SDC-ISI-TSC-80-1024

FEBRUARY 1980

INTERNATIONAL AIR PASSENGER TRAFFIC MODEL: GUIDE FOR CREATION OF QSI AND WATER.LEG FILES

Prepared by

John Bannick

SDC Intregrated Services, Inc.
ADP Support Services Project
Contract DOT-TSC-1745
Cambridge, Massachusetts

PROJECT REPORT

ADP JOB ORDER D0182A

THIS DOCUMENT CONTAINS PRELIMINARY INFORMATION SUBJECT TO CHANGE. IT IS CONSIDERED AN INTERNAL TSC WORKING PAPER WITH A SELECT DISTRIBUTION MADE BY THE AUTHOR. IT IS NOT A FORMAL REFERABLE REPORT.

Prepared for

Simon Prensky/200

through
Computer Services Division
DOT Transportation Systems Center
Cambridge, Massachusetts

ABSTRACT

This guide is partial documentation for the International Air Passenger Traffic Model. Its purpose is to show the new user how to create some of the data files necessary to run the model. It is also intended to present such information as is available to aid in the understanding and modification of the programs utilized.

A general overview of the model and the creation process for the other data files it needs can be found in "Guide to Programs Implementing the Air Passenger Traffic Model", SDC-ISI-TSC-80-1021, February 1980.

TABLE OF CONTENTS

Page

1.0 INTRODUCTION.....1

2.0 BASE CASE.....7

3.0 ALTERNATE SCENARIOS.....41

APPENDIX A
Program Listings:
OAGCOP.FOR, USFLT.FOR, UEFLT.FOR, EUFLT.FOR, CONSEG.FOR, READLL.FOR, REUDLL.FOR, JETAPT.FOR, JEUAPT.FOR, IADATE.FOR, FLTLEG.FOR, LEGDAY.FOR, PRDIS.FOR, USHFLR.FOR, EUHFLR.FOR, PDUPDU.FOR, PDEPDE.FOR, OPUOPU.FOR, OPEOPE.FOR, USEUHP.FOR, USPEUP.FOR, EUPUSP.FOR, UEHNC.FOR, SCENAR.FOR, SCODFL.FOR, ROUTE., PR3.....A1

APPENDIX B
Control File Listings:
BSCNOD.CTL, FLT.CTL, HFILES.CTL, BSCOOD.CTL.....B1

LIST OF FIGURES

Page

FIGURE 1.1
Base Case Data Flow.....4

FIGURE 1.2
Alternate Scenario Data Flow From Base Case Process.....5

FIGURE 2.1
Base Case QSI and WATER.LEG Development Process.....8

FIGURE 2.2
Record Format for OAG Tapes, OAGCOP.DAT..... 10

FIGURE 2.3
Record Format for OAGCOP.DAT..... 14

FIGURE 2.4
Record Format for USLL.DAT, EULL.DAT..... 15

FIGURE 2.5
Record Format for USFLT.DAT, EUFLT.DAT, UEFLT.DAT..... 16

FIGURE 2.6
Record Format for: USPR.DAT, UEPR.DAT, EUPR.DAT, DEPUS.DAT, DEPEU.DAT, ARRUS.DAT, ARRUE.DAT, ARREU.DAT, USPEUP.DAT, EUPUSP.DAT, OPUOPU.DAT, OPEOPE.DAT, PDUPDU.DAT, PDEPDE.DAT...20

FIGURE 2.7
Record Format for UEDIS.DAT.....23

FIGURE 2.8
Process Creating the Files of Selected Flights.....24

FIGURE 2.9
Record Format for HFILE.DAT, HFILE2.DAT.....25

FIGURE 2.10
Process further Selecting Transatlantic and Connecting Flights.....26

FIGURE 2.11
Record Format for CONNEC.DAT, CONECT.DAT.....34

FIGURE 2.12
Record Format for PRTNM.DAT..... 35

FIGURE 2.13
Record Format for SCENAR.DAT.....36

FIGURE 2.14
Record Format for SCODRT.DAT, BSCnOD.DAT.....37

FIGURE 2.15
Record Format for WATER.LEG.....38

LIST OF FIGURES (Continued)

Page

FIGURE 2.16
Process Creating QSI and WATER.LEG Files..... 40

1.0 INTRODUCTION

This guide is partial documentation for the International Air Passenger Traffic Model. Its purpose is to show the new user how to create some of the data files necessary to run the model. It is also intended to present such information as is available to aid in the understanding and modification of the programs utilized.

A general overview of the model and the creation process for the other data files it needs can be found in "Guide to Programs Implementing the Air Passenger Traffic Model", SDC-ISI-TSC-80-1021, Feb. 1980.

The model requires three data files:

A. A Base Case traffic file, elsewhere referred to as FFILE.DAT or the "demand file" quantifying air passenger traffic on itineraries between two regions for a given three month period. For the development project, the two regions were defined as:

   1. The "greater US" (US), consisting of the Western Hemisphere.
   2. "Greater Europe" (EUR), consisting of Europe, Africa, the Mideast, India, Sri Lanka, and Afghanistan.

   The three month period consisted of July-September 1977 (3Q77) although traffic totals for the period October 1976-September 1977 (CY77) were also a factor.

B. A Base Case Quality of Service Index (QSI) File, referred to elsewhere as the "supply file", or QSI1.DAT, quantifying the best attainable levels of access between directional origin and destination cities via certain transatlantic routes, as a function of scheduled airline flights.

C. An Alternate Scenario QSI File, elsewhere referred to as QSI2n.DAT, quantifying the best attainable levels of access on those routes given alternate flight schedules. Key variables in changing flight schedules are:

1. days of operation, and
2. intermediate stops

The model outputs an Alternate Scenario Traffic File predicting the air passenger traffic given the alternate flight schedules. This file is passed through a pre-processor program and then input to a financial modeling program to predict certain ecomonic effects of the Alternate Scenario. This output is then compared to that obtained by using the same financial model on the (pre-processed) Base Case Traffic File.

In addition to the traffic and QSI files, a third set of files, WATER.LEG for the Base Case and WATERn.LEG for the Alternate Scenarios, is needed. These are Binary Image files similar to QSI files. WATER.LEG is produced during the Base Case QSI production process as a necessary input to the later stages of the Base Case Traffic File production process. WATER.LEG and the WATERn.LEG files are also necessary for the pre-processing of the Base Case and Alternate Scenario Traffic Files, respecttively, prior to running the financial model.

The information in this guide is taken from "The ATPM Flight LOS and QSI Programs" (draft), Nov. 1979, produced by H. H. Aerospace Design, the principal implementers of the programs discussed. At present, there has been no verification that the programs perform the processes desired, merely that they do

execute and produce QSI and WATER.LEG files. All programs and control files have been modified to use private disk pack PT49.

The approach taken here will be to explain the general process performed by each of the five Base Case steps and five Alternate Scenario steps, the general purpose of each of their constituant programs or manual processes, the specific elements of the manual processes, and to describe all data files involved in those processes. Appendices containing full listings of all programs and control files are included. See Figures 1.1 and 1.2. All H. H. Aerospace programs are backed up on tape 3574, "IAPT-PROGS" at 1600 BPI.

