      C   FOR THE FIRST 12 YEARS CALL DOBLD. TAKE CARE TO MAKE
523.      C   SURE THAT THE CORRECT THING IS DONE WITH ZERO BLDS.
524.      C   IF CREMOD IS CONSISTENT THERE SHOULD BE NO PROBLEM.
525.      C*****
526.      JAY=1
527.      DO 400 IRUN=1,3
528.      TOTAL=0.0
529.      IYR=LSPAN(IRUN)
530.      DO 410 KYR=1,IYR
531.      JAY=JAY+1
532.      SHARES(JAY)=TOTNEW(JAY)-TOTNEW(JAY-1)
533.      410 TOTAL=TOTAL+SHARES(JAY)
534.      C***NOW NORMALIZE SHARES.
535.      IBEGYR=JAY-ISPAN(IRUN)+1
536.      DO 420 KYR=IBEGYR,JAY
537.      420 SHARES(KYR)=SHARES(KYR)/TOTAL
538.      400 CONTINUE
539.      PRINT 701,(SHARES(JAY),JAY=2,23)
540.      701 FORMAT(' SHARES ',18F5.2)
541.      PRINT 702,(TOTNEW(JAY),JAY=2,18)
542.      702 FORMAT(' NEWG ',18F6.1)
543.      720 DO 210 J=1,14
544.      IF (BUILDS(IREG,J,1,3).LE.0.001) GO TO 210
545.      DO 213 KAY=1,2
546.      DELTA(KAY,1)=BUILDS(IREG,J,KAY,1)
547.      DO 211 K=2,3
548.      DELTA(KAY,K)=BUILDS(IREG,J,KAY,K)-BUILDS(IREG,J,KAY,K-1)
549.      211 CONTINUE
550.      213 CONTINUE
551.      JAY=1
552.      DO 212 IRUN=1,3
553.      IYR=LSPAN(IRUN)
554.      IF (DELTA(1,IRUN).LT.0.001) GO TO 712
555.      AVEC=ABS(DELTA(2,IRUN)/DELTA(1,IRUN))
556.      712 DO 214 L=1,IYR
557.      JAY=JAY+1
558.      IF (DELTA(1,IRUN).LT.0.001) GO TO 214
559.      IF (IREG.NE.NATION) GO TO 721

```

```

560.      Q=USBILD(JAY,J)
561.      GO TO 722
562. 721 JAY1=JAY-1
563.      IF(J.GE.13.OR.JAY1.GT.12) GO TO 723
564.      CALL DOBLD(J,JAY1,IREG,Q)
565.      IF(JAY1.EQ.12) DEL=(Q-SHARES(JAY)*DELTA(1,IRUN)*1000.)/4.
566.      GO TO 724
567. 723 Q=SHARES(JAY)*DELTA(1,IRUN)*1000.
568.      IF(J.GE.13) DEL=0.
569.      IF(JAY1.LE.12) DEL=0.
570.      IF(JAY1.GE.17) DEL=0.
571.      WRITE(98,691)JAY1,IREG,J,DEL
572. 691 FORMAT('JAY1=',I3,2X,'IREG=',I3,2X,'IPLT=',I3,'DEL=',E12.4)
573.      Q=Q-DEL
574. 724 USBILD(JAY,J)=USBILD(JAY,J)+Q
575. 722 BUFF(1)=AVEC*DEF79
576.      IF(Q.LE.0.) GO TO 214
577.      WRITE(KFILE,105) JAY,J,Q ,BUFF(1)
578. 105 FORMAT(2I5,F20.3,F20.3)
579. 214 CONTINUE
580. 212 CONTINUE
581. C*****
582. C**NOW EXTRAPOLATE BUILDS TO YEAR 2005
583. C**FIRST EXTRAPOLATE SHARES TO YEAR 2005, ALSO CALCULATE FACTOR.
584. C**FACTOR IS THE RATIO OF NEW GEN FROM 95-2005 TO THAT OF 90-95.
585. C**TOTAL GEN IS ASSUMED TO GROW AT SAME RATE AS IT DID IN 90-95
586. C*****
587.      IRUN=4
588.      JAY1=JAY
589.      IYR=LSPAN(IRUN)
590.      DO 300 J1=1,IYR
591. 300 GENNEW(J1)=EXP(GROW(3,3)*J1)-1.0
592.      AOLD=0.0
593.      DO 310 J1=1,IYR
594.      JAY=JAY+1
595.      SHARES(JAY)=(GENNEW(J1)-AOLD)/GENNEW(IYR)
596.      AOLD=GENNEW(J1)
597. 310 CONTINUE
598.      FACTOR=GENNEW(IYR)*TOTGEN(JAY1)/(POINTS(4,3)-POINTS(3,3))
599.      IF (DELTA(1,3) .LT. .001) GO TO 210
600.      DO 320 J1=1,IYR
601.      JAY1=JAY1+1
602.      IF (IREG .NE. NATION) GO TO 321
603.      Q=USBILD(JAY1,J)
604.      GO TO 322
605. 321 Q=FACTOR*DELTA(1,3)*SHARES(JAY1)*1000.0
606.      USBILD(JAY1,J)=USBILD(JAY1,J)+Q
607. 322 WRITE(KFILE,105) JAY1,J,Q,BUFF(1)
608. C 322 CONTINUE
609. 320 CONTINUE
610. 210 CONTINUE
611.      JAY=0
612.      BUFF(1)=0.
613.      WRITE(KFILE,105) JAY,JAY,BUFF(1),BUFF(1)
614. C*****
615. C NOW FINER CATEGORIES OF NEW GENERATION ARE INTERPOLATED BY YEAR.

```

```

616. C *****SUMEND IS TOTAL GENERATION BY NEW PLANTS AT END OF PERIOD
617. C *****SUMBEG IS TOTAL GENERATION BY NEW PLANTS AT BEGINNING OF PERIOD
618. C*****
619.     JAY=1
620.     SUMEND=0.
621.     DO 220 IRUN=1,3
622.     SUMBEG=SUMEND
623.     SUMEND=0.
624.     DO 226 J=3,14
625. 226 SUMEND=SUMEND+OPER8S(IREG,J,IRUN)
626.     KYR=ISPAN(IRUN)
627.     DO 225 IYR=1,KYR
628.     TFAC1=FLOAT(IYR)/FLOAT(KYR)
629.     TFAC2=1.-TFAC1
630.     JAY=JAY+1
631.     IF (IREG.NE.NATION) GO TO 224
632.     DO 223 J=1,14
633. 223 BUFF(J)=USGEN(JAY,J)
634.     GO TO 229
635. 224 BUFF(1)=GENER8(JAY,1)
636.     BUFF(2)=GENER8(JAY,2)
637. C*****NEW GENERATION IS ALLOCATED BY LINEAR INTERPOLATION
638. C *****OF SHARES OF THE TOTAL NEW. THE TOTAL ITSELF
639. C *****GROWS GEOMETRICALLY
640.     DO 227 J=3,14
641.     ENDP=OPER8S(IREG,J,IRUN)/SUMEND
642.     BEGP=ENDP
643.     IF (IRUN.GT.1) BEGP=OPER8S(IREG,J,IRUN-1)/SUMBEG
644.     RATIO=TFAC1*ENDP+TFAC2*BEGP
645. 227 BUFF(J)=RATIO*TOTNEW(JAY)
646.     DO 228 J=1,14
647. 228 USGEN(JAY,J)=USGEN(JAY,J)+BUFF(J)
648. 229 BUFF1(1)=BUFF(1)+BUFF(2)+BUFF(3)+BUFF(4)
649.     BUFF1(2)=BUFF(2)+BUFF(4)
650.     WRITE(KFILE,106) (BUFF1(J),J=1,2)
651. 106 FORMAT(7F11.3)
652. 225 CONTINUE
653. 220 CONTINUE
654. C*****
655. C***** FIND GROWTH RATES FOR FUEL AND O&M. 1979-1985 WILL REMAIN
656. C     CONSTANT.
657.     DO 1230 IRUN=2,3
658.     KYR=ISPAN(IRUN)
659.     ALPHA=1./FLOAT(KYR)
660.     RATE(1,IRUN)=(ADS(FUEL(IREG,IRUN)/POINTS(IRUN+1,3))/ABS(FUEL(
661. + IREG,IRUN-1)/POINTS(IRUN,3)))*(ALPHA)
662.     RATE(2,IRUN)=((NUCFL(IREG,IRUN)/(OPER8S(IREG,2,IRUN)+OPER8S
663. + (IREG,4,IRUN)))/(NUCFL(IREG,IRUN-1)/(OPER8S(IREG,2,IRUN-1)+
664. + OPER8S(IREG,4,IRUN-1))))*(ALPHA)
665. C     RATE(2,IRUN)=1.
666.     RATE(3,IRUN)=((OANDM(IREG,IRUN)/POINTS(IRUN+1,3))/(OANDM(IREG,IRUN
667. + -1)/POINTS(IRUN,3)))*(ALPHA)
668. 1230 CONTINUE
669.     DO 1236 I=1,3
670. 1236 RATE(I,1)=1.
671.     DO 998 L=1,3

```