FIGURE 1.1 BASE CASE DATA FLOW

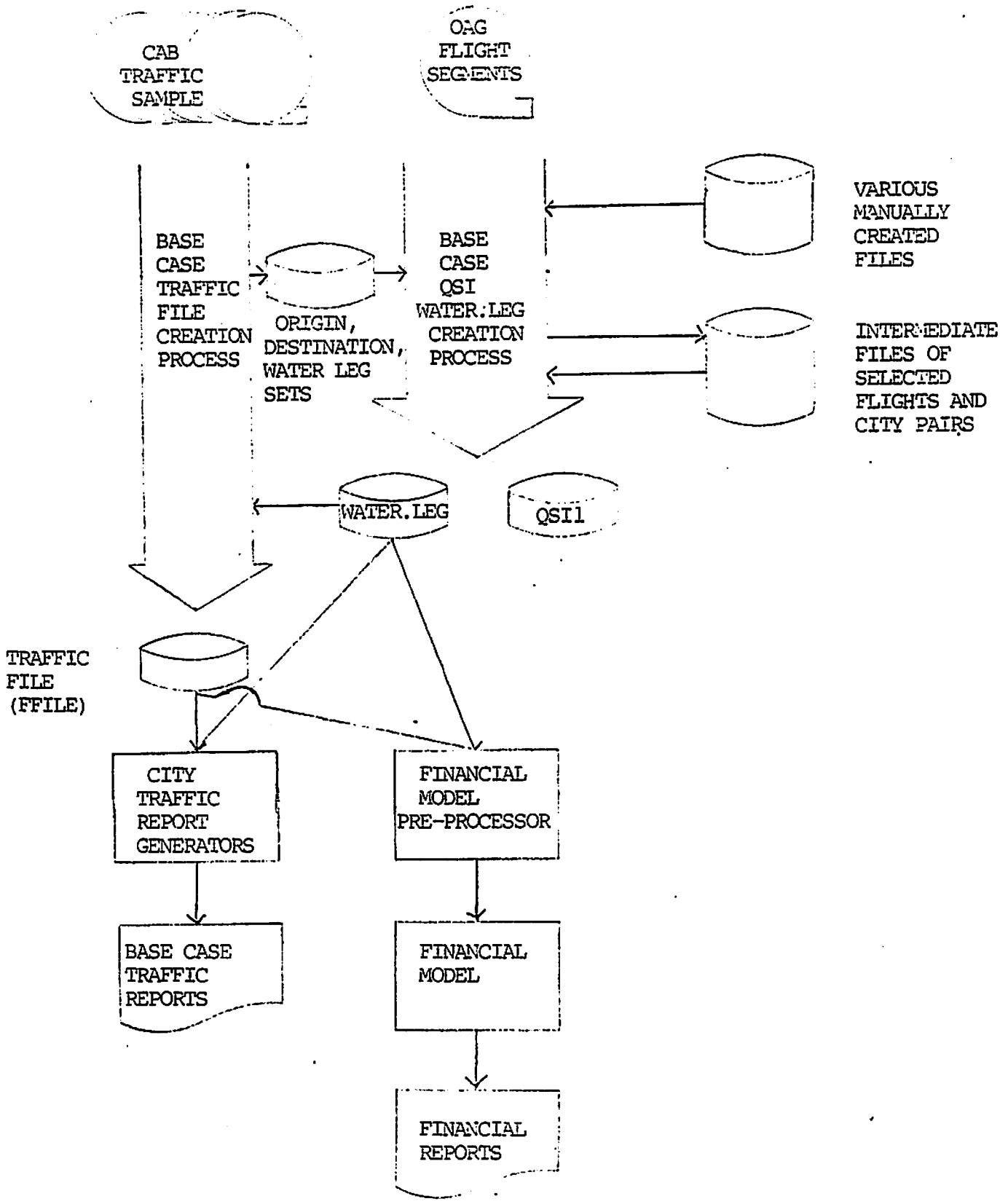


FIGURE 1.2 ALTERNATE SCENARIO DATA FLOW

FROM BASE CASE PROCESS

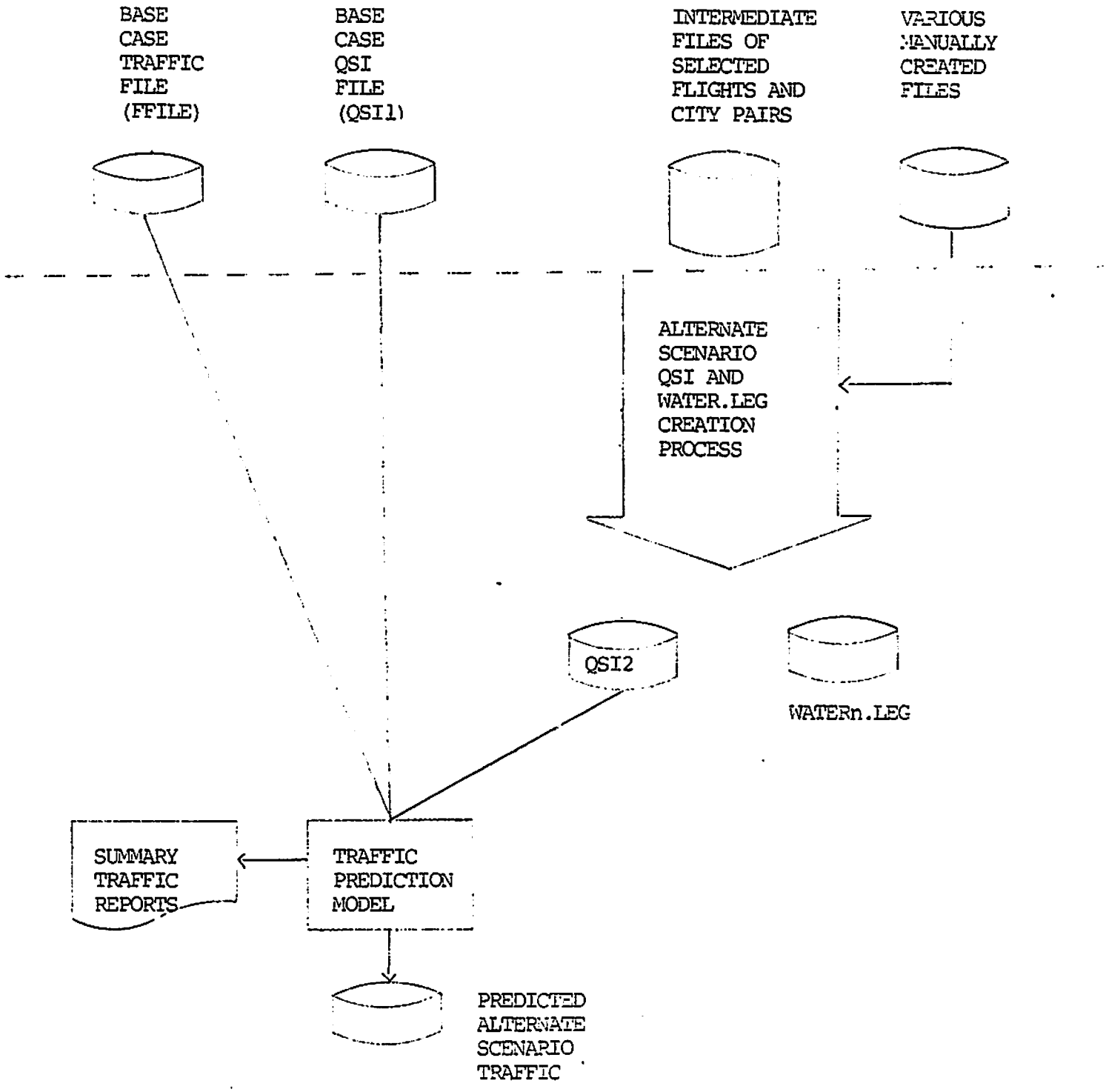
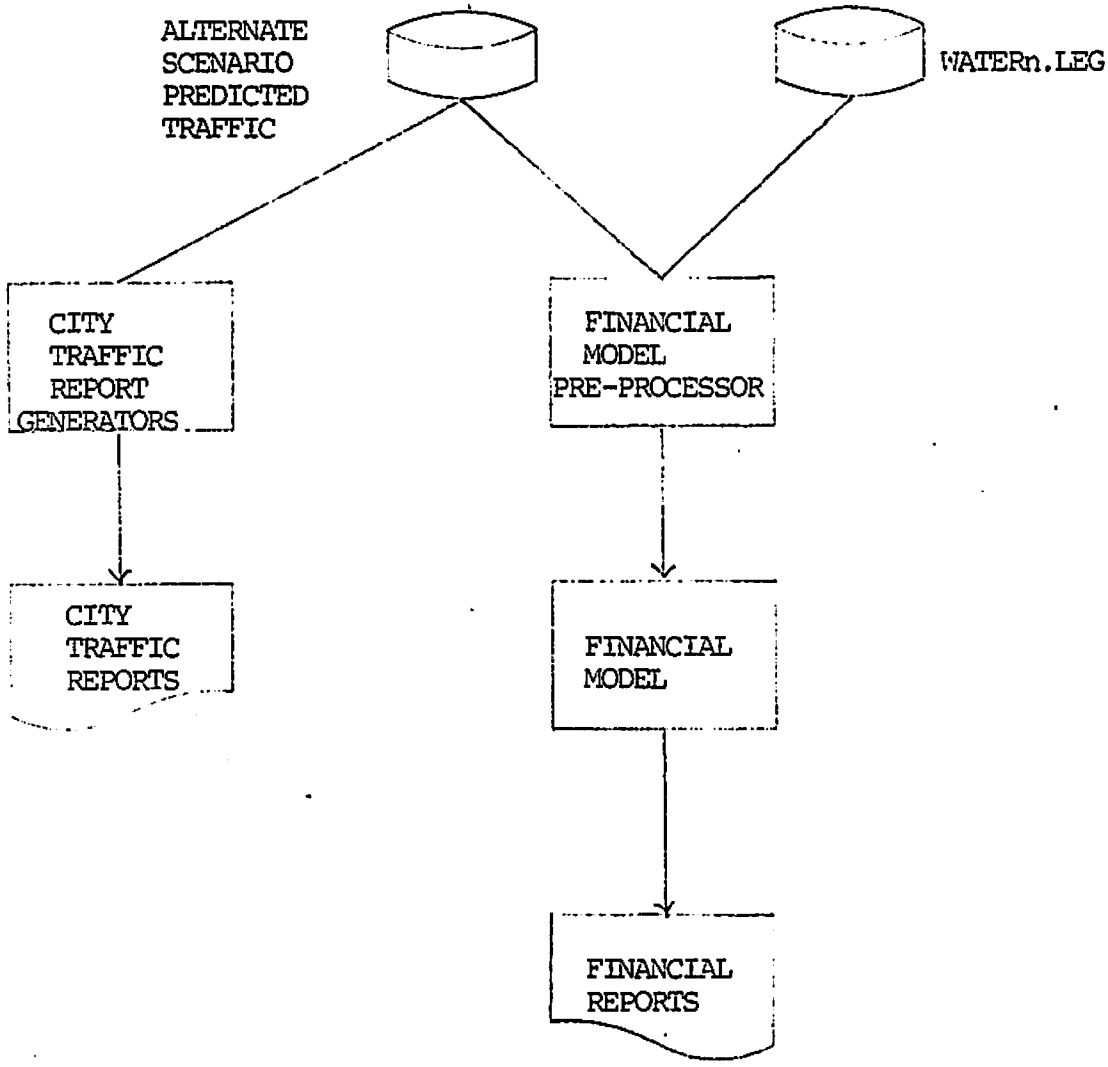


FIGURE 1.2 (Continued)



2.0 BASE CASE

Step 1: Copy Flight Data Off of Source Tape

1. Input:

   One Official Airline Guide (OAG) tape with a single file containing worldwide flight schedule data for a two week period. At present, these tapes have a blocksize of 2200, BPI of 800, and approximately 68K records. Each record is an airport to airport flight segment. See Figure 2.1.

2. Output:

   OAGCOP.DAT: an ASCII file presently containing approximately 64K records, in 11K blocks, in the same format as the source tape.

3. Process:

   a. Insure PT49 is mounted.

   b. Mount the selected OAG tape, setting the logical unit to 16. Note that the present blocksize is set by the program.

   c. Since the tape has a "dummy" first file containing two records of internal label, key in the command:

      SKIP 16:1 FILE

   d. Execute the program OAGCOP.FOR using the command:

      EXECUTE OAGCOP.FOR.

   e. Dismount the tape.

4. Programs:

   OAGCOP.FOR is the only program in this step. It reads the OAG tape and copies onto OAGCOP.DAT all passenger flight records, i.e., those with coupon type not blank.

Step 2: Create the Flight Files

1. Inputs:

   a. OAGCOP.DAT from Step 1.

FIGURE 2.1 BASE CASE QSI AND WATER.LEG DEVELOPMENT PROCESS

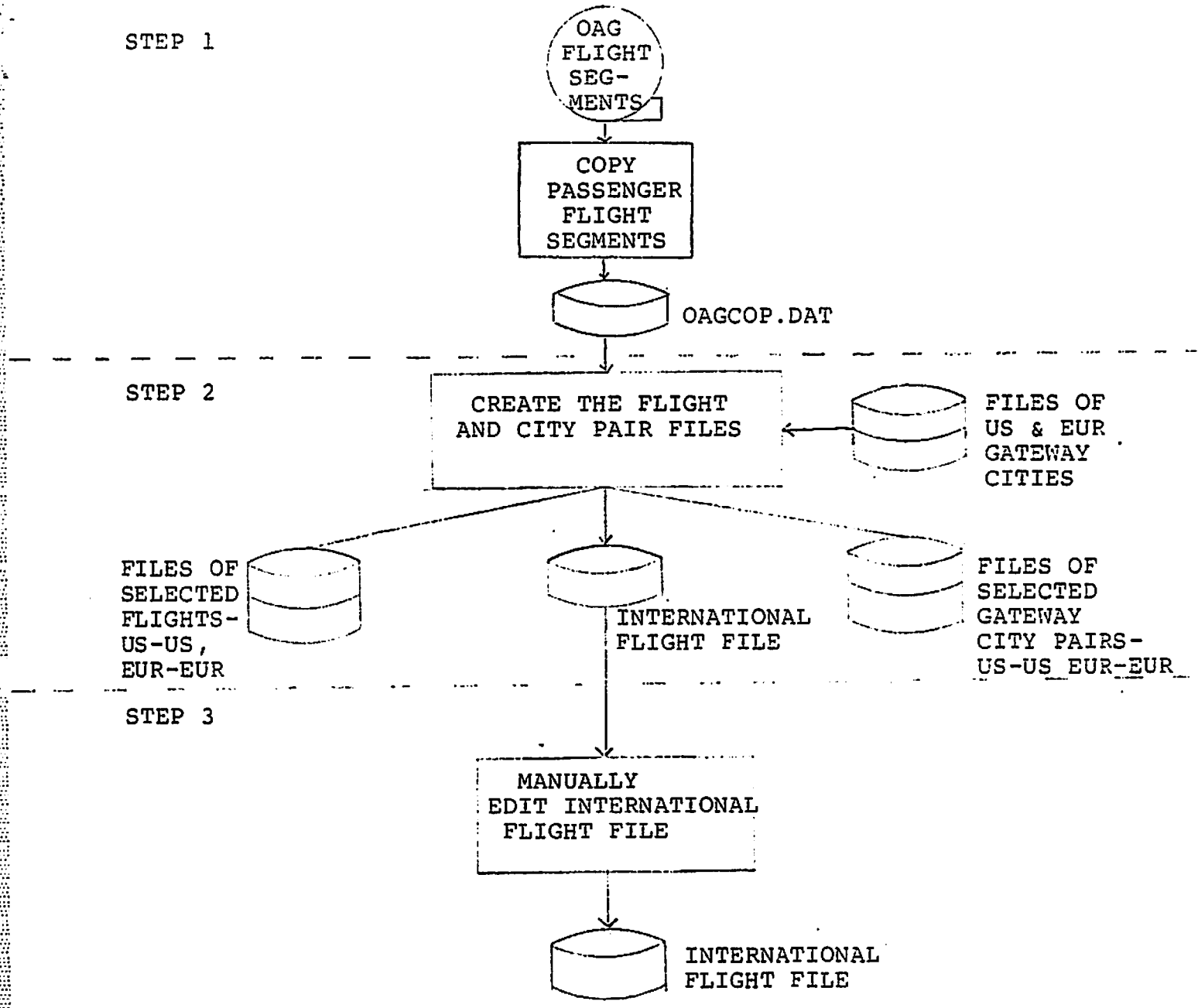
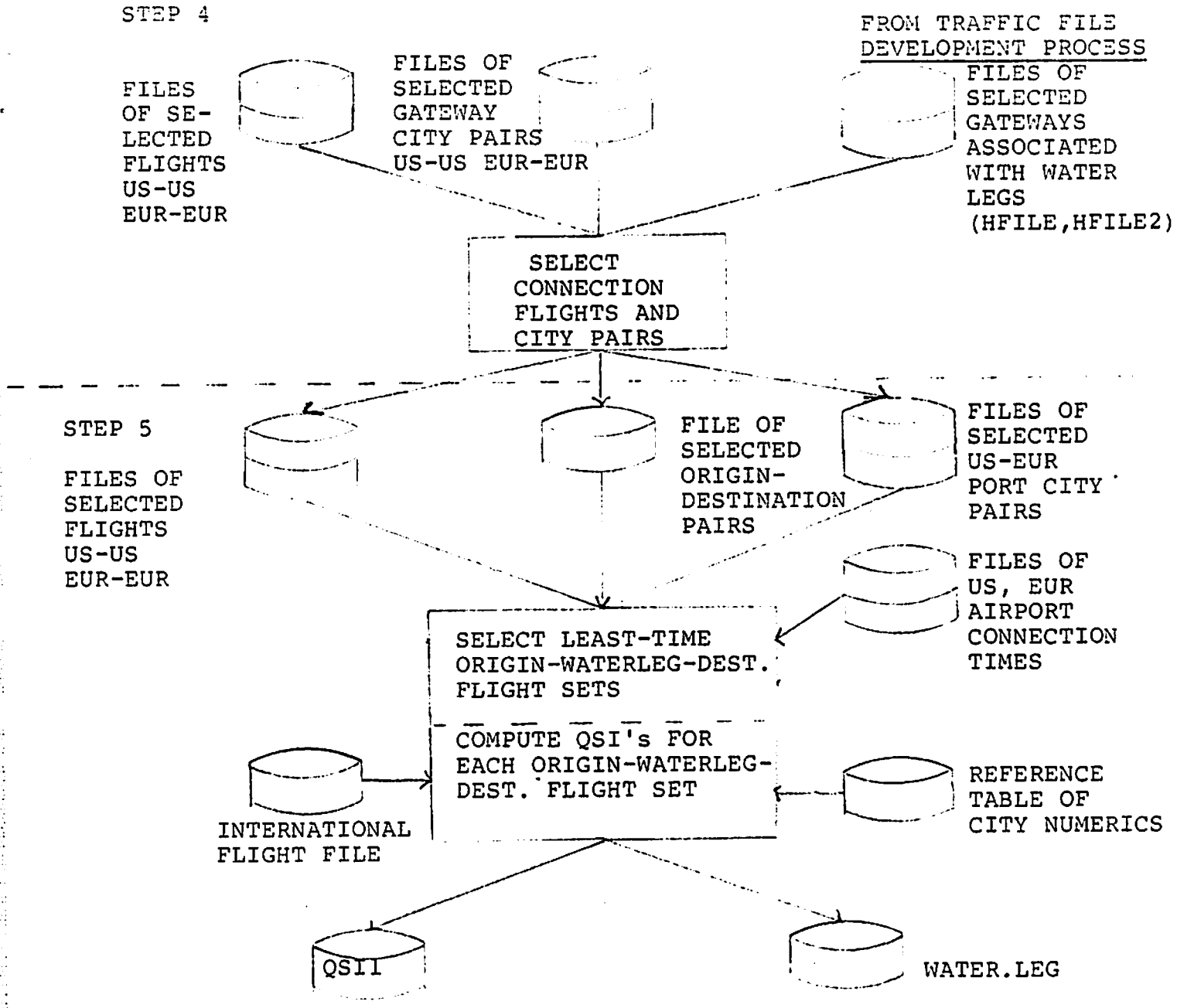