```

672.      998 WRITE(65,997) (RATE(L,K),K=1,3)
673.      997 FORMAT(3(F10.2,3X))
674.      C*****
675.      C NOW CALCULATE FUEL & OANDM COSTS PER KW.
676.          JAY=1
677.          IRUN=1
678.          KYR=ISPAN(IRUN)
679.          DO 2235 IYR=1,KYR
680.              JAY=JAY+1
681.              BUFF(1)=FUEL(IREG,IRUN)/POINTS(IRUN+1,3)*DEF79
682.              BUFF(2)=(NUCFL(IREG,IRUN)/(OPER8S(IREG,2,IRUN)+OPER8S(IREG,4,
683.              + IRUN)))*DEF79
684.              BUFF(3)=OANDM(IREG,IRUN)/POINTS(IRUN+1,3)*DEF79
685.              WRITE(KFILE,106) BUFF(1),BUFF(2),BUFF(3)
686.      2235 CONTINUE
687.      C*****
688.          JAY=1
689.          DO 230 IRUN=2,3
690.              KYR=ISPAN(IRUN)
691.              DO 235 IYR=1,KYR
692.                  JAY=JAY+1
693.                  BUFF(1)=FUEL(IREG,IRUN-1)/POINTS(IRUN,3)*DEF79*(RATE(1,IRUN)**IYR)
694.                  BUFF(2)=NUCFL(IREG,IRUN-1)/(OPER8S(IREG,2,IRUN-1)+OPER8S(IREG,4,
695.                  + IRUN-1))*DEF79*(RATE(2,IRUN)**IYR)
696.                  BUFF(3)=OANDM(IREG,IRUN-1)/POINTS(IRUN,3)*DEF79*(RATE(3,IRUN)**
697.                  + IYR)
698.                  WRITE(KFILE,106) BUFF(1),BUFF(2),BUFF(3)
699.      235 CONTINUE
700.      230 CONTINUE
701.      C*****
702.      C NOW ESTIMATE AND PRINT OUT CAPACITY BY YEAR.
703.      CCC*****
704.          CC      JAY=1
705.          CC      DO 240 IRUN=1,3
706.          C      KYR=ISPAN(IRUN)
707.          CC      DO 245 IYR=1,KYR
708.          CC      JAY=JAY+1
709.          CC      IF (IREG.EQ.NATION) TOTGEN(JAY)=USTOTG(JAY)
710.          CC      BUFF(1)=TOTGEN(JAY)*1.75/8.76
711.          CC      WRITE(KFILE,106) BUFF(1)
712.          CC      245 CONTINUE
713.          CC      240 CONTINUE
714.          CCC*****
715.          CCC NOW ESTIMATE & PRINT OUT PEAK DEMAND BY YEAR.
716.          CC*****
717.          C      JAY=1
718.          C      DO 250 IRUN=1,3
719.          C      KYR=ISPAN(IRUN)
720.          C      DO 255 IYR=1,KYR
721.          C      JAY=JAY+1
722.          C      BUFF(1)=TOTGEN(JAY)*2.2/8.76
723.          C      WRITE(KFILE,106) BUFF(1)
724.          C      255 CONTINUE
725.          C      250 CONTINUE
726.          WRITE(KFILE,107) (SLOSS(IREG,J),J=1,3)
727.      107  FORMAT(3(F7.4,1X))

```

```

728.      ENDFILE KFILE
729.      REWIND KFILE
730.      200 CONTINUE
731.      RETURN
732.      END
733.      SUBROUTINE LOOKUP(KEY,ARRAY,NK,NOFFK,NOFFA,NREC,KPTR,*)
734.      C****THIS SUBROUTINE IS USED TO FIND THE FIRST OCCURENCE OF
735.      C      OF A MATCH OF THE CHARACTER STRING IN THE WORD,KEY,
736.      C      WITH AN ELEMENT IN THE ARRAY,ARRAY.  THE SUBROUTINE
737.      C      RETURNS A VALUE FOR KPTR, THE POSITION IN THE
738.      C      ARRAY FOR WHICH A MATCH OCCURS.  IF NO MATCH OCCURS,
739.      C      THE CALLING PROGRAM JUMPS TO STATEMENT NUMBER *
740.      C      KPTR IS THE ONLY ARGUMENT VALUE CHANGED.
741.      C      THE SUBROUTINE CAN BE USED TO PERFORM SIMPLE COMPARISONS
742.      C      BETWEEN A STRING AND A LITERAL BY LETTING ARRAY
743.      C      BE A SINGLE WORD
744.      C      KEY      REAL*8 WORD CONTAINING CHAR. STRING
745.      C      ARRAY    REAL*4 ARRAY CONTAINING WORDS IN WHICH MATCH IS SOUGHT
746.      C      NK      INTEGER*4  NUMBER OF CHARACTERS IN STRING
747.      C      NOFFK   INTEGER*4  STARTING POSITION OF STRING IN KEY
748.      C                      (FROM THE LEFT)
749.      C      NOFFA   INTEGER*4  STARTING POSITION OF STRING IN WORDS OF ARRAY
750.      C      NREC    INTEGER*4  NUMBER OF WORDS IN ARRAY TO BE SEARCHED
751.      C      KPTR    INTEGER*4  ADDRESS IN ARRAY OF FIRST WORD
752.      C                      FOR WHICH MATCH OCCURS
753.      C      *      STATEMENT NUMBER IN MAIN TO JUMP IF FAILURE
754.      REAL*4 ARRAY(1)
755.      REAL*8 KEY,KEY1
756.      REAL*8 ARRY1
757.      REAL*8 BLANK8/' '/
758.      KEY1=BLANK8
759.      CALL MVECH(KEY,KEY1,NK,NOFFK,1)
760.      DO 50 KPTR=1,NREC
761.      ARRY1=BLANK8
762.      CALL MVECH(ARRAY(KPTR),ARRY1,NK,NOFFA,1)
763.      IF (KEY1.EQ.ARRY1) GO TO 100
764.      50 CONTINUE
765.      RETURN 1
766.      100 RETURN
767.      END
768.      SUBROUTINE MVECH(SRCE1,TARG1,NCHS,NF1,NT1)
769.      C*** THIS SUBROUTINE IS USED TO MOVE CHARACTER STRINGS FROM
770.      C      THE ARGUMENT SRCE1 INTO THE ARGUMENT TARG1
771.      C      IT IS USED BY THE SUBROUTINE LOOKUP
772.      C      SRCE1    REAL*8 WORD CONTAINING CHARACTERS TO BE
773.      C                      MOVED
774.      C      TARG1    REAL*8 WORD WHICH IS DESTINATION OF CHARACTERS
775.      C      NCHS     INTEGER*4  THE NUMBER OF CHARS TO BE MOVED
776.      C      NF1      INTEGER*4  THE STARTING POSITION IN SRC1
777.      C                      (FROM THE LEFT)
778.      C      NT1      INTEGER*4  THE STARTING POSITION IN TARG1
779.      REAL*8 TARGET
780.      REAL*8 SOURCE
781.      REAL*8 SRCE1,TARG1
782.      LOGICAL*1 A1(8),A2(8)

```



```

783.      EQUIVALENCE (TARGET,A2(1)),(SOURCE,A1(1))
784.      SOURCE=SRCE1
785.      TARGET=TARG1
786.      IEND=NCHS+NF1-1
787.      NJ1=NT1-1
788.      DO 10 I=NF1,IEND
789.      NJ1=NJ1+1
790. 10    A2(NJ1)=A1(I)
791.      TARG1=TARGET
792.      RETURN
793.      END
794.      SUBROUTINE RDATA(DUMMY,ROWDIM,NROWS,NCOLS,ROWNAM,COLNAM,TNAME)
795. C**** THIS SUBROUTINE READS TABLES OR LISTS
796. C ***TO CALL
797. C      DUMMY      ARRAY INTO WHICH NUMERIC DATA IS READ
798. C      ROWDIM     THE FIRST DIMENSION OF DUMMY
799. C      NROWS      THE NUMBER OF ROWS
800. C      NCOLS      THE NUMBER OF COLUMNS
801. C      ROWNAM     LIST OF NAMES OF THE ROWS
802. C      COLNAM     LIST OF NAMES OF COLUMNS
803. C      TNAME      THE NAME OF THE TABLE OR LIST
804. C ****ALL CALLING ARGUMENT VALUES ARE CHANGED BY THE
805. C      SUBROUTINE EXCEPT ROWDIM
806. C      1 FORMAT(2I2,6X,A8)      FORM FOR READING NAME CARD
807. C      2 FORMAT(22X,5(A4,6X))   FORM FOR READING COLUMN NAMES
808. C                               FOR A TABLE
809. C      3 FORMAT(9X,A4,4X,5F10.4) FORM FOR READING TABLE
810. C                               ROW NAME,5 ENTRIES
811. C      4 FORMAT ((9X,10(A4,1X))) FORM FOR READING LIST
812. C      INTEGER ROWDIM
813. C      REAL*8 TNAME
814. C      REAL DUMMY(ROWDIM,1),ROWNAM(1),COLNAM(1)
815. C      DATA STAR/'*'/
816. C*** READ THE NUMBER OF ROWS,NUMBER OF COLS AND NAME
817. 10    READ(16,5) BUFF
818.      IF (BUFF .EQ. STAR) GO TO 10
819.      BACKSPACE 16
820. 5     FORMAT(A1)
821.      READ (16,1) NROWS,NCOLS,TNAME
822.      1 FORMAT(2I2,6X,A8)
823. C****IF THE NUMBER OF ROWS=0 IT IS A LIST;OTHERWISE TABLE
824.      IF (NROWS.EQ.0) GO TO 200
825.      NGROUP=(NCOLS-1)/5+1
826.      DO 150 L=1,NGROUP
827.      JMIN=5*(L-1)+1
828.      JMAX=JMIN+4
829.      IF (JMAX.GT.NCOLS) JMAX=NCOLS
830. 20    READ(16,5) BUFF
831.      IF (BUFF .EQ. STAR) GO TO 20
832.      BACKSPACE 16
833.      READ (16,2) (COLNAM(J),J=JMIN,JMAX)
834.      2 FORMAT(22X,5(A4,6X))
835.      DO 100 I=1,NROWS
836. 30    READ(16,5) BUFF
837.      IF (BUFF .EQ. STAR) GO TO 30
838.      BACKSPACE 16

```

```

839.      READ (16,3) ROWNAM(I),(DUMMY(I,J),J=JMIN,JMAX)
840.      3 FORMAT(9X,A4,4X,5F10.4)
841.      100 CONTINUE
842.      150 CONTINUE
843.      RETURN
844.      C*****IT IS A LIST
845.      200 READ (16,4) (COLNAM(J),J=1,NCOLS)
846.      40 READ(16,5) BUFF
847.      IF (BUFF.EQ. STAR) GO TO 40
848.      BACKSPACE 16
849.      4 FORMAT ((9X,10(A4,1X)))
850.      RETURN
851.      END
852.      SUBROUTINE PDATA(DUMMY,ROWDIM,NROWS,NCOLS,ROWNAM,COLNAM,TNAME)
853.      C *****THIS SUBROUTINE PRINTS TABLES OR LISTS
854.      C      IT IS CALLED WITH THE SAME ARGUMENTS AS RDATA
855.      C      NONE OF THE ARGUMENT VALUES ARE CHANGED
856.      INTEGER ROWDIM
857.      REAL*8 TNAME
858.      REAL DUMMY(ROWDIM,1),ROWNAM(1),COLNAM(1)
859.      PRINT 1, NROWS,NCOLS,TNAME
860.      1 FORMAT(/1X,2I2,6X,A8)
861.      IF (NROWS.EQ.0) GO TO 200
862.      NGROUP=(NCOLS-1)/5+1
863.      DO 150 L=1,NGROUP
864.      JMIN=5*(L-1)+1
865.      JMAX=JMIN+4
866.      IF (JMAX.GT.NCOLS) JMAX=NCOLS
867.      PRINT 2, (COLNAM(J),J=JMIN,JMAX)
868.      2 FORMAT(1X,9X,'*',12X,5(A4,6X))
869.      DO 100 I=1,NROWS
870.      PRINT 3, ROWNAM(I),(DUMMY(I,J),J=JMIN,JMAX)
871.      3 FORMAT(1X,9X,A4,4X,5F10.4)
872.      100 CONTINUE
873.      150 CONTINUE
874.      RETURN
875.      200 PRINT 4,(COLNAM(J),J=1,NCOLS)
876.      4 FORMAT ((10X,10(A4,1X)))
877.      RETURN
878.      END
879.      SUBROUTINE DOBLD(IPLT,IYR,IREF,QTY)
880.      C*****
881.      C      THIS ROUTINE OUTPUTS THE NEW
882.      C      BUILDS FROM 1980 TO 1995. THE 1980-1991 NUMBERS ARE ACTUALS
883.      C      AND THE 1992-1995 NUMBERS ARE INTERPOLATED FROM 1991 CREMOD
884.      C      NUMBERS AND 1995 MEFS NUMBERS. THE 1996-2005 BUILDS ARE
885.      C      HANDLED BY THE ORIGINAL CODE AND ARE BASED ON THE GROWTH RATE
886.      C      CALCULATED FOR GENERATION BETWEEN 1990 AND 1995.
887.      C*****
888.      COMMON /OPS/ AMT(14,26,10),RATE1(14,10)
889.      REAL*4 CAPCST(14,26,10)
890.      DATA CAPCST/3640*2./
891.      COMMON /CENSUS/BUILDS(11,14,2,3),OPERGS(11,14,3),FUEL(11,3)
892.      X ,NUCFL(11,3),OANDM(11,3)
893.      REAL*4 NUCFL
894.      NUMPLT=9

```

```

895.      NUMYR=26
896.      NUMREG=10
897.      IF(IYR.LE.12) GO TO 1000
898.      NYR=IYR-12
899.      QTY=(AMT(IPLT,12,IREG)-AMT(IPLT,11,IREG))*(RATE1(IPLT,IREG)**NYR)
900.      RETURN
901. 1000 CONTINUE
902.      IF(IYR.NE.1) GO TO 600
903.      QTY=AMT(IPLT,IYR,IREG)
904.      GO TO 500
905. 600 QTY=AMT(IPLT,IYR,IREG)-AMT(IPLT,IYR-1,IREG)
906. C      WRITE(KFILE,105) IYR1,IPLT,QTY,COST
907. 500 CONTINUE
908. 105 FORMAT(2I5,2F20.3)
909. C      DO 250 IREG=1,NUMREG
910. C      DO 250 IYR=1,NUMYR
911. C 250 SUM(IYR,IREG)=0.0
912. C      DO 300 IREG=1,NUMREG
913. C      DO 300 IPLT=1,NUMPLT
914. C      DO 300 IYR=1,26
915. C 300 SUM(IYR,IREG)=AMT(IPLT,IYR,IREG)+SUM(IYR,IREG)
916. C      DO 400 IREG=1,NUMREG
917. C      DO 400 IYR=1,NUMYR
918. C      DO 400 IYR=1,26
919. C      IF(SUM(IYR,IREG).GT.0.) GO TO 425
920. C      FRACT(IYR,IREG)=0.0
921. C      GO TO 400
922. C 425 FRACT(IYR,IREG)=AMT(IPLT,IYR,IREG)/SUM(IYR,IREG)
923. C 400 CONTINUE
924.      RETURN
925.      END
926.      SUBROUTINE DEBUG(BUILDS,AMT)
927.      REAL*4 AMT(14,26,10),BUILDS(11,14,2,3),TOT(14,10)
928.      DO 1000 IRUN=1,2
929.      IYR=6
930.      IF(IRUN.NE.1) GO TO 75
931.      GO TO 80
932. 75 IYR=11
933. 80 DO 50 IREG=1,10
934.      DO 50 IPLT=1,14
935. 50 TOT(IPLT,IREG)=0.0
936.      DO 100 IREG=1,10
937.      DO 100 IPLT=1,12
938. C      DO 200 I=1,IRUN
939.      TOT(IPLT,IREG)=BUILDS(IREG,IPLT,1,IRUN)
940. 200 CONTINUE
941.      IF(ABS(TOT(IPLT,IREG)-AMT(IPLT,IYR,IREG)/1000.).LT..01) GO TO 100
942.      WRITE(60,1) IRUN,IREG,IPLT
943.      X=AMT(IPLT,IYR,IREG)/1000.
944.      WRITE(60,2) TOT(IPLT,IREG),X
945. 100 CONTINUE
946. 1000 CONTINUE
947. 1 FORMAT('THE FOLLOWING REGIONS HAVE DISCREPANCIES BETWEEN CREMOD
948. + AND NUFFS',/, 'IRUN=',I3,5X, 'IREG=',I3,5X, 'IPLT=',I3)
949. 2 FORMAT('NUFS TOTAL=',F10.5,4X, 'CREMOD TOTAL=',F10.5)
950.      RETURN

```