FIGURE 2.1 Continued



Record Format for OAG Tapes, OAGCOP.DAT

Field	Format	Description
1	A1	Departure World Area Code, 1st digit
2	A2	Departure World Area Code, rest
3	A1	Filler
4	A3	Departure airport alpha code
5	A2	Filler
6	A4	Departure time, local
7	A4	Departure time, GMT
8	A1	Arrival WAC, 1st digit
9	A2	Arrival WAC, rest
10	A1	Filler
11	A3	Arrival airport alpha code
12	A2	Filler
13	A4	Arrival time, local
14	A1	Flag code
15	A4	Arrival time, GMT
16	A1	Equipment type
17	A3	Equipment code
18	A1	Filler
19	A2	Carrier
20	I5	Flight number
21	A5	Class of service
22	7I1	Days of operation
To		Days of operation
28		Days of operation
29	A1	Suppress code
30	A1	ATIS code

FIGURE 2.2

Field	Format	Description
31	A4	Elapsed time
32	A4	Effective date
33	A4	Discontinued date
34	I6	Leave latitude (secs.÷324K)
35	A5	Leave longitude magnitude (10 secs.)
36	A1	Leave longitude zone sign
37	I6	Arrive latitude
38	A5	Arrive longitude magnitude
39	A1	Arrive longitude zone sign
40	11A1	Filler
	108	

FIGURE 2.2 (continued)

b. USLL.DAT: a manually created ASCII file of the selected US gateway airports - contains latitude and longitude, airport alpha and numeric codes. Note: the OAG lacks the numeric airport codes needed later for linking of flight schedule and traffic data. Also, the OAG latitude and longitude notation is a bit complex. At present, the programs using this file are set to read only the first 100 records. See Figure 2.2. Note: "gateway", or flight origin and destination airports should not be confused with "port" airports, which are the airports immediately on either side of the transatlantic waterleg. Also, note that some cities have more than one airport. At present, approximately 65 airports in 31 cities are in this file.

c. EULL.DAT: a manually created ASCII file of the selected EUR gateway airports--corresponds to USLL.DAT.

2. Output:

a. USFLT.DAT: an ASCII file of connected flight segments over a single flight number, ordered by flight direction. The file contains only those flights which connect any two US gateway airports, regardless of other connections. Records are variable length, with two levels of repeating groups. The fixed part gives general schedule data for the flight number. This is followed by a variable number of multifield data elements giving data on each flight segment. The whole flight record is repeated, with appropriate changes in the values of individual variables, when the flight schedule varies over different days of

the week. (Each version of the record for a flight is termed a "variant".) Flight segments beyond the first and last gateway airports are ignored. See Figure 2.3.

b. EUFLT.DAT: a corresponding file for flights connecting two EUR gateway airports.

c. UEFLT.DAT: a corresponding file for flights connecting a US to a EUR gateway airport in either direction.

d. USPR.DAT: an ASCII file containing one record for each USFLT.DAT gateway airport pair flight variant - summarizes data over the entire flight. See Figure 2.4.

e. EUPR.DAT: a corresponding file for EUR-EUR flights.

f. UEPR.DAT: a corresponding file for US-EUR flights in either direction.

g. DEPUS.DAT: USPR.DAT sorted by departure city.

h. DEPEU.DAT: EUPR.DAT sorted by departure city.

i. DEPUE.DAT: UEPR.DAT sorted by departure city.

j. ARRUS.DAT: USPR.DAT sorted by arrival city.

k. ARREU.DAT: EUPR.DAT sorted by arrival city

l. ARRUE.DAT: UEPR.DAT sorted by arrival city

m. UEDIS.DAT, the "International Flight File": an ASCII file containing one flight record for each transatlantic gateway airport pair variant. The fixed part of each record gives summary data over the entire flight, and the repeating parts give summary flight data for each segment. Note that this file gives much less data than the PR series and is a table of reference for later processes. See Figure 2.5.

Record Format for OAGCOP.DAT

Field	Format	Description
1	A1	Departure WAC, 1st digit
2	16A1	Dummy
3	A1	Arrival WAC, 1st digit
4	29A1	Dummy
5	A5	Fare class
6	58A1	Dummy
	110	

FIGURE 2.3

Record Format for USLL.DAT, EULL.DAT

Field	Format	Description
1	A3	Airport alpha code
2	I2	Degrees of latitude
3	I2	Minutes of latitude
4	I2	Seconds of latitude
5	A1	Hemisphere of latitude (N,S)
6	I3	Degrees of longitude
7	I2	Minutes of longitude
8	I2	Seconds of longitude
9	A1	Hemsphere of longitude
10	I6	City/Airport numeric code

FIGURE 2.4

Record Format for: USFLT.DAT, EUFLT.DAT, UEFLT.DAT

Field	Format	Description
1	1X	Filler
2	A2	Carrier
3	A5	Flight number
4	I1	Flight variant
5	I2	No. days operation per week
6	I2	No. of flight stages

(The following fields are repeated 1 to N times depending upon the number of variants for a given flight number.)

6	1X	Filler
7	I6	Departure city/airport numeric
8	I6	Arrival city/airport numeric
9	A3	Equipment code
10	1X	Filler
11	A4	Elapsed time
12	5X	Filler
13	I5	Ground time
14	A1	Departure WAC, 1st digit
15	A2	Departure WAC, rest
16	A1	Filler
17	A3	Departure airport alpha code
18	I4	Departure time, local
19	I4	Departure time, GMT
20	I1	Arrival WAC, 1st digit
21	A2	Arrival WAC, rest
22	A1	Filler
23	A3	Arrival airport alpha code

FIGURE 2.5

Field	Format	Description
24	I4	Arrival time, local
25	A1	Flag code
26	I4	Arrival time, GMT
27	A1	Equipment type
28	A2	Carrier
29	A5	Flight number
30	A5	Class of service
31	7I1	Days of departure, local time
38	A1	Suppress code
39	A1	ATIS code
40	A4	Effective date
41	A4	Discontinued date
42	I6	Leave latitude (secs.+324K)
43	A5	Leave longitude (10 secs
44	A1	Leave longitude sign
45	I6	Arrive latitude
46	A5	Arrive longitude
47	A1	Arrive longitude sign
48	7I1	Days of arrival, local time
55	I1	Arrival date minus departure date

FIGURE 2.5 (continued)

3. Process:

a. Insure PT49 is mounted.

b. Submit the batch control file, FLT.CTL, which:

1. Executes USFLT.FOR creating USFLT.DAT.
2. Executes UEFLT.FOR creating UEFLT.DAT.
3. Executes EUFLT.FOR creating EUFLT.DAT.
4. Copies USFLT.DAT to FLT.DAT.
5. Executes PRDIS.FOR creating PR.DAT and DIS.DAT.
6. Copies PR.DAT to USPR.DAT.
7. Copies DIS.DAT to USDIS.DAT.
8. Copies UEFLT.DAT to FLT.DAT.
9. Executes PRDIS.FOR creating new PR.DAT and DIS.DAT.
10. Copies PR.DAT to UEPR.DAT.
11. Copies DIS.DAT to UEDIS.DAT.
12. Copies EUFLT.DAT to FLT.DAT.
13. Executes PRDIS.FOR creating new PR.DAT and DIS.DAT.
14. Copies PR.DAT to EUPR.DAT.
15. Copies DIS.DAT to EUDIS.DAT.
16. Sorts EUPR.DAT by: arrival city, departure city, (arrival date - departure date), departure time (descending), on to DEPEU.DAT.
17. Sorts EUPR.DAT by: departure city, arrival city, (arrival date - departure date), departure time, arrival time, onto ARREU.DAT.
18. Sorts UEPR.DAT as is DEUPP.DAT onto DEPUE.DAT.
19. Sorts UEPR.DAT as is ARREU.DAT onto ARREU.DAT.
20. Sorts USPR.DAT as is DEPEU.DAT onto DEPUS.DAT.
21. Sorts USPR.DAT as is ARREU.DAT onto ARRUS.DAT.

Note that USDIS.DAT and EUDIS.DAT are not used. Therefore the control file should be modified so that steps 2 and 13 are replaced with deletes. See Figure 2.6.

USFLT, EUFLT, and UEFLT.FOR select from OAGCOP.DAT those flights whose combined segments link the selected gateway cities and assemble the segments for those linkages into flight records. Segments separated by a ground time of over six hours may not be linked. At Dulles airport only, departure time for a departing segment may precede the landing time of the arriving flight segment by up to 30 minutes to account for mobile lounges.

PRDIS.FOR reads the above flight files and outputs those flights crossing the Atlantic to UEDIS.DAT, the "International Flight File". It also creates files giving flight data on every variant of every flight linking each pair of the set of selected gateway airports.

4. Programs:

   a. USFLT.FOR: creates USFLT.DAT.

   b. EUFLT.FOR: creates EUFLT.DAT.

   c. UEFLT.FOR: creates UEFLT.DAT.

   d. PRDIS.FOR: creates PR.DAT and DIS.DAT.

5. Separately Compiled Subroutines:

   a. CONSEG.FOR

   b. READLL.FOR

   c. REUDLL.FOR

   d. JETAPT.FOR

   e. JEUAPT.FOR

Record Format for:

USPR.DAT, UEPR.DAT, EUPR.DAT,

DEPUS.DAT, DEPUE.DAT, DEPEU.DAT,

ARRUS.DAT, ARRUE.DAT, ARREU.DAT,

USPEUP.DAT, EUPUSP.DAT,

OPUOPU.DAT, OPEOPE.DAT,

PDPUPDU.DAT, PDEPDE.DAT

Field	Format	Description
1	1X	Filler
2	A2	Carrier
3	A5	Flight number
4	A1	Variant number
5	I2	City pair number
6	I2	No. days operation per week
7	I2	No. intermediate stops
8	I6	Departure city/airport numeric
9	I6	Arrival city/airport numeric
10	A3	Equipment code for segment with largest elapsed time
11	I5	Sum of segment airborne times
12	I5	Sum of segment distances
13	I5	Sum of intersegment ground times
14	A4	Departure WAC
15	A3	Departure airport alpha
16	I4	Departure time, local
17	A4	Departure time, GMT
18	7A1	Days of departure, local time

FIGURE 2.6

Field	Format	Description
25	A4	Arrival WAC
26	A3	Arroval airport alpha
27	A4	Arrival time, local
28	A4	Arrival time, GMT
29	7A1	Days of arrival, local time
36	I1	Largest value of flag code
37	A1	Equipment type for segment with largest elapsed time
38	A5	Class of service for segment with largest elapsed time
39	I1	Sum of suppress codes
40	A1	ATIS code for segment with largest elapsed time
41	A4	Effective date of last segment with effective date
42	A4	Discontinued date of last segment with discontinued date
43	I1	Arrival minus departure date

FIGURE 2.6 (Continued)

f. NONUM.FOR

g. IADATE.FOR

h. FLTLEG.FOR

i. LEGDAY.FOR

Step 3: Manually Edit the International Flight File

1. Input:

UEDIS.DAT

2. Output:

UEDIS.DAT

3. Process:

a. Mount PT49.

b. Using the system editor, delete all UEDIS.DAT flight segments with both airports out of the US/EUR region. Note: these should have been eliminated automatically earlier, but bugs remain in the programs.

Step 4: Select Connection and Flight Files

1. Inputs:

a. HFILE.DAT: an ASCII file listing every (water leg) - (EUR directional destination airport) pair such that total CY77 traffic is greater than or equal to 30. Note: Since this is a 10% sample, the recorded value is three. Water legs are defined by the two port airports and the transatlantic carrier. This file is obtained during the traffic file production process. See Figures 1.1 and 2.7. It serves as a reference table of permitted flight interconnection for later processes.