```

951.      END
952.      SUBROUTINE RDCREM
953.      COMMON /OPS/ AMT(14,26,10),RATE1(14,10)
954.      REAL*4 CAPCST(14,26,10)
955.      REAL*4 AMT1(14,101,10)
956.      DATA CAPCST/3640*2./
957.      COMMON /CENSUS/BUILDS(11,14,2,3),OPER8S(11,14,3),FUEL(11,3)
958.      X ,NUCFL(11,3),OANDM(11,3)
959.      REAL*4 NUCFL
960.      NUMPLT=12
961.      NUMYR=26
962.      NUMREG=10
963.      DO 300 I=1,NUMREG
964.      DO 300 J=1,NUMPLT
965.      DO 300 K=1,NUMYR
966.      300 AMT(J,K,I)=0.0
967.      800 READ(??,1,CND=700) IYR1,IPLT,(AMT1(IPLT,IYR1,IREG),IREG=1,NUMREG)
968.      IYR=IYR1-78
969.      DO 100 IREG=1,NUMREG
970.      100 AMT(IPLT,IYR,IREG)=AMT1(IPLT,IYR1,IREG)
971.      GO TO 800
972.      1 FORMAT(I2,1X,I2,3X,10F7.4)
973.      700 DO 200 IPLT=1,NUMPLT
974.      DO 200 IYR=1,12
975.      DO 200 IREG=1,NUMREG
976.      200 AMT(IPLT,IYR,IREG)=AMT(IPLT,IYR,IREG)*1000.
977.      DO 600 IREG=1,NUMREG
978.      DO 600 IPLT=1,NUMPLT
979.      IF(AMT(IPLT,12,IREG).GT.0.) GO TO 625
980.      RATE1(IPLT,IREG)=0.
981.      GO TO 600
982.      625 RATE1(IPLT,IREG)=(BUILDS(IREG,IPLT,1,3)*1000./AMT(IPLT,12,IREG))
983.      + **.25
984.      600 CONTINUE
985.      RETURN
986.      END
987.      //LKED.SYSLIB DD
988.      //LKED.SYSLMOD DD DISP=(NEW,CATLG),
989.      //      DSN=CN6376.IS3.EYNON.GO.FINANCE(LARGE),
990.      //      SPACE=(TRK,(10,1,1)),UNIT=3350,
991.      //      DCB=(DSORG=PO,RECFM=U,BLKSIZE=7294)

```

ii) LISTING OF NUF.S.MEFS.BASEDATA

1.	0101TABLE DEFL79	CONVERTS 75 DOLLARS TO REPORT YEAR(79) DOLLARS				
2.	*	DATA				
3.	1	1.295				
4.	0010CLASS NAMDOE	LIST OF REGIONS				
5.	U1 U2	U3	U4	U5	U6	U7 U8 U9 UA
6.	0012CLASS FUENAM	LIST OF FUELS WITH SHADOW PRICES				
7.	RS CA	CJ	CL	DG	DS	CK CP CF CT
8.	UF E1					
9.	0011CLASS ZUENAM	LINKED LIST OF FUEL QUANTITY NAMES				
10.	ZR ZC	ZJ	ZL	ZG	ZD	ZK ZP ZF ZT
11.	ZN **					
12.	0004CLASS EXLOAD	LOAD CODES USED FOR EXISTING PLANTS				
13.	7 8	9	6			
14.	0004CLASS NWLOAD	LOAD CODES USED FOR NEW PLANTS				
15.	1 2	3	5			
16.	0002CLASS TRANS	CODES USED FOR TRANSMISSION COSTS (ACTIVITIES)				
17.	EN EO					
18.	0009CLASS MISC GEN	CODES FOR MISC. GENERATION ACTIVITIES (GEO.SYN)				
19.	GD GJ	GO	GS	GM	GF	G4 LD LF
20.	0209TABLE MISCBLD	CODES FOR MISC. BUILD ACTIVITIES (AND KODE&DCF)				
21.	*	GE	GK	GP	G2	GN
22.	KODE	12.	12.	12.	12.	12.
23.	DCF	0.865	0.865	0.865	0.865	0.865
24.	*	GG	G1	LC	LE	
25.	KODE	12.	12.	12.	12.	
26.	DCF	0.865	0.865	0.865	0.865	
27.	2401TABLE BLDLOGIC					
28.	*	KODE	BUILD CODES FOR OUTPUT			
29.	1	2.	OIL FIRED STEAM			
30.	2	3.	BITUMINOUS/HI SULFUR/WITH SCR			
31.	3	3.	BITUMINOUS/MED SULFUR/WITH SCR			
32.	6	9.	DISTILLATE TURBINE			
33.	5	9.	NATURAL GAS TURBINE			
34.	7	8.	COMBINED CYCLE DISTILLATE FIRED			
35.	A	8.	COMBINED CYCLE GAS FIRED			
36.	8	1.	NATURAL GAS FIRED STEAM			
37.	4	3.	BITUMINOUS/LOW SULFUR/WITH SCR			
38.	J	4.	BITUMINOUS/HIGH SULFUR/NO SCR			
39.	Q	4.	BITUMINOUS/MED SULFUR/NO SCR			
40.	X	4.	BITUMINOUS/LOW SULFUR/NO SCR			
41.	P	3.	SUB-BITUMINOUS/MED SUL/WITH SCR			
42.	E	3.	SUB-BITUMINOUS/LOW SUL/WITH SCR			
43.	M	4.	SUB-BITUMINOUS/MED SUL/NO SCR			
44.	O	4.	SUB-BITUMINOUS/LOW SUL/NO SCR			
45.	V	3.	LIGNITE/MEDIUM SULFUR/WITH SCR			
46.	Z	3.	LIGNITE/LOW SULFUR/WITH SCR			
47.	T	4.	LIGNITE/MEDIUM SULFUR/NO SCR			
48.	Y	4.	LIGNITE/LOW SULFUR/NO SCR			
49.	9	6.	HYDRO PONDAGE			
50.	H	6.	HYDRO PONDAGE			
51.	K	5.	HYDRO PUMP STORAGE			
52.	N	7.	NUCLEAR			
53.	0203TABLE BLD FAC					
54.	*	0	Z	Y		
55.	ROW2	1.0	0.9	1.0		

56.	NUC	1.0	1.0	1.0		
57.	2210TABLE CPCSY85 , EMPTY, R=	50, C=	50			
58.	*	U1	U2	U3	U4	U5
59.	1	453.0000	482.0000	402.0000	417.0000	462.0000
60.	5	126.0000	126.0000	126.0000	126.0000	126.0000
61.	6	136.0000	136.0000	136.0000	136.0000	136.0000
62.	A	330.0000	330.0000	298.0000	298.0000	298.0000
63.	7	342.0000	342.0000	311.0000	311.0000	311.0000
64.	8	453.0000	482.0000	402.0000	417.0000	462.0000
65.	9	426.0000	393.0000	465.0000	348.0000	787.0000
66.	K	399.0000	181.0000	210.0000	218.0000	179.0000
67.	2	729.0000	777.0000	647.0000	671.0000	744.0000
68.	3	658.0000	701.0000	583.0000	605.0000	671.0000
69.	4	607.0000	646.0000	538.0000	558.0000	619.0000
70.	J	607.0000	646.0000	538.0000	558.0000	619.0000
71.	Q	607.0000	646.0000	538.0000	558.0000	619.0000
72.	X	607.0000	646.0000	538.0000	558.0000	619.0000
73.	M	666.0000	710.0000	591.0000	613.0000	680.0000
74.	O	666.0000	710.0000	591.0000	613.0000	680.0000
75.	P	723.0000	771.0000	642.0000	665.0000	738.0000
76.	E	666.0000	710.0000	591.0000	613.0000	680.0000
77.	T	664.0000	708.0000	590.0000	611.0000	678.0000
78.	Y	664.0000	708.0000	590.0000	611.0000	678.0000
79.	V	756.0000	805.0000	670.0000	695.0000	771.0000
80.	Z	664.0000	708.0000	590.0000	611.0000	678.0000
81.	*	U6	U7	U8	U9	UA
82.	1	408.0000	447.0000	430.0000	473.0000	456.0000
83.	5	126.0000	126.0000	126.0000	126.0000	126.0000
84.	6	136.0000	136.0000	136.0000	136.0000	136.0000
85.	A	282.0000	298.0000	298.0000	293.0000	293.0000
86.	7	295.0000	311.0000	311.0000	306.0000	306.0000
87.	8	408.0000	447.0000	430.0000	473.0000	456.0000
88.	9	537.0000	948.0000	636.0000	242.0000	257.0000
89.	K	237.0000	196.0000	177.0000	271.0000	0.
90.	2	657.0000	720.0000	693.0000	762.0000	734.0000
91.	3	592.0000	649.0000	625.0000	687.0000	662.0000
92.	4	546.0000	599.0000	576.0000	634.0000	611.0000
93.	J	546.0000	599.0000	576.0000	634.0000	611.0000
94.	Q	546.0000	599.0000	576.0000	634.0000	611.0000
95.	X	546.0000	599.0000	576.0000	634.0000	611.0000
96.	M	600.0000	658.0000	633.0000	697.0000	671.0000
97.	O	600.0000	658.0000	633.0000	697.0000	671.0000
98.	P	651.0000	714.0000	687.0000	756.0000	728.0000
99.	E	600.0000	658.0000	633.0000	697.0000	671.0000
100.	T	598.0000	656.0000	631.0000	695.0000	669.0000
101.	Y	598.0000	656.0000	631.0000	695.0000	669.0000
102.	V	680.0000	746.0000	718.0000	790.0000	761.0000
103.	Z	598.0000	656.0000	631.0000	695.0000	669.0000
104.	2210TABLE CPCSY90 , EMPTY, R=	50, C=	50			
105.	*	U1	U2	U3	U4	U5
106.	1	453.0000	482.0000	402.0000	417.0000	462.0000
107.	5	126.0000	126.0000	126.0000	126.0000	126.0000
108.	6	136.0000	136.0000	136.0000	136.0000	136.0000
109.	A	330.0000	330.0000	298.0000	298.0000	298.0000
110.	7	342.0000	342.0000	311.0000	311.0000	311.0000
111.	8	453.0000	482.0000	402.0000	417.0000	462.0000

112.	9	426.0000	393.0000	465.0000	348.0000	787.0000
113.	K	399.0000	181.0000	210.0000	218.0000	179.0000
114.	2	729.0000	777.0000	647.0000	671.0000	744.0000
115.	3	658.0000	701.0000	583.0000	605.0000	671.0000
116.	4	607.0000	646.0000	538.0000	558.0000	619.0000
117.	J	607.0000	646.0000	538.0000	558.0000	619.0000
118.	Q	607.0000	646.0000	538.0000	558.0000	619.0000
119.	X	607.0000	646.0000	538.0000	558.0000	619.0000
120.	M	666.0000	710.0000	591.0000	613.0000	680.0000
121.	O	666.0000	710.0000	591.0000	613.0000	680.0000
122.	P	723.0000	771.0000	642.0000	665.0000	738.0000
123.	E	666.0000	710.0000	591.0000	613.0000	680.0000
124.	T	664.0000	708.0000	590.0000	611.0000	678.0000
125.	Y	664.0000	708.0000	590.0000	611.0000	678.0000
126.	V	756.0000	805.0000	670.0000	695.0000	771.0000
127.	Z	664.0000	708.0000	590.0000	611.0000	678.0000
128.	*	U6	U7	U8	U9	UA
129.	1	408.0000	447.0000	430.0000	473.0000	456.0000
130.	5	126.0000	126.0000	126.0000	126.0000	126.0000
131.	6	136.0000	136.0000	136.0000	136.0000	136.0000
132.	A	282.0000	298.0000	298.0000	293.0000	293.0000
133.	7	295.0000	311.0000	311.0000	306.0000	306.0000
134.	8	408.0000	447.0000	430.0000	473.0000	456.0000
135.	9	537.0000	948.0000	636.0000	242.0000	257.0000