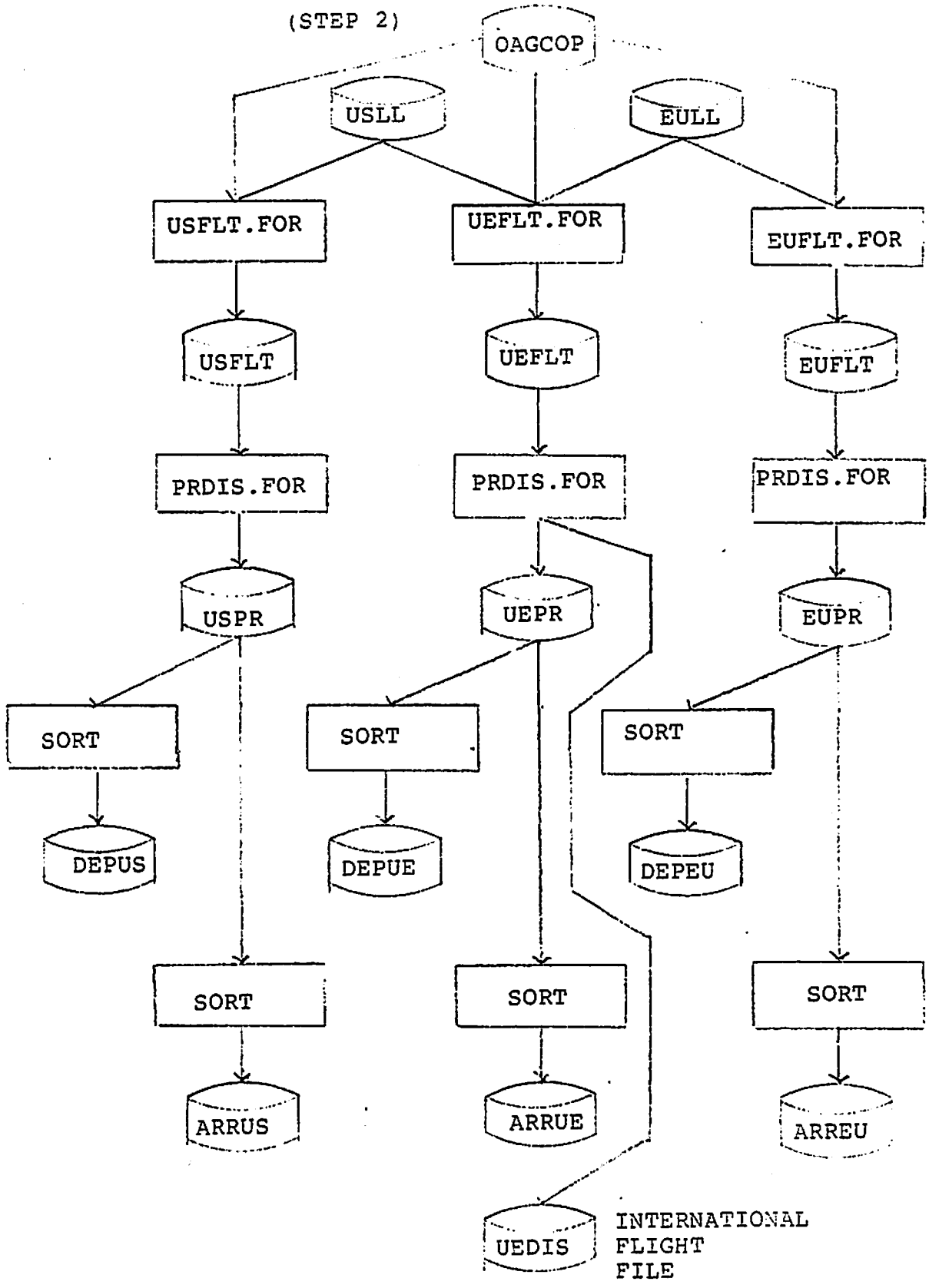
b. HFILE2.DAT: an ASCII file listing selected (water leg) - (US destination airport) connections.

Record Format for UEDIS.DAT

Field	Format	Description
1	1X	Filler
2	A2	Carrier
3	A5	Flight number
4	A1	Flight variant number
5	I2	No. days/week operation
6	I2	No. of flight segments
(the following fields occur once per flight segment)		
7	1X	Filler
8	I6	Departure city/airport numeric
9	I6	Arrival city/airport numeric
10	A3	Equipment code
11	I5	Segment elapsed airborne time
12	I5	segment distance in statute miles
13	I5	Ground time before next segment departure
14	A3	Departure airport alpha
15	A3	Arrival airport alpha

FIGURE 2.7

FIGURE 2.8 PROCESS CREATING THE FILES OF SELECTED FLIGHTS



Record Format for HFILE.DAT, HFILE2.DAT

Field	Format	Description
1	1X	Filler
2	I5	US port city numeric
3	I2	US port city numeric
4	I4	US port city airport numeric
5	1X	Filler
6	A2	Transatlantic carrier
7	I6	EUR port city numeric
8	I2	EUR port city airport
9	I4	EUR port city WAC numeric
10	3X	Filler
11	I6	No. of US origin cities for this water leg
12	I6	US gateway city numeric
13	I2	US gateway airport numeric
14	I4	US gateway WAC

Notes:

1. This data is bidirectional, i.e., gateway cities are either directional origins or destinations.

2. For HFILE2.DAT, fields 12-14 refer to EUR gateway cities.

FIGURE 2.9

FIGURE 2.10 PROCESS FURTHER SELECTING TRANSATLANTIC AND CONNECTING FLIGHTS

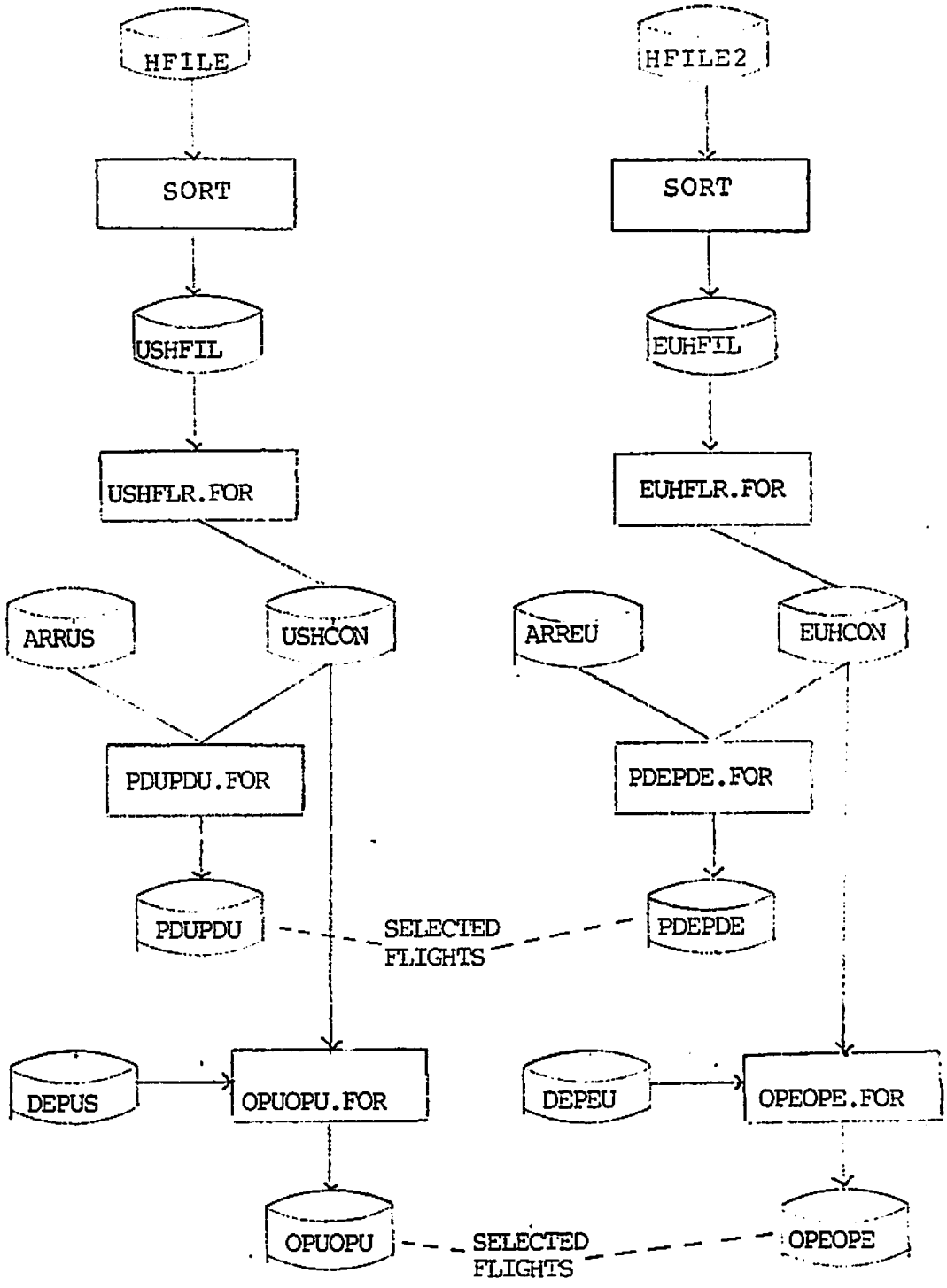
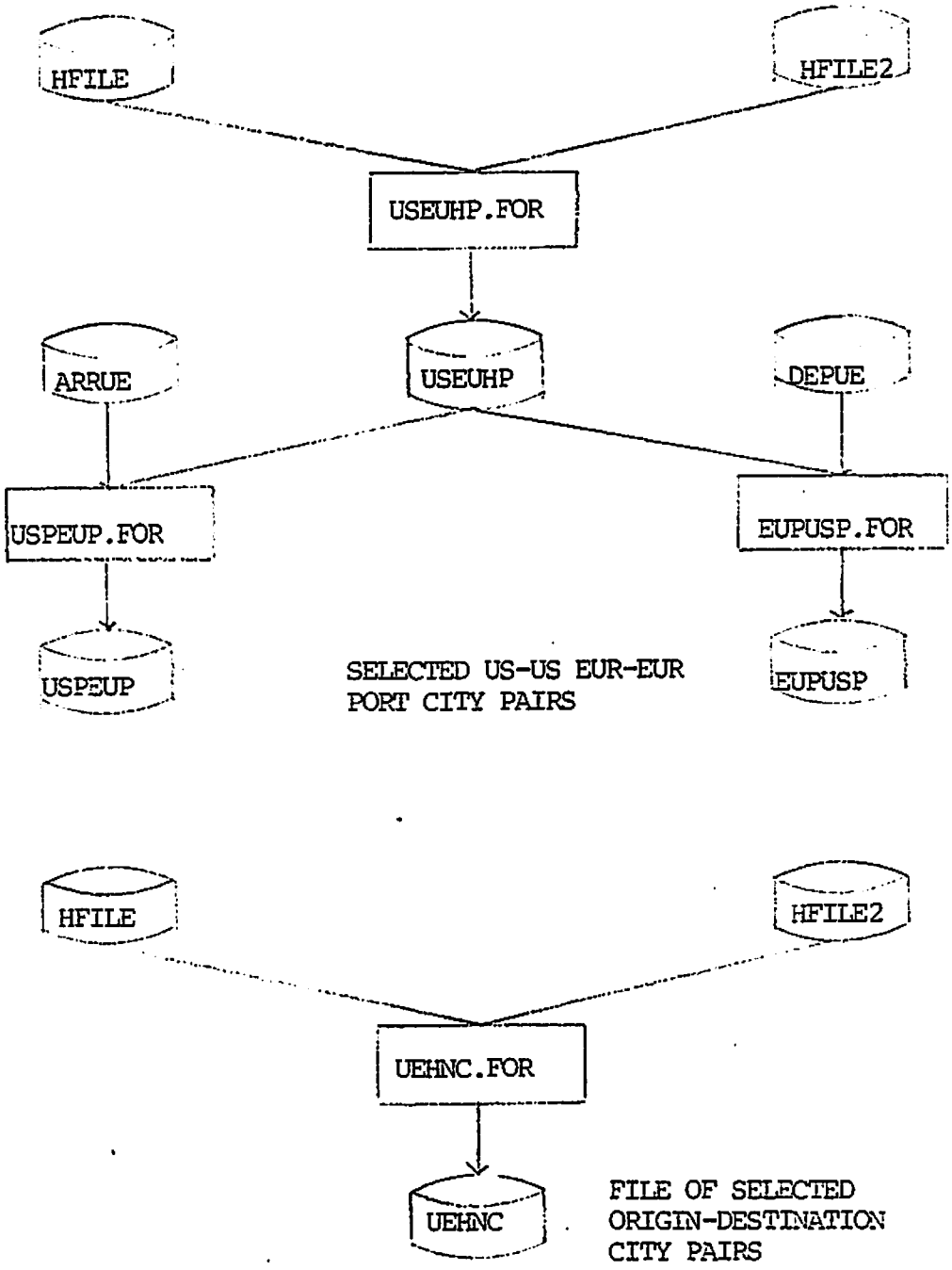


FIGURE 2.10 Continued



c. Flight files from Step 2:

1. ARRUS.DAT
2. ARREU.DAT
3. DEPUS.DAT
4. DEPEU.DAT
5. ARRUE.DAT
6. DEPUE.DAT

2. Outputs:

a. USHFIL.DAT: HFILE.DAT sorted by US port city, US origin city.

b. EUHFIL.DAT: HFILE2.DAT sorted by EUR port city.

c. USHCON.DAT: HFILE.DAT merged over airports, i.e., just by cities. HFILE.DAT format.

d. EUCON.DAT: HFILE2.DAT merged over airports.

e. PDUPDU.DAT: an ASCII file which is the subset of ARRUS.DAT such that all records have port and destination cities also listed in USHCON.DAT.

f. PDEPDE.DAT: a similar subset of ARREU.DAT for records whose port and destination cities are in EUHCON.DAT.

g. OPUOPU.DAT: a similar subset of DEPUS.DAT for records whose port and origin cities are in USHCON.DAT.

h. OPEOPE.DAT: a similar subset of DEPEU.DAT for records whose port and origin cities are in EUHCON.DAT

i. USEUHP.DAT: an ASCII file of unique (US port city) (EUR port city) pairs found in HFILE.DAT and HFILE2.DAT.

j. USPEUP.DAT: an ASCII file which is a subset of ARRUE.DAT such that all US departure and EUR arrival port cities are in USEUHP.DAT.

k. EUPUSP.DAT: a similar subset of DEPUE.DAT such that EUR departure and US arrival port cities are in USEUHP.DAT.

l. UEHNC.DAT: an ASCII file of one HFILE record for each US origin/destination city for each US/EUR port city pair and one HFILE2 record for each EUR origin/destination city for each EUR/US port city pair.