136.	K	237.0000	196.0000	177.0000	271.0000	0.
137.	2	657.0000	720.0000	693.0000	762.0000	734.0000
138.	3	592.0000	649.0000	625.0000	687.0000	662.0000
139.	4	546.0000	599.0000	576.0000	634.0000	611.0000
140.	J	546.0000	599.0000	576.0000	634.0000	611.0000
141.	Q	546.0000	599.0000	576.0000	634.0000	611.0000
142.	X	546.0000	599.0000	576.0000	634.0000	611.0000
143.	M	600.0000	658.0000	633.0000	697.0000	671.0000
144.	O	600.0000	658.0000	633.0000	697.0000	671.0000
145.	P	651.0000	714.0000	687.0000	756.0000	728.0000
146.	E	600.0000	658.0000	633.0000	697.0000	671.0000
147.	T	598.0000	656.0000	631.0000	695.0000	669.0000
148.	Y	598.0000	656.0000	631.0000	695.0000	669.0000
149.	V	680.0000	746.0000	718.0000	790.0000	761.0000
150.	Z	598.0000	656.0000	631.0000	695.0000	669.0000
151.	2210TABLE CPCSY95 , EMPTY, R= 50, C= 50					
152.	*	U1	U2	U3	U4	U5
153.	1	453.0000	402.0000	402.0000	417.0000	462.0000
154.	5	126.0000	126.0000	126.0000	126.0000	126.0000
155.	6	136.0000	136.0000	136.0000	136.0000	136.0000
156.	0	330.0000	330.0000	298.0000	298.0000	298.0000
157.	7	342.0000	342.0000	311.0000	311.0000	311.0000
158.	8	453.0000	482.0000	402.0000	417.0000	462.0000
159.	9	426.0000	393.0000	465.0000	348.0000	787.0000
160.	K	399.0000	181.0000	210.0000	218.0000	179.0000
161.	2	729.0000	777.0000	647.0000	671.0000	744.0000
162.	3	658.0000	701.0000	583.0000	605.0000	671.0000
163.	4	607.0000	646.0000	538.0000	558.0000	619.0000
164.	J	607.0000	646.0000	538.0000	558.0000	619.0000
165.	Q	607.0000	646.0000	538.0000	558.0000	619.0000
166.	X	607.0000	646.0000	538.0000	558.0000	619.0000
167.	M	666.0000	710.0000	591.0000	613.0000	680.0000

168.	O	666.0000	710.0000	591.0000	613.0000	680.0000
169.	P	723.0000	771.0000	642.0000	665.0000	738.0000
170.	E	666.0000	710.0000	591.0000	613.0000	680.0000
171.	T	664.0000	708.0000	590.0000	611.0000	678.0000
172.	Y	664.0000	708.0000	590.0000	611.0000	678.0000
173.	V	756.0000	805.0000	670.0000	695.0000	771.0000
174.	Z	664.0000	708.0000	590.0000	611.0000	678.0000
175.	*	U6	U7	U8	U9	UA
176.	1	408.0000	447.0000	430.0000	473.0000	456.0000
177.	5	126.0000	126.0000	126.0000	126.0000	126.0000
178.	6	136.0000	136.0000	136.0000	136.0000	136.0000
179.	A	282.0000	298.0000	298.0000	293.0000	293.0000
180.	7	295.0000	311.0000	311.0000	306.0000	306.0000
181.	8	408.0000	447.0000	430.0000	473.0000	456.0000
182.	9	537.0000	948.0000	636.0000	242.0000	257.0000
183.	K	237.0000	196.0000	177.0000	271.0000	0.
184.	2	657.0000	720.0000	693.0000	762.0000	734.0000
185.	3	592.0000	649.0000	625.0000	687.0000	662.0000
186.	4	546.0000	599.0000	576.0000	634.0000	611.0000
187.	J	546.0000	599.0000	576.0000	634.0000	611.0000
188.	Q	546.0000	599.0000	576.0000	634.0000	611.0000
189.	X	546.0000	599.0000	576.0000	634.0000	611.0000
190.	M	600.0000	658.0000	633.0000	697.0000	671.0000
191.	O	600.0000	658.0000	633.0000	697.0000	671.0000
192.	P	651.0000	714.0000	687.0000	756.0000	728.0000
193.	E	600.0000	658.0000	633.0000	697.0000	671.0000
194.	T	598.0000	656.0000	631.0000	695.0000	669.0000
195.	Y	598.0000	656.0000	631.0000	695.0000	669.0000
196.	V	680.0000	746.0000	718.0000	790.0000	761.0000
197.	Z	598.0000	656.0000	631.0000	695.0000	669.0000
198.	2210TABLE CPCSZ85 , EMPTY, R= 50, C= 50					
199.	*	U1	U2	U3	U4	U5
200.	1	453.0000	482.0000	402.0000	417.0000	462.0000
201.	5	126.0000	126.0000	126.0000	126.0000	126.0000
202.	6	136.0000	136.0000	136.0000	136.0000	136.0000
203.	A	330.0000	330.0000	298.0000	298.0000	298.0000
204.	7	342.0000	342.0000	311.0000	311.0000	311.0000
205.	8	453.0000	482.0000	402.0000	417.0000	462.0000
206.	9	426.0000	393.0000	465.0000	348.0000	787.0000
207.	K	399.0000	181.0000	210.0000	218.0000	179.0000
208.	2	729.0000	777.0000	647.0000	671.0000	744.0000
209.	3	658.0000	701.0000	583.0000	605.0000	671.0000
210.	4	607.0000	646.0000	538.0000	558.0000	619.0000
211.	J	607.0000	646.0000	538.0000	558.0000	619.0000
212.	Q	607.0000	646.0000	538.0000	558.0000	619.0000
213.	X	607.0000	646.0000	538.0000	558.0000	619.0000
214.	M	666.0000	710.0000	591.0000	613.0000	680.0000
215.	O	666.0000	710.0000	591.0000	613.0000	680.0000
216.	P	723.0000	771.0000	642.0000	665.0000	738.0000
217.	E	666.0000	710.0000	591.0000	613.0000	680.0000
218.	T	664.0000	708.0000	590.0000	611.0000	678.0000
219.	Y	664.0000	708.0000	590.0000	611.0000	678.0000
220.	V	756.0000	805.0000	670.0000	695.0000	771.0000
221.	Z	664.0000	708.0000	590.0000	611.0000	678.0000
222.	*	U6	U7	U8	U9	UA
223.	1	408.0000	447.0000	430.0000	473.0000	456.0000

224.	5	126.0000	126.0000	126.0000	126.0000	126.0000
225.	6	136.0000	136.0000	136.0000	136.0000	136.0000
226.	A	282.0000	298.0000	298.0000	293.0000	293.0000
227.	7	295.0000	311.0000	311.0000	306.0000	306.0000
228.	8	408.0000	447.0000	430.0000	473.0000	456.0000
229.	9	537.0000	948.0000	636.0000	242.0000	257.0000
230.	K	237.0000	196.0000	177.0000	271.0000	0.
231.	2	657.0000	720.0000	693.0000	762.0000	734.0000
232.	3	592.0000	649.0000	625.0000	687.0000	662.0000
233.	4	546.0000	599.0000	576.0000	634.0000	611.0000
234.	J	546.0000	599.0000	576.0000	634.0000	611.0000
235.	Q	546.0000	599.0000	576.0000	634.0000	611.0000
236.	X	546.0000	599.0000	576.0000	634.0000	611.0000
237.	M	600.0000	658.0000	633.0000	697.0000	671.0000
238.	O	600.0000	658.0000	633.0000	697.0000	671.0000
239.	P	651.0000	714.0000	687.0000	756.0000	728.0000
240.	E	600.0000	658.0000	633.0000	697.0000	671.0000
241.	T	598.0000	656.0000	631.0000	695.0000	669.0000
242.	Y	598.0000	656.0000	631.0000	695.0000	669.0000
243.	V	680.0000	746.0000	718.0000	790.0000	761.0000
244.	Z	598.0000	656.0000	631.0000	695.0000	669.0000
245.	2210TABLE CPCSZ90 , EMPTY, R= 50, C= 50					
246.	*	U1	U2	U3	U4	U5
247.	1	453.0000	482.0000	402.0000	417.0000	462.0000
248.	5	126.0000	126.0000	126.0000	126.0000	126.0000
249.	6	136.0000	136.0000	136.0000	136.0000	136.0000
250.	A	330.0000	330.0000	298.0000	298.0000	298.0000
251.	7	342.0000	342.0000	311.0000	311.0000	311.0000
252.	8	453.0000	482.0000	402.0000	417.0000	462.0000
253.	9	426.0000	393.0000	465.0000	348.0000	787.0000
254.	K	399.0000	181.0000	210.0000	218.0000	179.0000
255.	2	729.0000	777.0000	647.0000	671.0000	744.0000
256.	3	658.0000	701.0000	583.0000	605.0000	671.0000
257.	4	607.0000	646.0000	538.0000	558.0000	619.0000
258.	J	607.0000	646.0000	538.0000	558.0000	619.0000
259.	Q	607.0000	646.0000	538.0000	558.0000	619.0000
260.	X	607.0000	646.0000	538.0000	558.0000	619.0000
261.	M	666.0000	710.0000	591.0000	613.0000	680.0000
262.	O	666.0000	710.0000	591.0000	613.0000	680.0000
263.	P	723.0000	771.0000	642.0000	665.0000	738.0000
264.	E	666.0000	710.0000	591.0000	613.0000	680.0000
265.	T	664.0000	708.0000	590.0000	611.0000	678.0000
266.	Y	664.0000	708.0000	590.0000	611.0000	678.0000
267.	V	756.0000	805.0000	670.0000	695.0000	771.0000
268.	Z	664.0000	708.0000	590.0000	611.0000	678.0000
269.	*	U6	U7	U8	U9	UA
270.	1	408.0000	447.0000	430.0000	473.0000	456.0000
271.	5	126.0000	126.0000	126.0000	126.0000	126.0000
272.	6	136.0000	136.0000	136.0000	136.0000	136.0000
273.	A	282.0000	298.0000	298.0000	293.0000	293.0000
274.	7	295.0000	311.0000	311.0000	306.0000	306.0000
275.	8	408.0000	447.0000	430.0000	473.0000	456.0000
276.	9	537.0000	948.0000	636.0000	242.0000	257.0000
277.	K	237.0000	196.0000	177.0000	271.0000	0.
278.	2	657.0000	720.0000	693.0000	762.0000	734.0000
279.	3	592.0000	649.0000	625.0000	687.0000	662.0000

280.	4	546.0000	599.0000	576.0000	634.0000	611.0000
281.	J	546.0000	599.0000	576.0000	634.0000	611.0000