3. Process:

a. Mount PT49.

b. Submit the batch control file HFILES.CTL:

1. Sorts HFILE.DAT into USHFIL.DAT by: US port city numeric, US origin/destination numeric.
2. Sorts HFILE2.DAT into EUHFIL.DAT by: EUR port city numeric, EUR origin/destination numeric.
3. Executes USHFLR.FOR reading USHFIL.DAT and creating USHCON.DAT.
4. Executes OPUOPU.FOR reading DEPUS.DAT and USHCON.DAT and creating OPUOPU.DAT.
5. Executes PDUPDU.FOR reading ARRUS.DAT and USHCON.DAT and creating PDUPDU.DAT.
6. Executes EUHFLR.FOR reading EUHFIL.DAT and creating EUHCON.DAT.
7. Executes OPEOPE.FOR reading DEPEU.DAT and EUHCON.DAT and creating OPEOPE.DAT.
8. Executes PDEPDE.FOR reading ARREU.DAT and EUHCON.DAT and creating PDEPDE.DAT.
9. Executes USEUHP.FOR reading HFILE.DAT and HFILE2.DAT and creating USEUHP.DAT.
10. Executes EUPUSP.FOR reading DEPUE.DAT and USEUHP.DAT and creating EUPUSP.DAT.

11. Copies ARRUE.DAT into PT49: SUSEU.DAT. Note: This is probably an error as SUSEU.DAT is not used later.

12. Executes USPEUP.FOR reading ARRUE.DAT and USEUHP.DAT and creating USPEUP.DAT.

13. Executes UEHNC.FOR reading HFILE and HFILE2 and creating UEHNC.DAT.

This step takes the set of scheduled flight records created in Step 2 and the set of origin/destination/water leg connections selected during the traffic file creation process and creates corresponding flight and connection files representing their (set theory) intersection.

4. Programs:

1. USHFLR.FOR: creates the intermediate file USHCON.DAT.
2. OPUOPU.FOR: creates the selected flight file OPUOPU.DAT
3. PDUPDU.FOR: creates the selected-flight file PDUPDU.DAT.
4. EUHFLR.FOR: creates the intermediate file EUHCON.DAT
5. OPEOPE.FOR: creates the selected-flight file OPEOPE.DAT.
6. PDEPDE.FOR: creates the selected-flight file PDEPDE.DAT.
7. USEUHP.FOR: creates the intermediate file USEUHP.DAT.
8. EUPUSP.FOR: creates the selected-flight file EUPUSP.DAT.
9. USPEUP.FOR: creates the selected-flight file USPEUP.DAT.
10. UEHNC.FOR: creates the selected-connection file UEHNC.DAT.

There are no separately compiled subroutines for this step.

<u>Step 5:</u> Creates the Base Case QSI File

1. Inputs:

a. CONNEC.DAT: a manually created ASCII file of connection times for US port cities.

b. CONECT.DAT: a similar ASCII file of connection times for EUR port cities.

c. files from Step 4:

1. UEHNC.DAT: lists selected connections.
2. PDEPDE.DAT: lists selected flights.
3. PDUPDU.DAT: lists selected flights.
4. OPEOPE.DAT: lists selected flights.
5. OPUOPU.DAT: lists selected flights.
6. EUUS.DAT: lists selected flights, copied from UEPEUP.DAT created in Step 4.
7. USEU.DAT: lists selected flights, copied from EUPUEP.DAT created in Step 4.

d. UEDIS.DAT: from Step 2, gives full flight data, copied into UEDSC.DAT.

e. PRTNM.DAT: a manually created ASCII file of US and EUR origin/destination cities and their numeric codes.

2. Outputs:

a. SCENAR.DAT: an ASCII file containing selected "best", or access maximizing, origin/port or port/destination connecting flights.

b. QSI1.DAT: the Base Case QSI file, sorted information SCODFL.DAT: an ASCII file of selected "greatest access" O/WL/D flight connections.

c. WATER.LEG: a Binary Image file listing international flights and QSI's by water leg.

3. Process:

a. Mount PT49.

b. Submit batch control file BSCOD.CTL which:

1. Copies UEPEUP.DAT into USEU.DAT.
2. Copies EUPUEP.DAT into EUUS.DAT.
3. Executes SCENAR.FOR creating the "best" i.e., least time, origin-waterleg-destination set for each water leg files.
4. Copies UEDIS.DAT into UEDSC.DAT
5. Executes SCODFIL.FOR.
6. Copies WATERS.LEG into WATER.LEG.
7. Sorts SCODRT.DAT into QSI1.DAT by origin, destination, US port, EUR port, water leg carrier.

Program SCENAR.FOR reads the water leg files, USEU and EUUS, and the file of possible connection cities for each water leg, EUHNC, and forms an intermediate reference file. It then examines the files of selected connecting flights, PDEPDE, PDUPDU, OPEOPE, OPUOPE, using this reference file and two files of connect times, CONNEC, CONECT, to select the first valid origin-water leg-destination set of flight segments. Because of the time sort order of the four files of connecting flight, this first valid O/WL/D flight set requires the least total origin/destination travel time. However, the connecting flights are then searched again with on line connections given a two hour advantage. The output file of chosen O/WL/D connections is sorted by water leg carrier and flight number.

Program SCODFL.FOR reads the previous output and calculates QSI's. It outputs flight connection and QSI data in Binary Image format to WATER.LEG for later inclusion in traffic

processing. (Binary Image format is necessitated by later processing by PASCAL programs.) The data is also output to SCODRT.DAT and sorted into QSI1.DAT "Base Case QSI File".

4. Programs:

   a. SCENAR.FOR finds the adjusted least time set of flights connecting the selected origins and destinations through the given set of water legs.

   b. SCODFL.FOR: calculates the QSI's for these origin-water leg-destination paths. Neither program utilizes separately compiled subroutines.

Record Format for CONNEC.DAT, CONECT.DAT

Field	Format	Description
1	A3	Airport alpha
2	I5	City numeric
3	I1	Airport numeric
4	I4	Off line transatlantic-to-connection flag
5	I4	On line transatlantic-to-connection flag
6	I4	Off line connection-to-transatlantic flag
7	I4	Off line connection-to-transatlantic flag

(Both files contain records of two different formats. All 7-field "airport connection" records are followed by all 8-field "interairport connection" records. The last airport connection record has a field 6 value of 0.)

1	AB	First airport alpha
2	I5	First city numeric
3	I1	First airport numeric
4	A3	Second airport alpha
5	I5	Second city numeric
6	I1	Second airport numeric
7	I4	Transatlantic-to-connection flag
8	I4	Connection-to-transatlantic flag

FIGURE 2.11

Record Format for PRTNM.DAT

Field	Format	Description
1	5X	Filler
2	I5	US city numeric
3	5X	Filler
4	I5	EUR city numeric

FIGURE 2.12

Record Format for SCENAR.DAT

Field	Format	Description
1	1X	Filler
2	A2	International carrier
3	A5	International flight no.
4	A1	International flight variant
5	I5	Port city numeric
6	I1	Port city airport numeric
7	I5	Connection city numeric
8	I1	Connection city airport numeric
9	I5	Connection of airborne times
10	I5	Connection of segment distances
11	I5	Connection of intersegment ground times
12	I1	Multi-airport flag (1=different)
13	A3	Connection equipment code for segment with largest elapsed time
14	I2	Connection-number of intermediate stops
15	A2	Connection carrier
16	A3	Departure airport alpha
17	A3	Arrival airport alpha
18	I5	Ground time between international and connection flight
19	I2	Connection-sum of suppress codes

FIGURE 2.13

Record Format for SCODRT.DAT, BSCnD.DAT

Field	Format	Description
1	1X	Filler
2	I5	Origin city numeric
3	I1	Origin city airport numeric
4	I5	Destination city numeric
5	I1	Destination airport numeric
6	I5	1st port city numeric
7	I1	1st port city airport numeric
8	I5	2nd port city numeric
9	I1	2nd port city airport numeric
10	I5	Sum of segment distances
11	I2	Frequency/week
12	I1	1st connection carrier code *
13	I1	International carrier code *
14	I1	2nd connection carrier code *
15	I6	Flight and variant no.
16	A2	1st connection carrier
17	A2	International carrier
18	A2	2nd connection carrier
19	I1	Directional code **
20	F10-4	QSIM
21	F10.4	QSIS

* 0 = none 1 = US 2 = EUR

** 1 = US origin 2 = EUR origin

FIGURE 2.14

Record Format for WATER.LEG

Field	Type	Description
1	I	Origin city numeric
2	I	Origin city airport numeric
3	I	Destination city numeric
4	I	Destination city airport numeric
5	I	Sum of segment distances
6	A	Equipment code for segment with largest elapsed time
7	I	Frequency/week
8	I	1st conneciton carrier code *
9	I	International carrier code *
10	I	2nd connection carrier code *
11	I	Flight number
12	A	International carrier
13	I	Origin code **
14	I	Travel time in min., 1st segment
15	I	Travel time in min., 2nd segment
16	I	Travel time in min., 3rd segment
17	I	Travel time in min., 4th segment
18	I	Travel time in min., 5th segment
19	I	Travel time in min., 6th segment
20	I	Travel time in min., 7th segment
21	I	Travel time in min., 8th segment
22	I	Travel time in min., 9th segment
23	I	Ground time in min., 2nd city
24	I	Ground time in min., 3rd city
25	I	Ground time in min., 4th city