282.	Q	546.0000	599.0000	576.0000	634.0000	611.0000
283.	X	546.0000	599.0000	576.0000	634.0000	611.0000
284.	M	600.0000	658.0000	633.0000	697.0000	671.0000
285.	O	600.0000	658.0000	633.0000	697.0000	671.0000
286.	P	651.0000	714.0000	687.0000	756.0000	728.0000
287.	E	600.0000	658.0000	633.0000	697.0000	671.0000
288.	T	598.0000	656.0000	631.0000	695.0000	669.0000
289.	Y	598.0000	656.0000	631.0000	695.0000	669.0000
290.	V	680.0000	746.0000	718.0000	790.0000	761.0000
291.	Z	598.0000	656.0000	631.0000	695.0000	669.0000
292.	2210TABLE CPCSZ95 , EMPTY, R= 50, C= 50					
293.	*	U1	U2	U3	U4	U5
294.	1	453.0000	482.0000	402.0000	417.0000	462.0000
295.	5	126.0000	126.0000	126.0000	126.0000	126.0000
296.	6	136.0000	136.0000	136.0000	136.0000	136.0000
297.	A	330.0000	330.0000	298.0000	298.0000	298.0000
298.	7	342.0000	342.0000	311.0000	311.0000	311.0000
299.	8	453.0000	482.0000	402.0000	417.0000	462.0000
300.	9	426.0000	393.0000	465.0000	348.0000	787.0000
301.	K	399.0000	181.0000	210.0000	218.0000	179.0000
302.	2	729.0000	777.0000	647.0000	671.0000	744.0000
303.	3	658.0000	701.0000	583.0000	605.0000	671.0000
304.	4	607.0000	646.0000	538.0000	558.0000	619.0000
305.	J	607.0000	646.0000	538.0000	558.0000	619.0000
306.	Q	607.0000	646.0000	538.0000	558.0000	619.0000
307.	X	607.0000	646.0000	538.0000	558.0000	619.0000
308.	M	666.0000	710.0000	591.0000	613.0000	680.0000
309.	O	666.0000	710.0000	591.0000	613.0000	680.0000
310.	P	723.0000	771.0000	642.0000	665.0000	738.0000
311.	E	666.0000	710.0000	591.0000	613.0000	680.0000
312.	T	664.0000	708.0000	590.0000	611.0000	678.0000
313.	Y	664.0000	708.0000	590.0000	611.0000	678.0000
314.	V	756.0000	805.0000	670.0000	695.0000	771.0000
315.	Z	664.0000	708.0000	590.0000	611.0000	678.0000
316.	*	U6	U7	U8	U9	UA
317.	1	408.0000	447.0000	430.0000	473.0000	456.0000
318.	5	126.0000	126.0000	126.0000	126.0000	126.0000
319.	6	136.0000	136.0000	136.0000	136.0000	136.0000
320.	A	282.0000	298.0000	298.0000	293.0000	293.0000
321.	7	295.0000	311.0000	311.0000	306.0000	306.0000
322.	8	408.0000	447.0000	430.0000	473.0000	456.0000
323.	9	537.0000	948.0000	636.0000	242.0000	257.0000
324.	K	237.0000	196.0000	177.0000	271.0000	0.
325.	2	657.0000	720.0000	693.0000	762.0000	734.0000
326.	3	592.0000	649.0000	625.0000	687.0000	662.0000
327.	4	546.0000	599.0000	576.0000	634.0000	611.0000
328.	J	546.0000	599.0000	576.0000	634.0000	611.0000
329.	Q	546.0000	599.0000	576.0000	634.0000	611.0000
330.	X	546.0000	599.0000	576.0000	634.0000	611.0000
331.	M	600.0000	658.0000	633.0000	697.0000	671.0000
332.	O	600.0000	658.0000	633.0000	697.0000	671.0000
333.	P	651.0000	714.0000	687.0000	756.0000	728.0000
334.	E	600.0000	658.0000	633.0000	697.0000	671.0000
335.	T	598.0000	656.0000	631.0000	695.0000	669.0000

336.	Y	598.0000	656.0000	631.0000	695.0000	669.0000
337.	V	680.0000	746.0000	718.0000	790.0000	761.0000
338.	Z	598.0000	656.0000	631.0000	695.0000	669.0000
339.	2210TABLE CPCS085 , EMPTY, R= 50, C= 50					
340.	*	U1	U2	U3	U4	U5
341.	1	453.0000	482.0000	402.0000	417.0000	462.0000
342.	5	126.0000	126.0000	126.0000	126.0000	126.0000
343.	6	136.0000	136.0000	136.0000	136.0000	136.0000
344.	A	330.0000	330.0000	298.0000	298.0000	298.0000
345.	7	342.0000	342.0000	311.0000	311.0000	311.0000
346.	8	453.0000	482.0000	402.0000	417.0000	462.0000
347.	9	426.0000	393.0000	465.0000	348.0000	787.0000
348.	K	399.0000	181.0000	210.0000	218.0000	179.0000
349.	2	754.0000	803.0000	669.0000	693.0000	769.0000
350.	3	706.0000	752.0000	626.0000	649.0000	720.0000
351.	4	702.0000	748.0000	623.0000	646.0000	716.0000
352.	J	607.0000	646.0000	538.0000	558.0000	619.0000
353.	Q	607.0000	646.0000	538.0000	558.0000	619.0000
354.	X	607.0000	646.0000	538.0000	558.0000	619.0000
355.	M	666.0000	710.0000	591.0000	613.0000	680.0000
356.	O	666.0000	710.0000	591.0000	613.0000	680.0000
357.	P	770.0000	821.0000	683.0000	709.0000	786.0000
358.	E	762.0000	813.0000	677.0000	702.0000	778.0000
359.	T	664.0000	708.0000	590.0000	611.0000	678.0000
360.	Y	664.0000	708.0000	590.0000	611.0000	678.0000
361.	V	790.0000	842.0000	701.0000	727.0000	806.0000
362.	Z	760.0000	810.0000	675.0000	700.0000	776.0000
363.	*	U6	U7	U8	U9	UA
364.	1	408.0000	447.0000	430.0000	473.0000	456.0000
365.	5	126.0000	126.0000	126.0000	126.0000	126.0000
366.	6	136.0000	136.0000	136.0000	136.0000	136.0000
367.	A	282.0000	298.0000	298.0000	293.0000	293.0000
368.	7	295.0000	311.0000	311.0000	306.0000	306.0000
369.	8	408.0000	447.0000	430.0000	473.0000	456.0000
370.	9	537.0000	948.0000	636.0000	242.0000	257.0000
371.	K	237.0000	196.0000	177.0000	271.0000	0.
372.	2	679.0000	744.0000	716.0000	788.0000	759.0000
373.	3	635.0000	697.0000	671.0000	738.0000	711.0000
374.	4	632.0000	693.0000	667.0000	733.0000	707.0000
375.	J	546.0000	599.0000	576.0000	634.0000	611.0000
376.	Q	546.0000	599.0000	576.0000	634.0000	611.0000
377.	X	546.0000	599.0000	576.0000	634.0000	611.0000
378.	M	600.0000	658.0000	633.0000	697.0000	671.0000
379.	O	600.0000	658.0000	633.0000	697.0000	671.0000
380.	P	694.0000	761.0000	732.0000	805.0000	776.0000
381.	E	687.0000	753.0000	725.0000	797.0000	768.0000
382.	T	598.0000	656.0000	631.0000	695.0000	669.0000
383.	Y	598.0000	656.0000	631.0000	695.0000	669.0000
384.	V	711.0000	780.0000	751.0000	826.0000	795.0000
385.	Z	685.0000	751.0000	723.0000	795.0000	766.0000
386.	2210TABLE CPCS090 , EMPTY, R= 50, C= 50					
387.	*	U1	U2	U3	U4	U5
388.	1	453.0000	482.0000	402.0000	417.0000	462.0000
389.	5	126.0000	126.0000	126.0000	126.0000	126.0000
390.	6	136.0000	136.0000	136.0000	136.0000	136.0000
391.	A	330.0000	330.0000	298.0000	298.0000	298.0000

392.	7	342.0000	342.0000	311.0000	311.0000	311.0000
393.	8	453.0000	482.0000	402.0000	417.0000	462.0000
394.	9	426.0000	393.0000	465.0000	348.0000	787.0000
395.	K	399.0000	181.0000	210.0000	218.0000	179.0000
396.	2	754.0000	803.0000	669.0000	693.0000	769.0000
397.	3	706.0000	752.0000	626.0000	649.0000	720.0000
398.	4	702.0000	748.0000	623.0000	646.0000	716.0000
399.	J	607.0000	646.0000	538.0000	558.0000	619.0000
400.	Q	607.0000	646.0000	538.0000	558.0000	619.0000
401.	X	607.0000	646.0000	538.0000	558.0000	619.0000
402.	M	666.0000	710.0000	591.0000	613.0000	680.0000
403.	O	666.0000	710.0000	591.0000	613.0000	680.0000
404.	P	770.0000	821.0000	683.0000	709.0000	786.0000
405.	E	762.0000	813.0000	677.0000	702.0000	778.0000
406.	T	664.0000	708.0000	590.0000	611.0000	678.0000
407.	Y	664.0000	708.0000	590.0000	611.0000	678.0000
408.	V	790.0000	842.0000	701.0000	727.0000	806.0000
409.	Z	760.0000	810.0000	675.0000	700.0000	776.0000
410.	*	U6	U7	U8	U9	UA
411.	1	408.0000	447.0000	430.0000	473.0000	456.0000
412.	5	126.0000	126.0000	126.0000	126.0000	126.0000
413.	6	136.0000	136.0000	136.0000	136.0000	136.0000
414.	A	282.0000	298.0000	298.0000	293.0000	293.0000
415.	7	295.0000	311.0000	311.0000	306.0000	306.0000
416.	8	408.0000	447.0000	430.0000	473.0000	456.0000
417.	9	537.0000	948.0000	636.0000	242.0000	257.0000
418.	K	237.0000	196.0000	177.0000	271.0000	0.
419.	2	679.0000	744.0000	716.0000	788.0000	759.0000
420.	3	635.0000	697.0000	671.0000	738.0000	711.0000
421.	4	632.0000	693.0000	667.0000	733.0000	707.0000
422.	J	546.0000	599.0000	576.0000	634.0000	611.0000
423.	Q	546.0000	599.0000	576.0000	634.0000	611.0000
424.	X	546.0000	599.0000	576.0000	634.0000	611.0000
425.	M	600.0000	658.0000	633.0000	697.0000	671.0000
426.	O	600.0000	658.0000	633.0000	697.0000	671.0000
427.	P	694.0000	761.0000	732.0000	805.0000	776.0000