FIGURE 2.15

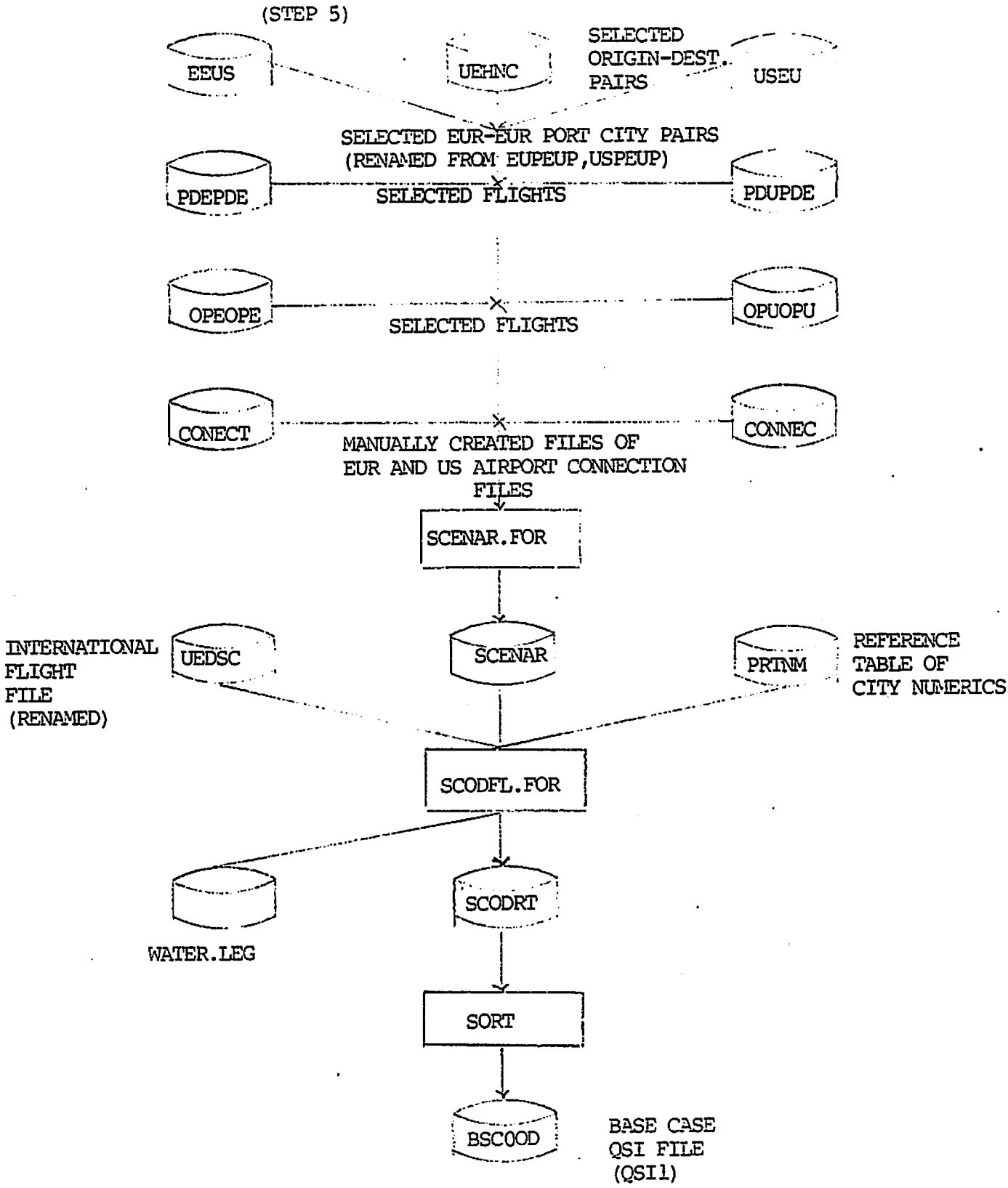
Field	Type	Description
26	I	Ground time in min., 5th city
27	I	Ground time in min., 6th city
28	I	Ground time in min., 7th city
29	I	Ground time in min., 8th city
30	I	Ground time in min., 9th city
31	F	QSIM

* 0 = none 1 = US 2 = EUR

** 1 = US origin 2 = EUR origin

FIGURE 2.15 (Continued)

FIGURE 2.16 PROCESS CREATING QSI AND WATER.LEG FILES



3.0 ALTERNATE SCENARIOS

Step 1: Change the Pattern of Gateway Interconnections (optional)

Just repeat Step 3 of the Base Case QSI creation process.

Step 2: Select Connection and Flight Files (required only if Step 1 performed)

Just repeat Step 4 of the Base Case QSI creation process.

Step 3: Delete Records of Flights to be Cancelled or Changed

1. Inputs:
   a. UEDIS.DAT: from Base Case or Step 2.
   b. USPEUP.DAT : from Base Case or Step 2.
   c. EUPUSP.DAT: from Base Case or Step 2.
2. Outputs:
   a. UEnDIS.DAT: Alternate Scenario International Flight file.
   b. USnEU.DAT: Alternate Scenario connection file.
   c. EUnUS.DAT: Alternate Scenario connection file.
3. Process:
   a. Mount PT49.
   b. Using PIP:
      1. COPY UEnDIS.DAT = UEDIS.DAT.
      2. COPY USnDIS.DAT = USDIS.DAT.
      3. COPY EUnDIS.DAT = EUDIS.DAT.
   c. For all flights being deleted or changed, use a system editor to:
      1. In UEnDIS.DAT, set the number of days of operation, NFR, to zero. See Fig. 2.5.

2. in USnEU.DAT or EUnUS.DAT, as indicated by the flight route, set the days of departure, DI(5) and DI(6) and days of arrival, AI(5) and AI(6), to all zeros. See Fig. 2.4.

Note that input files and output QSI records are not changed, per se, but are either deleted or deleted and replaced with similar records of new values. Note also that the small "n" in the Alternate Scenario file names represents a digit 1, 2, 3... indicating which Alternate Scenario is being prepared.

Step 4: Add Records of Flights to be Changed

1. Inputs:
   a. UEnDIS.DAT: from Step 3.
   b. USnEU.DAT: from Step 3.
   c. EUnUS.DAT: from Step 3.
2. Outputs:
   a. UEnDIS.DAT
   b. USnEU.DAT
   c. EUnUS.DAT
3. Process:
   a. Mount PT49.
   b. Using a system editor:
      1. Add to UEnDIS.DAT flight records of an identical format but with new flight schedule data.
      2. Add to USnEU.DAT flight records corresponding to each added to UEnDIS.DAT, for each transatlantic pair of cities from the UEnDIS.DAT itineary.

3. Similarly add corresponding records to EUnUS.DAT. Note that days of operation for added flight records should not be zero for all seven days of operation.

Step 5: Create the Alternate Scenario QSI and WATERn.LEG Files

1. Inputs:

   a. EUnUS.DAT: from Step 4 above. Lists selected city pairs.

   b. USnEU.DAT: from Step 4 above. Lists selected city pairs.

   c. UEnDIS.DAT: from Step 4 above. Lists selected full flight data.

   d. CONNEC.DAT: manually created for Step 5 of Base Case process. Lists connection times at US port cities.

   e. CONECT.DAT: similar file for EUR cities.

   f. PRTNM.DAT: manually created for Step 5 of Base Case process. Reference table of US and EUR origin/destination city numerics.

   g. Files from Step 4 of Base Case process. List selected connecting flights:

      1. PDEPDE.DAT
      2. PDUPDU.DAT
      3. OPEOPE.DAT
      4. OPUOPU.DAT

   h. UEHNC.DAT: from Step 4 of Base Case process. Lists selected connections.

2. Outputs:

   a. QSI2n.DAT: QSI records which change under the Alternate Scenario. Input to model.

b. WATERn.LEG: similar file for Financial Model pre-processing.

3. Process:

a. Mount PT49.

b. Submit batch control file BSCnOD.CTL which:

1. Copies USnEU.DAT into USEU.DAT.

2. Copies EUnUS.DAT into EUUS.DAT.

3. Executes SCENAR.FOR which selects the least-time set of O/D interconnecting flights and outputs them to SCENAR.DAT.

4. Copies UEnDIS.DAT into UEDSC.DAT.

5. Executes SCODFL.FOR which computes QSI's for the routes selected.

6. Sort SCODRT.DAT into QSI2n.DAT by: origin, destination, transatlantic carrier, and flight number.

This step is identical to Step 5 of the Base Case process except that the only QSI2n.DAT and WATERn.LEG records produced are those associated with flight schedule changes.

4. Programs:

a. SCENAR.FOR: finds the least-time flight connections for the changed schedule and given cities.

b. SCODRT.FOR: computers the QSI's for those changed flights and outputs the results to QSI2n.DAT and WATERn.LEG. No separately compiled subroutines are involved.

APPENDIX A

Program Listings:

OAGCOP.FOR, USFLT.FOR, UEFLT.FOR, EUFLT.FOR, CONSEG.FOR, READLL.FOR, REUDLL.FOR, JETAPT.FOR, JEUAPT.FOR, IADATE.FOR, FLTLEG.FOR, LEGDAY.FOR, PRDIS.FOR, USHFLR.FOR, EUHFLR.FOR, PDUPDU.FOR, PDEPDE.FOR, OPUOPU.FOR, OPEOPE.FOR, USEUHP.FOR, USPEUP.FOR, EUPUSP.FOR, UEHNC.FOR, SCENAR.FOR, SCODFL.FOR

OAGCOP.FOR

```
      PROGRAM oagcop
C     PROGRAM READS OAG TAPE-IDENTIFIES US/[illegible]
C       LINKS-COPIES ONTO oagcop.DAT ON PT49
      IMPLICIT INTEGER (A-Z)
      DIMENSION DUM1(16),DUM2(29),DUM3(58)
      DATA NREC,NWL/0,0/
      OPEN(UNIT=16,ACCESS='SEQIN',BLOCKSIZE=2200)
      OPEN(UNIT=21,FILE='oagcop.DAT'
     1,ACCESS='SEQOUT',DEVICE='PT49')
c     skip RECORD 16
c     skip RECORD 16
5     READ(16,10,END=999) DWAC,DUM1,AWAC,DUM2,FARE,DUM3
10    FORMAT(A1,16A1,A1,29A1,A5,58A1)
      NREC=NREC+1
      IF(FARE.EQ.' ')GO TO 5
c     IF(DWAC.EQ.'0'.AND.AWAC.EQ.'4'.OR.
c     1DWAC.EQ.'4'.AND.AWAC.EQ.'0')GO TO 15
c     GO TO 5
15    NWL=NWL+1
      WRITE(21,10) DWAC,DUM1,AWAC,DUM2,FARE,DUM3
c17   FORMAT(1X,A1,16A1,A1,29A1,A5,58A1)
      GO TO 5
999   TYPE 20,NREC,NWL
20    FORMAT(1X,I6,' TOTAL RECORDS PROCESSED',5X,I6,
     1' passenger LINKS')
      CLOSE(UNIT=16)
      CLOSE(UNIT=21)
      STOP
      END
```

USFLT.FOR

```
c          PROGRAM READS OAG TAPE-IDENTIFIES US/EUR
c            FINKS-COPIES ONTO usflt.DAT
           IMPLICIT INTEGER (A-Z)
           parAmeter maxleg=29,lrec=37
           DIMENSION DUM3(11)
           common/rec/dwac,drac,dum,dapt,dum1,dtl,dtg,awac,arac,aum,aapt
           8,dum2,atl
           9,fc,atg,etec,car
           1,fn,fare,freq(7),dum3
           dimension rc(lrec),flt(lrec,maxleg)
           equivalence(rc(1),dwac),(flt(17,1),ocar),(flt(18,1),ofn)
           DATA NREC,NWL/0,0/,leg,bad/2*0/,eof/0/,nleg/0/
           lrec2=lrec-2
           OPEN(UNIT=16,ACCESS='SEQIN',file='OAGCOP.dat',device='PT49')
           OPEN(UNIT=21,FILE='usflt.DAT',device='PT49'
           1,ACCESS='SEQOUT')
c          open(unit=22,file='aptpr.dat',access='seqout')
           open(unit=23,file='USbad.dat',access='seqout',DEVICE='PT49')
c          rewind 16
c          skip file 16
           READ(16,10,END=999)RCD
           READ(16,10,END=999)RCD
5        READ(16,10,END=999)rc
10         FORMAT(A1,a2,a1,a3,a2,2a4,A1,a2,a1,a3,a2,a4,a1,a4,a5,a2,a5,A5,
           77i1,a2,a4,2A4,2(I6,A5,A1),A5)
9        FORMAT(A1,a2,a1,a3,a2,2a4,A1,a2,a1,a3,a2,a4,a1,a4,a5,a2,a5,A5,
           77i1,a2,a4,2A4,2(I6,A5,A1),A5)
           NREC=NREC+1
           if(mod(nrec,500).eq.1)type 9,rc
           IF(FARE.EQ.' ')GO TO 5
           IF(DUM3(3).NE.' ')GO TO 500
c          if(dapt.gt.aapt)goto 200
c          apt1=dapt
c          apt2=aapt
c          goto210
c200       apt1=aapt
c          apt2=dapt
c210       write(22,3)apt1,apt2
c3         format(2a3)
           if(leg.eq.0)go to 100
           if(car-ocar)900,30,50
30         if(fn-ofn)900,40,50
40         if(us.gt.0)go to 60
           call readll(dapt,ndapt,dlat,dlon)
           if(ndapt.eq.0.or.ndapt.eq.eur)go to 60
           us=ndapt
60         if (eur.gt.0)go to 70
           call readll(aapt,naapt,alat,alon)
           if(naapt.eq.0.or.naapt.eq.us)go to 70
           eur=naapt
70         if(leg.ge.maxleg)go to 900
           leg=leg+1
           do 80 i=1,lrec
80         flt(i,leg)=rc(i)
           go to 5
50         if(us.eq.0.or.eur.eq.0)go to 100
           nwl=nwl+1
           call conseg(flt,leg,1)
           nleg=nleg+leg
100        if(eof.eq.1)go to 1001
           leg=0
```

```
      us=0
      eur=0
      go to 40
500   WRITE(23,9)RC
      BAD=BAD+1
      GO TO 5
900   write(23,9)(flt(i,leg),i=1,lrec)
      write(23,9)rc
      bad=bad+1
      go to 50
999   eof=1
      go to 50
1001  TYPE 20,NREC,NWL,nleg,bad
20    FORMAT(1X,I6,' TOTAL RECORDS PROCESSED',5X,I6,
     1' US/EUR LINKS',5x,i6,' Legs',5x,i6,' Bad Records')
      CLOSE(UNIT=16)
      CLOSE(UNIT=21)
c     close(unit=22)
      if(bad.eq.0)close(unit=23,dispose='delete')
      STOP
      END
```

UEFLT.FOR

```
C         PROGRAM READS ORG TAPE-IDENTIFIES UE/EUR
C           LINKS-COPIES ONTO UEflt.DAT
          IMPLICIT INTEGER (A-Z)
          parAmeter maxleg=99,lrec=37
          DIMENSION DUM3(11)
          common/rec/dwac,drac,dum,iapt,dum1,dtl,dtg,awac,arac,aum,aapt
          8,dum2,atl
          9,fc,atg,etec,car
          1,fn,fare,freq(7),dum3
          dimension rc(lrec),flt(lrec,maxleg)
          equivalence(rc(1),dwac),(flt(17,1),ocar),(flt(18,1),ofn)
          DATA NREC,NWL/0,0/,leg,bad/2*0/,eof/0/,nleg/0/
          lrec2=lrec-2
          OPEN(UNIT=16,ACCESS='SEQIN',file='OAGCOP.dat',device='PT49')
          OPEN(UNIT=21,FILE='UEflt.DAT',device='PT49'
          1,ACCESS='SEQOUT')
C         open(unit=22,file='aptpr.dat',access='seqout')
          open(unit=23,file='UEbad.dat',access='seqout',DEVICE='PT49')
C         rewind 16
C         skip file 16
          READ(16,10,END=999)RCD
          READ(16,10,END=999)RCD
5       READ(16,10,END=999)rc
10        FORMAT(A1,a2,a1,a3,a2,2a4,A1,a2,a1,a3,a2,a4,a1,a4,a5,a2,a5,A5,
          77i1,a2,a4,2A4,2(I6,A5,A1),A5)
9       FORMAT(A1,a2,a1,a3,a2,2a4,A1,a2,a1,a3,a2,a4,a1,a4,a5,a2,a5,A5,
          77i1,a2,a4,2A4,2(I6,A5,A1),A5)
          NREC=NREC+1
          if(mod(nrec,500).eq.1)type 9,rc
          IF(FARE.EQ.' ')GO TO 5
          IF(DUM3(3).NE.' ')GO TO 500
C         if(dapt.gt.aapt)goto 200
C         apt1=dapt
C         apt2=aapt
C         goto210
C200      apt1=aapt
C         apt2=dapt
C210      write(22,3)apt1,apt2
C3        format(2a3)
          if(leg.eq.0)go to 100
          if(car-ocar)900,30,50
30        if(fn-ofn)900,40,50
40        if(us.gt.0)go to 60
          call readll(dapt,ndapt,dlat,dlon)
          us=ndapt
          IF(US.GT.0)GO TO 60
          call readll(aapt,naapt,alat,alon)
          US=NAAPT
60        if (eur.gt.0)go to 70
          call reUdll(aapt,naapt,alat,alon)
          eur=naapt
          IF(EUR.GT.0)GO TO 70
          call reUdll(dapt,ndapt,dlat,dlon)
          EUR=NDAPT
70        if(leg.ge.maxleg)go to 900
          leg=leg+1
          do 80 i=1,lrec
80        flt(i,leg)=rc(i)
          go to 5
50        if(us.eq.0.or.eur.eq.0)go to 100
          nwl=nwl+1
```

```
10200          call conseq(flt,leg,3)
10300          nleg=nleg+leg
10600    100   if(eof.eq.1)go to 1001
10700          leg=0
10800          us=0
10900          eur=0
11000          go to 40
11010    500   WRITE(23,9)RC
11020          BAD=BAD+1
11030          GO TO 5
11100    900   write(23,9)(flt(i,leg),i=1,lrec)
11200          write(23,9)rc
11300          bad=bad+1
11400          go to 50
11500    999   eof=1
11600          go to 50
11700    1001  TYPE 20,NREC,NWI,nleg,bad
11800    20    FORMAT(1X,I6,' TOTAL RECORDS PROCESSED',5X,I6,
11900          1' UE/EUR LINKS',5x,i6,' Legs',5x,i6,' Bad Records')
12000          CLOSE(UNIT=16)
12100          CLOSE(UNIT=21)
12200    c     close(unit=22)
12300          if(bad.eq.0)close(unit=23,dispose='delete')
12400          STOP
12500          END
```

EUFLT.FOR

```
C         PROGRAM READS OAG TAPE-IDENTIFIES US/EUR
C           LINKS-COPIES ONTO euflt.DAT
          IMPLICIT INTEGER (A-Z)
          parameter maxleg=99,lrec=37
          DIMENSION DUM3(11)
          common/rec/dwac,drac,dum,dapt,dum1,dtl,dtg,awac,arac,aum,aapt
          8,dum2,atl
          9,fc,atg,etec,car
          1,fn,fare,freq(7),dum3
          dimension rc(lrec),flt(lrec,maxleg)
          equivalence(rc(1),dwac),(flt(17,1),ocar),(flt(18,1),ofn)
          DATA NREC,NWL/0,0/,leg,bad/2*0/,eof/0/,nleg/0/
          lrec2=lrec-2
          OPEN(UNIT=16,ACCESS='SEQIN',file='oagcop.dat',device='PT49')
          OPEN(UNIT=21,FILE='euflt.DAT',device='PT49'
          1,ACCESS='SEQOUT')
C         open(unit=22,file='aptpr.dat',access='seqout')
          open(unit=23,file='oagbad.dat',access='seqout')
C         rewind 16
C         skip file 16
          reAD(16,10,END=999)rcd
          reAD(16,10,END=999)rcd
5       READ(16,10,END=999)rc
10        FORMAT(A1,a2,a1,a3,a2,2a4,A1,a2,a1,a3,a2,a4,a1,a4,a5,a2,a5,A5,
          77i1,a2,a4,2A4,2(I6,A5,A1),a5)
9       FORMAT(A1,a2,a1,a3,a2,2a4,A1,a2,a1,a3,a2,a4,a1,a4,a5,a2,a5,A5,
          77i1,a2,a4,2A4,2(I6,A5,A1),a5)
          NREC=NREC+1
          if(mod(nrec,2000).eq.1)type 9,rc
          IF(FARE.EQ.' ')GO TO 5
          if(dum3(3).ne.' ')go to 5
C         if(dwac.lt.4.or.dwac.gt.6.or.awac.lt.4.or.awac.gt.6)goto 5
C         if(dapt.gt.aapt)goto 200
C         apt1=dapt
C         apt2=aapt
C         goto210
c200      apt1=aapt
C         apt2=dapt
c210      write(22,3)apt1,apt2
c3        format(2a3)
          if(leg.eq.0)go to 100
          if(car-ocar)900,30,50
30        if(fn-ofn)900,40,50
40        if(us.gt.0)go to 60
          call reudll(dapt,ndapt,dlat,dlon)
          if(ndapt.eq.0.or.ndapt.eq.eur)go to 60
          us=ndapt
60        if (eur.gt.0)go to 70
          call reudll(aapt,naapt,alat,alon)
          if(naapt.eq.0.or.naapt.eq.us)go to 70
          eur=naapt
70        if(leg.ge.maxleg)go to 900
          leg=leg+1
          do 80 i=1,lrec
80        flt(i,leg)=rc(i)
          go to 5
50        if(us.eq.0.or.eur.eq.0)go to 100
          nwl=nwl+1
          call conseg(flt,leg,2)
          nleg=nleg+leg
100       if(eof.eq.1)go to 1001
```

```
10700              leg=0
10800              us=0
10900              eur=0
11000              go to 40
11100    500       write(23,9)(flt(i,leg),i=1,lrec)
11200              write(23,9)rc
11300              bad=bad+1
11400              go to 50
11500    999       eof=1
11600              go to 50
11700    1001      TYPE 20,NREC,NWI,nleg,bad
11800    20        FORMAT(1X,I6,' TOTAL RECORDS PROCESSED',5X,I6,
11900              1' US/EUR LINKS',5x,i6,' Legs',5x,i6,' Bad Records')
12000              CLOSE(UNIT=16)
12100              CLOSE(UNIT=21)
12200    c         close(unit=22)
12300              if(bad.eq.0)close(unit=23,dispose='delete')
12400              STOP
12500              END
```

```
00100          subroutine consed(flt,leg,iue)
00300          IMPLICIT INTEGER (A-Z)
00400          parameter maxleg=99,lrec=37
00500          DIMENSION DUMB(11),afreq(7,maxleg),ida(maxleg),list(maxleg)
01000          dimension rc(lrec),rlt(lrec,maxleg),prev(maxleg),next(maxleg)
01200          dimension mdapt(maxleg),maapt(maxleg)
01300          lrec2=lrec-1
05200          do 110 ii=1,leg
05202          i=ii
05203          mdapt(i)=0
05204          maapt(i)=0
05207          if(iue.eq.2)go to 350
05210          call readll(flt(4,i),mdapt(i),dlt,dln)
05220          call readll(flt(11,i),maapt(i),alt,aln)
05221          if(iue.eq.1)go to 360
05222   350    if(mdapt(i).eq.0)call reudll(flt(4,i),mdapt(i),dlt,dln)
05223          if(maapt(i).eq.0)call reudll(flt(11,i),maapt(i),alt,aln)
05224   360    continue
05226          CALL SLON(FLT(1,I),FLT(2,I),FLT(33,I),DLN)
05228          CALL SLON(FLT(8,I),FLT(9,I),FLT(36,I),ALN)
05230          ff=1
05240          if(flt(6,i).eq.'FUEL'.or.flt(6,i).eq.'FLAG')ff=ff+1
05250          if(flt(13,i).eq.'FUEL'.or.flt(13,i).eq.'FLAG')ff=ff+2
05260          goto(301,302,303,304),ff
05270   301    continue
05271   303    decode(10,293,flt(6,i))dtl,dtg
05272          flt(6,i)=dtl
05273          flt(7,i)=dtg
05274          if(ff.eq.1)go to 302
05275          flt(13,i)=flt(6,i)+1
05276          flt(15,i)=flt(7,i)+1
05277          go to 310
05282   302    decode(15,292,flt(13,i))atl,atg
05283          flt(13,i)=atl
05284          flt(15,i)=atg
05285          if(ff.eq.1)go to 310
05286          flt(6,i)=flt(13,i)-1
05287          flt(7,i)=flt(15,i)-1
05288          go to 310
05289   292    format(i4,6x,i4)
05290   293    format(2(i4,1x))
05291   304    flt(6,i)=1200
05292          flt(7,i)=1200
05293          flt(13,i)=1201
05294          flt(15,i)=1201
05295   310    continue
05300          ida(i)=iadate(flt(6,i),flt(7,i),flt(13,i),flt(15,i),flt(20,i)
05400          1,afreq(1,i))
05500          prev(i)=0
05600   110    next(i)=0
05700          call fltleg(flt,4,11,leg,prev,next,lrec)
05750          call nonum(flt,next,prev,mdapt,maapt,leg,lrec)
05800          call legday(flt,afreq,next,prev,leg,lrec,ida,mdapt,maapt,
05900          jj=0
06000          do 120 ii=1,leg
06100          if(prev(ii).ne.0)go to 120
06200   120    jj=jj+1
06300          kk=0
06400          assign 120 to i120
06500          i=ii
06600   90     kk=kk+1
```

```
      list(kk)=i
      if(next(i).eq.0)go to 150
      if(next(i).lt.100)go to 140
      j=mod(next(i),100)
      next(i)=next(i)/100
      i=j
      assign 130 to i120
      go to 90
150   nf=0
      do 170 ll=1,7
170   nf=nf+afreq(ll,ii)
      write(21,1)flt(17,1),flt(18,1),jj,nf,kk
1     format(1h0,a2,a5,i1,2i2)
      do 160 ll=1,kk
      i=list(ll)
      llp=ll+1
      if(llp.gt.kk)go to 180
      j=list(llp)
      atim=flt(13,i)
      dtim=flt(6,j)
      if(atim.gt.dtim)dtim=dtim+2400
      dh=dtim/100
      dm=mod(dtim,100)
      ah=atim/100
      am=mod(atim,100)
      dtim=dh*60+dm
      atim=ah*60+am
      gtim=dtim-atim
      go to 220
180   gtim=0
220   continue
      decode(5,225,flt(15,i))eqp
225   format(1x,a3)
      write(21,2)mdapt(i),maapt(i),eqp,flt(28,i),gtim,
      1(flt(j,i),j=1,4),
      2(flt(j,i),j=6,11),
      2(flt(j,i),j=13,27),
      3(flt(j,i),j=29,lrec2),(afreq(j,i),j=1,7),ida(i)
160   continue
      go to i120,(120,130)
140    i=next(i)
      go to 90
120   continue
      nleg=nleg+leg
2     FORMAT(1x,2i6,a3,1x,a4,5x,i5,A1,a2,a1,a3,2i4
      1,A1,a2,a1,a3,i4,a1,i4,a1,a2,a5,A5,
      77i1,a2,2A4,2(I6,A5,A1),7i1,i1)
      return
      END
      SUBROUTINE SLON(WAC,RAC,SGN,LN)
      IMPLICIT INTEGER(A-Z)
      IF(LN)10,20,30
10    SGN='-'
      RETURN
50    CONTINUE
30    SGN='+'
      RETURN
20    IF(WAC.EQ.'0'.OR.WAC.EQ.'1'.OR.WAC.EQ.'2'.OR.WAC.EQ.'3'.OR.
      1WAC.EQ.'9')GO TO 10
      IF(WAC.EQ.'6'.OR.WAC.EQ.'7'.OR.WAC.EQ.'8')GO TO 30
      IF(WAC.EQ.'5')GO TO 50
```

```
C       HERE IF RAC=4
        IF(RAC.EQ.'39'.OR.RAC.EQ.'41'.OR.RAC.EQ.'69'.OR.RAC.EQ.'82'.OR.
       1RAC.EQ.'93') GO TO  10
        GO TO 30
        END
```