428.	E	687.0000	753.0000	725.0000	797.0000	768.0000
429.	T	598.0000	656.0000	631.0000	695.0000	669.0000
430.	Y	598.0000	656.0000	631.0000	695.0000	669.0000
431.	V	711.0000	780.0000	751.0000	826.0000	795.0000
432.	Z	685.0000	751.0000	723.0000	795.0000	766.0000
433.	2210TABLE CPCS095 , EMPTY, R= 50, C= 50					
434.	*	U1	U2	U3	U4	U5
435.	1	453.0000	482.0000	402.0000	417.0000	462.0000
436.	5	126.0000	126.0000	126.0000	126.0000	126.0000
437.	6	136.0000	136.0000	136.0000	136.0000	136.0000
438.	A	330.0000	330.0000	298.0000	298.0000	298.0000
439.	7	342.0000	342.0000	311.0000	311.0000	311.0000
440.	8	453.0000	482.0000	402.0000	417.0000	462.0000
441.	9	426.0000	393.0000	465.0000	348.0000	787.0000
442.	K	399.0000	181.0000	210.0000	218.0000	179.0000
443.	2	754.0000	803.0000	669.0000	693.0000	769.0000
444.	3	706.0000	752.0000	626.0000	649.0000	720.0000
445.	4	702.0000	748.0000	623.0000	646.0000	716.0000
446.	J	607.0000	646.0000	538.0000	558.0000	619.0000
447.	Q	607.0000	646.0000	538.0000	558.0000	619.0000

448.	X	607.0000	646.0000	538.0000	558.0000	619.0000
449.	M	666.0000	710.0000	591.0000	613.0000	680.0000
450.	O	666.0000	710.0000	591.0000	613.0000	680.0000
451.	P	770.0000	821.0000	683.0000	709.0000	786.0000
452.	E	762.0000	813.0000	677.0000	702.0000	778.0000
453.	T	664.0000	708.0000	590.0000	611.0000	678.0000
454.	Y	664.0000	708.0000	590.0000	611.0000	678.0000
455.	V	790.0000	842.0000	701.0000	727.0000	806.0000
456.	Z	760.0000	810.0000	675.0000	700.0000	776.0000
457.	*	U6	U7	U8	U9	UA
458.	1	408.0000	447.0000	430.0000	473.0000	456.0000
459.	5	126.0000	126.0000	126.0000	126.0000	126.0000
460.	6	136.0000	136.0000	136.0000	136.0000	136.0000
461.	A	282.0000	298.0000	298.0000	293.0000	293.0000
462.	7	295.0000	311.0000	311.0000	306.0000	306.0000
463.	8	408.0000	447.0000	430.0000	473.0000	456.0000
464.	9	537.0000	948.0000	636.0000	242.0000	257.0000
465.	K	237.0000	196.0000	177.0000	271.0000	0.
466.	2	679.0000	744.0000	716.0000	788.0000	759.0000
467.	3	635.0000	697.0000	671.0000	738.0000	711.0000
468.	4	632.0000	693.0000	667.0000	733.0000	707.0000
469.	J	546.0000	599.0000	576.0000	634.0000	611.0000
470.	Q	546.0000	599.0000	576.0000	634.0000	611.0000
471.	X	546.0000	599.0000	576.0000	634.0000	611.0000
472.	M	600.0000	658.0000	633.0000	697.0000	671.0000
473.	O	600.0000	658.0000	633.0000	697.0000	671.0000
474.	P	694.0000	761.0000	732.0000	805.0000	776.0000
475.	E	687.0000	753.0000	725.0000	797.0000	768.0000
476.	T	598.0000	656.0000	631.0000	695.0000	669.0000
477.	Y	598.0000	656.0000	631.0000	695.0000	669.0000
478.	V	711.0000	780.0000	751.0000	826.0000	795.0000
479.	Z	685.0000	751.0000	723.0000	795.0000	766.0000
480.	0410TABLE CAPNUC85, EMPTY, R= 50, C= 50					
481.	*	U1	U2	U3	U4	U5
482.	NO	0.	0.	0.	0.	0.
483.	NY	0.	780.0000	725.0000	510.0000	695.0000
484.	NZ	0.	0.	0.	0.	0.
485.	OL	0.	0.	0.	0.	0.
486.	*	U6	U7	U8	U9	UA
487.	NO	0.	0.	0.	0.	0.
488.	NY	535.0000	835.0000	1070.0000	760.0000	875.0000
489.	NZ	0.	0.	0.	0.	0.
490.	OL	0.	0.	0.	0.	0.
491.	0410TABLE CAPNUC90, EMPTY, R= 50, C= 50					
492.	*	U1	U2	U3	U4	U5
493.	NO	0.	1020.0000	0.	0.	0.
494.	NY	720.0000	815.0000	770.0000	525.0000	635.0000
495.	NZ	735.0000	0.	645.0000	630.0000	670.0000
496.	OL	220.0000	0.	465.0000	175.0000	180.0000
497.	*	U6	U7	U8	U9	UA
498.	NO	835.0000	0.	0.	0.	1070.0000
499.	NY	535.0000	835.0000	1070.0000	745.0000	795.0000
500.	NZ	625.0000	0.	0.	680.0000	0.
501.	OL	320.0000	0.	0.	140.0000	0.
502.	0410TABLE CAPNUC95, EMPTY, R= 50, C= 50					
503.	*	U1	U2	U3	U4	U5

504.	NO	960.0000	975.0000	1000.0000	750.0000	920.0000
505.	NY	720.0000	815.0000	770.0000	520.0000	635.0000
506.	NZ	735.0000	675.0000	665.0000	695.0000	725.0000
507.	OL	220.0000	200.0000	370.0000	195.0000	175.0000
508.	*	U6	U7	U8	U9	UA
509.	NO	845.0000	0.	0.	830.0000	995.0000
510.	NY	535.0000	835.0000	1070.0000	745.0000	795.0000
511.	NZ	685.0000	950.0000	0.	680.0000	925.0000
512.	OL	305.0000	95.0000	0.	140.0000	275.0000

2401TABLE COSTFIXN, EMPTY, R= 50, C= 50

514.	*	COST
515.	1	2.4500
516.	6	0.1800
517.	5	0.1400
518.	7	3.3600
519.	A	2.2800
520.	8	2.2200
521.	9	1.0200
522.	H	1.0200
523.	K	4.9000
524.	N	0.
525.	2	20.0400
526.	3	20.0400
527.	4	20.0400
528.	J	19.2400
529.	Q	19.2400
530.	X	19.2400
531.	M	19.2400
532.	O	19.2400
533.	P	20.0400
534.	E	20.0400
535.	T	19.2400
536.	Y	19.2400
537.	V	20.0400
538.	Z	20.0400

2403TABLE DCF

540.	*	0	Z	Y
541.	1	.0864	.0864	.0864
542.	6	.0643	.0643	.0643
543.	5	.0643	.0643	.0643
544.	7	.0817	.0817	.0817
545.	A	.0817	.0817	.0817
546.	8	.0845	.0845	.0845
547.	9	.0740	.0740	.0740
548.	K	.0740	.0740	.0740
549.	N	.0844	.0844	.0844
550.	U	.0643	.0643	.0643
551.	2	.0984	.0982	.0982
552.	3	.0977	.0970	.0970
553.	4	.0959	.0959	.0959
554.	J	.0914	.0914	.0914
555.	Q	.0914	.0914	.0914
556.	X	.0914	.0914	.0914
557.	M	.0914	.0914	.0914
558.	O	.0914	.0914	.0914
559.	P	.0977	.0970	.0970

560.	E	.0959	.0959	.0959		
561.	T	.0862	.0862	.0862		
562.	Y	.0862	.0862	.0862		
563.	V	.0931	.0921	.0921		
564.	Z	.0907	.0907	.0907		
565.	0130TABLE FCONV					
566.	*	CF1J	CG1Q	CH1X	CI12	CJ13
567.	DATA	10.0	10.0	10.0	10.0	10.0
568.	*	CK14	CL8J	CM8Q	DN8X	CO82
569.	DATA	10.0	10.0	10.0	10.0	10.0
570.	*	CP83	CQ84	CR18	CS18	CT81
571.	DATA	10.0	10.0	11.0	11.0	11.0
572.	*	CU65	CVX2	CWX3	CXQ2	CYOP
573.	DATA	11.0	11.0	11.0	11.0	11.0
574.	*	CZVY	DFX2	DGX3	DHQ2	DIOP
575.	DATA	11.0	11.0	11.0	11.0	11.0
576.	*	DJYV	DK56	DL7A	DM7A	DOA7
577.	DATA	11.0	11.0	11.0	11.0	11.0
578.	0610TABLE TDCAP TABLE OF T&D CAPITAL COSTS					
579.	*	U1	U2	U3	U4	U5
580.	FX85	129.	227.	157.	15.	968.
581.	VR85	59.	63.	68.	74.	68.
582.	FX90	222.	389.	269.	25.	1659.
583.	VR90	59.	63.	68.	74.	68.
584.	FX95	314.	552.	381.	36.	2350.
585.	VR95	59.	63.	68.	74.	68.
586.	*	U6	U7	U8	U9	UA
587.	FX85	363.	485.	525.	1636.	635.
588.	VR85	77.	74.	67.	67.	67.
589.	FX90	622.	832.	900.	2805.	1089.
590.	VR90	77.	74.	67.	67.	67.
591.	FX95	881.	1178.	1274.	3973.	1542.
592.	VR95	77.	74.	67.	67.	67.
593.	0610TABLE TDOM TABLE OF T&D O&M COSTS					
594.	*	U1	U2	U3	U4	U5
595.	FX85	74.6	124.4	84.1	97.2	143.2
596.	VR85	5.1	5.0	2.1	1.1	1.4
597.	FX90	80.0	133.0	86.3	97.3	152.8
598.	VR90	5.1	5.0	2.1	1.1	1.4
599.	FX95	85.2	141.5	88.6	97.4	162.5
600.	VR95	5.1	5.0	2.1	1.1	1.4
601.	*	U6	U7	U8	U9	UA
602.	FX85	45.7	32.6	23.1	105.9	77.7
603.	VR85	0.6	1.1	0.8	1.0	2.6
604.	FX90	47.1	36.0	26.1	117.0	89.2
605.	VR90	0.6	1.1	0.8	1.0	2.6
606.	FX95	48.6	39.5	29.0	128.1	100.7
607.	VR95	0.6	1.1	0.8	1.0	2.6

?

iii) LISTING OF NUF5.CREMOD.DATA

1.	79	1	0.0	0.0	0.0	0.0	0.02	0.0	0.0	0.0	0.0	0.0
2.	80	1	0.0	0.0	0.0	0.0	0.02	0.0	0.0	0.0	0.0	0.0
3.	81	1	0.0	0.0	0.0	0.0	0.02	0.0	0.0	0.0	0.0	0.0
4.	82	1	0.0	0.0	0.0	0.0	0.02	0.0	0.0	0.0	0.0	0.0
5.	83	1	0.0	0.0	0.0	0.0	0.02	0.0	0.0	0.0	0.0	0.0
6.	84	1	0.0	0.0	0.0	0.0	0.02	0.0	0.0	0.0	0.0	0.0
7.	85	1	0.0	0.0	0.0	0.0	0.02	0.0	0.0	0.0	0.0	0.0
8.	86	1	0.0	0.0	0.0	0.0	0.02	0.0	0.0	0.0	0.0	0.0
9.	87	1	0.0	0.0	0.0	0.0	0.02	0.0	0.0	0.0	0.0	0.0
10.	88	1	0.0	0.0	0.0	0.0	0.02	0.0	0.0	0.0	0.0	0.0
11.	89	1	0.0	0.0	0.0	0.0	0.02	0.0	0.0	0.0	0.0	0.0
12.	90	1	0.0	0.0	0.0	0.0	0.02	0.0	0.0	0.0	0.0	0.0
13.	79	2	0.0	0.85	0.0	0.16	1.54	0.48	0.0	0.04	0.0	0.0
14.	80	2	0.0	0.85	0.0	0.94	1.54	0.48	0.0	0.04	0.14	0.0
15.	81	2	0.0	0.85	0.0	1.71	1.54	0.48	0.0	0.04	0.14	0.10
16.	82	2	0.0	0.85	1.21	1.71	1.54	0.48	0.0	0.04	0.16	0.10
17.	83	2	0.0	0.85	1.21	1.71	1.54	0.48	0.0	0.04	0.16	0.10
18.	84	2	0.0	0.85	1.21	1.71	1.54	0.48	0.0	0.04	0.16	0.10
19.	85	2	0.0	0.85	1.21	1.71	1.54	0.48	0.0	0.04	0.16	0.10
20.	86	2	0.0	0.85	1.21	1.71	1.54	0.48	0.0	0.04	0.16	0.10
21.	87	2	0.0	0.85	1.21	1.71	1.54	0.48	0.0	0.04	0.16	0.10
22.	88	2	0.0	0.85	1.21	1.71	1.54	0.48	0.0	0.04	0.16	0.10
23.	89	2	0.0	0.85	1.21	1.71	1.54	0.48	0.0	0.04	0.16	0.10
24.	90	2	0.0	0.85	1.21	1.71	1.54	0.48	0.0	0.04	0.16	0.10
25.	79	3	0.0	0.14	0.63	0.42	1.02	1.62	0.0	1.76	0.69	0.0
26.	80	3	0.0	0.28	2.08	1.41	1.63	2.43	0.68	3.65	1.28	0.0
27.	81	3	0.0	0.43	2.08	2.49	2.75	3.38	1.52	4.47	1.63	0.0
28.	82	3	0.0	0.57	2.08	2.50	4.17	4.66	1.67	5.67	1.63	0.0
29.	83	3	0.0	0.71	2.08	2.50	4.29	5.07	1.67	5.67	1.63	0.0
30.	84	3	0.0	0.85	2.08	2.50	4.40	6.22	1.67	5.67	1.96	0.0
31.	85	3	0.11	1.22	3.27	4.96	5.00	8.11	1.67	5.67	2.74	0.0
32.	86	3	0.23	1.59	4.47	7.42	5.60	10.00	1.67	5.67	3.53	0.0
33.	87	3	0.34	1.96	5.67	9.88	6.19	11.89	1.67	5.67	4.31	0.0
34.	88	3	0.45	2.33	6.87	12.34	6.79	13.78	1.67	5.67	5.09	0.0
35.	89	3	0.57	2.70	8.06	14.79	7.38	15.67	1.67	5.67	5.87	0.0
36.	90	3	1.62	3.34	12.81	17.98	7.38	23.50	2.19	5.98	10.74	0.0
37.	79	4	0.0	0.0	0.0	0.0	0.96	2.75	1.75	0.20	0.0	0.0
38.	80	4	0.0	0.0	1.70	0.72	2.33	6.90	2.62	0.67	0.0	0.53
39.	81	4	0.0	0.0	1.70	3.17	2.93	8.61	4.18	1.17	0.25	0.53
40.	82	4	0.0	0.0	1.70	6.01	5.11	12.04	5.16	1.55	0.25	0.53
41.	83	4	0.0	0.0	1.70	6.53	5.11	14.45	5.16	1.55	0.25	0.53
42.	84	4	0.0	0.0	1.70	7.13	5.11	15.50	5.16	1.55	0.50	0.53
43.	85	4	0.0	0.0	1.70	7.41	5.79	15.61	5.54	1.55	0.50	0.53
44.	86	4	0.0	0.0	1.70	7.69	6.47	15.71	5.54	1.55	0.50	0.53
45.	87	4	0.0	0.0	1.70	7.97	7.15	15.81	5.54	1.55	0.50	0.53
46.	88	4	0.0	0.0	1.70	8.25	7.82	15.92	5.54	1.55	0.50	0.53
47.	89	4	0.0	0.0	1.70	8.53	8.50	16.02	5.54	1.55	0.50	0.53
48.	90	4	0.0	0.0	1.70	8.53	8.50	16.02	5.59	1.55	0.50	0.53
49.	79	5	0.0	0.0	0.45	0.20	0.0	0.0	0.04	0.0	0.08	0.0
50.	80	5	0.0	0.0	0.90	0.40	0.0	0.0	0.07	0.0	0.16	0.0
51.	81	5	0.0	0.0	1.35	0.61	0.0	0.0	0.11	0.0	0.24	0.0
52.	82	5	0.0	0.0	1.80	0.81	0.0	0.0	0.15	0.0	0.32	0.0
53.	83	5	0.0	0.0	2.25	1.01	0.0	0.0	0.18	0.0	0.40	0.0
54.	84	5	0.0	0.0	2.70	1.21	0.0	0.0	0.22	0.0	0.48	0.0
55.	85	5	0.0	0.60	3.24	1.41	0.0	0.12	0.22	0.0	0.66	0.0

56.	86	5	0.0	1.20	3.78	1.61	0.0	0.24	0.22	0.0	0.84	0.0
57.	87	5	0.0	1.80	4.33	1.81	0.0	0.36	0.22	0.0	1.01	0.0
58.	88	5	0.0	2.40	4.87	2.01	0.0	0.48	0.22	0.0	1.19	0.0
59.	89	5	0.0	3.00	5.41	2.21	0.0	0.60	0.22	0.0	1.37	0.0
60.	90	5	0.01	3.10	6.04	2.47	0.0	0.67	0.32	0.0	1.49	0.0
61.	79	6	0.0	0.02	0.04	0.16	0.01	0.01	0.01	0.0	0.20	0.58
62.	80	6	0.0	0.03	0.07	0.32	0.03	0.01	0.01	0.0	0.40	1.15
63.	81	6	0.01	0.05	0.11	0.48	0.04	0.02	0.02	0.0	0.60	1.73
64.	82	6	0.01	0.06	0.15	0.63	0.05	0.02	0.02	0.0	0.80	2.30
65.	83	6	0.01	0.08	0.18	0.79	0.07	0.03	0.03	0.0	1.00	2.88
66.	84	6	0.01	0.09	0.22	0.95	0.08	0.03	0.03	0.0	1.20	3.45
67.	85	6	0.03	0.10	0.22	0.96	0.08	0.03	0.03	0.0	1.22	3.45
68.	86	6	0.05	0.11	0.22	0.98	0.08	0.03	0.03	0.0	1.25	3.45
69.	87	6	0.06	0.12	0.22	0.99	0.08	0.03	0.03	0.0	1.27	3.45
70.	88	6	0.08	0.12	0.22	1.01	0.08	0.03	0.03	0.0	1.30	3.45
71.	89	6	0.09	0.13	0.22	1.02	0.08	0.03	0.03	0.0	1.32	3.45
72.	90	6	0.18	0.14	0.28	1.08	0.10	0.06	0.03	0.31	1.41	4.06
73.	79	7	0.0	0.0	0.0	0.79	0.0	0.0	0.0	0.33	0.0	0.0
74.	80	7	0.0	0.0	0.94	1.93	0.0	0.91	0.0	0.33	1.08	0.0
75.	81	7	0.0	1.11	0.94	4.01	1.89	0.91	0.0	0.33	2.19	0.0
76.	82	7	0.0	1.11	0.94	8.41	1.89	0.91	0.0	0.33	3.33	0.0
77.	83	7	0.0	1.97	1.99	10.76	5.21	3.23	0.0	0.33	4.47	1.10
78.	84	7	0.0	1.97	3.04	13.14	7.36	4.48	2.30	0.33	5.78	1.10
79.	85	7	1.19	1.97	3.04	15.21	10.42	5.63	2.30	0.33	5.78	2.37
80.	86	7	1.19	4.12	5.87	18.46	12.03	6.88	2.30	0.33	7.08	3.61
81.	87	7	3.54	4.12	5.87	21.58	13.16	6.88	2.30	0.33	7.08	4.88
82.	88	7	3.54	4.12	6.93	21.82	14.37	7.81	2.30	0.33	8.05	4.88
83.	89	7	3.54	5.18	6.93	21.82	16.16	7.81	2.30	0.33	8.05	4.88
84.	90	7	3.54	5.18	6.93	23.54	16.16	7.81	2.30	0.33	8.05	4.88
85.	79	8	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.22	0.01
86.	80	8	0.11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.43	0.03
87.	81	8	0.17	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.65	0.04
88.	82	8	0.23	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.86	0.05
89.	83	8	0.28	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.08	0.07
90.	84	8	0.34	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.29	0.08
91.	85	8	0.34	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.29	0.08
92.	86	8	0.34	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.29	0.08
93.	87	8	0.34	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.29	0.08
94.	88	8	0.34	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.29	0.08
95.	89	8	0.34	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.29	0.08
96.	90	8	0.34	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.29	0.08
97.	79	9	0.17	0.01	0.0	0.13	0.04	1.10	0.49	0.03	0.06	0.02
98.	80	9	0.33	0.02	0.0	0.26	0.08	2.20	0.99	0.05	0.13	0.04
99.	81	9	0.50	0.03	0.01	0.38	0.12	3.30	1.48	0.08	0.19	0.05
100.	82	9	0.67	0.04	0.01	0.51	0.16	4.40	1.97	0.11	0.25	0.07
101.	83	9	0.83	0.05	0.01	0.64	0.20	5.50	2.47	0.14	0.32	0.09
102.	84	9	1.00	0.06	0.01	0.77	0.24	6.60	2.96	0.16	0.38	0.11
103.	85	9	1.00	0.06	0.01	1.51	0.24	7.00	3.27	0.16	0.89	0.11
104.	86	9	1.00	0.06	0.01	2.25	0.24	7.40	3.57	0.16	1.39	0.11
105.	87	9	1.00	0.06	0.01	2.98	0.24	7.80	3.88	0.16	1.90	0.11
106.	88	9	1.00	0.06	0.01	3.72	0.24	8.20	4.18	0.16	2.41	0.11
107.	89	9	1.00	0.06	0.01	4.46	0.24	8.60	4.49	0.16	2.91	0.11
108.	90	9	1.00	0.06	0.01	4.46	0.24	9.18	4.76	0.22	3.49	0.11
109.	79	10	0.27	0.20	0.29	0.29	0.09	0.0	0.0	0.0	0.0	0.0
110.	80	10	0.54	0.40	0.59	0.59	0.19	0.0	0.0	0.0	0.0	0.0
111.	81	10	0.80	0.60	0.88	0.88	0.28	0.0	0.0	0.0	0.0	0.0

112.	82 10	1.07	0.81	1.18	1.17	0.37	0.0	0.0	0.0	0.0	0.0
113.	83 10	1.34	1.01	1.47	1.46	0.46	0.0	0.0	0.0	0.0	0.0
114.	84 10	1.61	1.21	1.76	1.76	0.56	0.0	0.0	0.0	0.0	0.0
115.	85 10	1.61	1.21	1.76	1.76	0.56	0.0	0.0	0.0	0.0	0.0
116.	86 10	1.61	1.21	1.76	1.76	0.56	0.0	0.0	0.0	0.0	0.0
117.	87 10	1.61	1.21	1.76	1.76	0.56	0.0	0.0	0.0	0.0	0.0
118.	88 10	1.61	1.21	1.76	1.76	0.56	0.0	0.0	0.0	0.0	0.0
119.	89 10	1.61	1.21	1.76	1.76	0.56	0.0	0.0	0.0	0.0	0.0
120.	90 10	1.61	1.21	1.76	1.76	0.56	0.0	0.0	0.0	0.0	0.0
121.	79 11	0.01	0.03	0.03	0.30	0.41	0.81	0.0	0.0	1.56	0.0
122.	80 11	0.03	0.05	0.05	0.60	0.83	1.62	0.0	0.0	3.12	0.0
123.	81 11	0.04	0.08	0.08	0.90	1.24	2.42	0.0	0.0	4.69	0.0
124.	82 11	0.05	0.11	0.10	1.20	1.66	3.23	0.0	0.0	6.25	0.0
125.	83 11	0.07	0.14	0.13	1.50	2.07	4.04	0.0	0.0	7.81	0.0
126.	84 11	0.08	0.16	0.15	1.80	2.48	4.85	0.0	0.0	9.37	0.0
127.	85 11	0.08	0.49	0.15	1.80	2.48	6.59	0.10	0.0	9.37	0.0
128.	86 11	0.08	0.82	0.15	1.80	2.48	8.32	0.20	0.0	9.37	0.0
129.	87 11	0.08	1.15	0.15	1.80	2.48	10.06	0.29	0.0	9.37	0.0
130.	88 11	0.08	1.48	0.15	1.80	2.48	11.80	0.39	0.0	9.37	0.0
131.	89 11	0.08	1.81	0.15	1.80	2.48	13.53	0.49	0.0	9.37	0.0
132.	90 11	0.08	1.81	0.15	1.80	2.48	13.53	0.49	0.0	9.37	0.0
133.	79 12	0.12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.59	0.0
134.	80 12	0.24	0.0	0.01	0.01	0.01	0.0	0.0	0.0	1.17	0.0
135.	81 12	0.35	0.0	0.01	0.01	0.01	0.01	0.0	0.0	1.76	0.0
136.	82 12	0.47	0.0	0.02	0.01	0.02	0.01	0.0	0.0	2.35	0.0
137.	83 12	0.59	0.01	0.02	0.02	0.02	0.01	0.0	0.0	2.93	0.0
138.	84 12	0.71	0.01	0.02	0.02	0.02	0.01	0.0	0.0	3.52	0.0
139.	85 12	1.03	0.14	0.12	0.42	0.03	0.62	0.0	0.01	3.97	0.0
140.	86 12	1.35	0.26	0.22	0.81	0.03	1.23	0.0	0.02	4.41	0.0
141.	87 12	1.67	0.39	0.32	1.21	0.03	1.84	0.0	0.03	4.86	0.0
142.	88 12	2.00	0.52	0.42	1.61	0.03	2.44	0.0	0.04	5.31	0.0
143.	89 12	2.32	0.65	0.52	2.00	0.03	3.05	0.0	0.05	5.76	0.0
144.	90 12	3.35	1.55	0.56	3.42	0.84	3.47	0.23	0.05	6.67	0.0

?