READLL.FOR

```
00100          subroutine readll(iapt,napt,lat,lon)
00150          parameter lth=100
00200          common lst(4,lth),num
00300          data init/0/
00400          if(init.gt.0)go to 10
00450          open(unit=1,file='usll.dat',access='seqin')
00500          do 20 i=1,lth
00600        read(1,1,end=30)lst(1,i),ltd,ltm,lts,ltsg
00700          1,lnd,lnm,lns,lnsg,lst(2,i)
00800    1     format(a3,3i2,a1,i3,2i2,a1,i6)
00900          lst(3,i)=lts+60*ltm+3600*ltd
01000          if(ltsg.eq.'S')lst(3,i)=lst(3,i)
01100          lst(4,i)=lns+60*lnm+3600*lnd
01200          if(lnsg.eq.'W')lst(4,i)=-lst(4,i)
01250          init=1
01300    20    continue
01350          type*,i
01400          num=lth
01500          go to 40
01600    30    num=i-1
01700    40    close(unit=1)
01800    10    continue
01900          i=jetapt(iapt)
02000          if(i.gt.0)go to 60
02200          napt=0
02300          lat=0
02400          lon=0
02500          return
02600    60    napt=lst(2,i)
02700          lat=lst(3,i)
02800          lon=lst(4,i)
02900          return
03000          end
```

REUDLL.FOR

```
00100          subroutine reudll(iapt,napt,lat,lon)
00150          parameter lth=100
00200          common/eull/ lst(4,lth),num
00300          data init/0/
00400          if(init.gt.0)go to 10
00450          open(unit=1,file='eull.dat',access='seqin')
00500          do 20 i=1,lth
00600        read(1,1,end=30)lst(1,i),ltd,ltm,lts,ltsg
00700          1,lnd,lnm,lns,lnsg,lst(2,i)
00800    1     format(a3,3i2,a1,i3,2i2,a1,i6)
00900          lst(3,i)=lts+60*ltm+3600*ltd
01000          if(ltsg.eq.'S')lst(3,i)=lst(3,i)
01100          lst(4,i)=lns+60*lnm+3600*lnd
01200          if(lnsg.eq.'W')lst(4,i)=-lst(4,i)
01250          init=1
01300   20     continue
01350          type*,i
01400          num=lth
01500          go to 40
01600   30     num=i-1
01700   40     close(unit=1)
01800   10     continue
01900          i=jeuapt(iapt)
02000          if(i.gt.0)go to 60
02200          napt=0
02300          lat=0
02400          lon=0
02500          return
02600   60     napt=lst(2,i)
02700          lat=lst(3,i)
02800          lon=lst(4,i)
02900          return
03000          end
```

JETAPT.FOR

```
03000              function jetapt(smgt)
03050              implicit integer(a-z)
03100              common tmgt(4,100),im
03200              i7=1
03300              i6=im
03400       121    if(i7.gt.i9)go to 126
03500              m8=(i7+i9)/2
03600              if(tmgt(1,m8)-smgt)124,127,125
03700       124    i7=m8+1
03800              go to 121
03900       125    i8=m8-1
04000              go to 121
04100       126    m8=-i8
04300       127    jetapt=m8
04400              return
04500              end
```

JEUAPT.FOR

```
03000              function jeuapt(smgt)
03050              implicit integer(a-z)
03100              common/eull/ tmgt(4,100),im
03200              i8=1
03300              l8=im
03400       121    if(i8.gt.l8)go to 126
03500              m8=(i8+l8)/2
03600              if(tmgt(1,m8)-smgt)124,127,125
03700       124    i8=m8+1
03800              go to 121
03900       125    l8=m8-1
04000              go to 121
04100       126    m8=-l8
04300       127    jeuapt=m8
04400              return
04500              end
```

IADATE.FOR

```
00100          function iadate(dtl,dtg,atl,atg,dfr,afr)
00200          implicit integer(a-z)
00300          dimension dfr(7),afr(7)
00400          iadate=0
00500          dgl=dtg-dtl
00600          if(dgl.gt.1200)iadate=iadate-1
00650          if(dgl.lt.-1200)iadate=iadate+1
00700          if(atg.lt.dtg)iadate=iadate+1
00800          alg=atl-atg
00830          if(alg.gt.1200)iadate=iadate-1
00855          if(alg.lt.-1200)iadate=iadate+1
00900          do 10 i=1,7
01000          j=i+iadate
01100          if(j.le.0) j=j+7
01200          if(j.gt.7)j=j-7
01300          afr(j)=dfr(i)
01400     10   continue
01500          return
01600          end
```

FLTLEG.FOR

```
00100          subroutine fltleg(leg,id,ia,m,ip,in,nleg)
00200          dimension leg(nleg,m),ip(m),in(m)
00300          if(m.le.1)return
00350          m1=m-1
00400          do 10 i=1,m1
00500          j=i+1
00600          do 20 k=j,m
00700          if(leg(id,i).ne.leg(ia,k))go to 30
00725    1     format(2i3,7i10/8i10)
00750          if(ip(i).gt.100000000.or.in(k).gt.100000000)print 1,i,k,
00775          1(ip(n),in(n),n=1,m)
00800          if(ip(i).lt.100000000)ip(i)=100*ip(i)+k
00900          if(in(k).lt.100000000)in(k)=100*in(k)+i
01000    30    if(leg(ia,i).ne.leg(id,k))go to 20
01040          if(ip(k).gt.100000000.or.in(i).gt.100000000)print 1,i,k,
01080          1(ip(n),in(n),n=1,m)
01100          if(ip(k).lt.100000000)ip(k)=100*ip(k)+i
01200          if(in(i).lt.100000000)in(i)=100*in(i)+k
01300    20    continue
01400    10    continue
01500          return
01600          end
```

LEG DAY. FOR

```
00100           subroutine legday(flt,afreq,next,prev,leg,lrec,ida,mdapt,maapt)
00200           implicit integer(a-z)
00300           dimension flt(lrec,leg),afreq(7,leg),next(leg),prev(leg)
00400           dimension ida(leg),afn(7,2),dfn(7,2),mdapt(leg),maapt(leg)
00475   300     CONTINUE
00500   100     CONTINUE
00510   C:DO 100 i=1,leg
00525   C:1100:TYPE 1,flt(4,i),flt(6,i),(flt(j,i),j=20,26),prev(i),i
00537   C       1,flt(11,i),flt(13,i),(afreq(j,i),j=1,7),next(i)
00543   1       format(1x,a3,i5,i2,6i1,I7,1h-,i2,1h-,a3,i5,i2,6i1,I7)
00546   C       do 10 ii=1,leg
00549           DO 30 II=1,LEG
00550           i=ii
00600           if(next(i).lt.1)go to 10
00650           k=next(i)
00700   C       next(i)=0
00800   20      j=mod(k,100)
00900           kall=kheck(flt(13,i),afreq(1,i),next(i),i,flt(6,j),flt(20,j)
01000           1,prev(j),j,afn,dfn)
01025   C:TYPE*,i,k,j,flt(13,i),next(i),flt(6,j),prev(j),KALL
01050           if(kall.gt.0)go to 50
01100   500     k=k/100
01200           if(k.gt.0)go to 20
01300   10      continue
01400   C300    do 30 ii=1,leg
01450           i=ii
01500           if(prev(i).lt.1)go to 30
01600           k=prev(i)
01700   C       prev(i)=0
01800   40      j=mod(k,100)
01900           kall=kheck(flt(13,j),afreq(1,j),next(j),j,flt(6,i),flt(20,i)
02000           1,prev(i),i,afn,dfn)
02025   C:TYPE*,i,k,j,flt(13,j),next(j),flt(6,i),prev(i),KALL
02050           if(kall.gt.0)go to 60
02100   600     k=k/100
02200           if(k.gt.0)go to 40
02300   30      continue
02400           return
02410   50      if(addflt(i,j,kall,leg,lrec,flt,next,prev,afreq,afn,dfn,ida
02420           1,mdapt,maapt)) 100,500,100
02430   60      if(addflt(j,i,kall,leg,lrec,flt,next,prev,afreq,afn,dfn,ida
02440           1,mdapt,maapt)) 300,600,300
02500           end
02600           FUNCTION kheck(at,af,an,ai,dt,df,dn,di,afn,dfn)
02650   CHECK ARRIVAL AT ONE SEGMENT TO SEE IF IT IS LINKED WITH DEPARTURE FRO
02675   C       ANOTHER SPECIFIED SEGMENT.
02700           implicit integer(a-z)
02800           dimension af(7),df(7),dmf(7),nf(4),afn(7,2),dfn(7,2)
02900           id=1
02910           kheck=0
02955   CHECK IF MIDNIGHT DURING GROUND TIME(30 MINUTE ALLOWANCE FOR LAD)
03000           if(at.lt.dt+30)id=0
03025   C       MAXIMUM 6 HOUR LAYOVER ALLOWED
03070           if(dt+2400*id-at.gt.600)go to 50
03100           do 20 i= 1,4
03200   20      nf(i)=0
03210           do 70 k=1,2
03220           do 70 i=1,7
03230           afn(i,k)=0
03240   70      dfn(i,k)=0
03270           kk=0
```

```
03300            do 10 i=1,7
03400            j=i+id
03500            if(j.eq.8)j=1
03600            if(af(i)+df(j).eq.2)nf(4)=nf(4)+1
03610    c       nf(4)=frequency[#days]
03700            kf=af(i)-df(j)+2
03800            nf(kf)=nf(kf)+1
03810            go to(1,2,3),kf
03820    1       dfn(j,2)=df(j)
03830    c       departure, but no arrival
03840            go to 10
03850    2       afn(i,1)=af(i)
03860            dfn(j,1)=df(j)
03865            if(afn(i,1).eq.1.or.dfn(j,1).eq.1)km=1
03870    c       both agree
03880            go to 10
03890    3       afn(i,2)=af(i)
03895    c       arrival, but no departure
03900    10      continue
03950            if(nf(4).eq.0)go to 30
04000            if(nf(2).ne.7)go to 40
04100            an=di
04200            dn=ai
04300            return
04350    60      continue
04375            TYPE *,at,dt,id,an,dn,ai,di
04400    30      continue
04410    c       no connection
04500            call delete(an,di)
04600            call delete(dn,ai)
04700            return
04800    40      if(km.eq.0)go to 30
04850            call add(an,di)
04900            call add(dn,ai)
04910    c       part connection
04920            if(nf(3).gt.0)kheck=kheck+1
04930    c       #days arrival, but no departure¬
04940            if(nf(1).gt.0)kheck=kheck+2
04950    c       #days departure, but no arrival¬
04975    C:TYPE*,KHECK,NF,AN,DI,DN,AI
05000            return
05100            end
05200            subroutine add(n,i)
05225            if(n.gt.100000000)go to 20
05250            k=i
05300            m=n
05400    10      j=mod(m,100)
05500            if(i.eq.j)k=0
05600            m=m/100
05700            if(m.gt.0)go to 10
05750            if(k.gt.0)n=100*n+k
05800            return
05810    20      TYPE *,n,i
05820            return
05900            end
06000            subroutine delete(n,i)
06100            m=n
06150            n=0
06200    10      j=mod(m,100)
06300            if(j.ne.i)n=100*n+j
06400            m=m/100
```

```
      if(m.gt.0)go to 10
      return
      end
       functionaddflt(ia,id,kall,leg,lrec,flt,next,prev,afreq,
     1afn,dfn,ida,mdapt,maapt)
      implicit integer(a-z)
      dimension flt(lrec,leg),afreq(7,leg),next(leg),prev(leg)
      dimension ida(leg),afn(7,2),dfn(7,2),mdapt(leg),maapt(leg)
      addflt=leg
      if(leg.ge.99)go to 110
      if(kall.eq.2)go to 180
      leg=leg+1
      do 80 l=1,lrec
80    flt(l,leg)=flt(l,ia)
      mdapt(leg)=mdapt(ia)
      maapt(leg)=maapt(ia)
      ida(leg)=ida(ia)
      next(leg)=next(ia)
      call delete(next(ia),id)
      prev(leg)=prev(ia)
      call delete(prev(id),ia)
C:TYPE*,IA,PREV(IA),NEXT(IA),LEG,PREV(LEG),NEXT(LEG)
      f=0
      fi=0
      do 90 l=1,7
      f=f+afn(l,1)
      fi=fi+afn(l,2)
      afreq(l,leg)=afn(l,1)
      afreq(l,ia)=afn(l,2)
      ld=l+ida(ia)
      if(ld.gt.7)ld=1
      flt(l+19,ia)=afn(ld,2)
90    flt(l+19,leg)=afn(ld,1)
      if(fi.gt.0)go to 10
      next(ia)=next(leg)
      do 20 i=1,7
      afreq(i,ia)=afreq(i,leg)
20    flt(i+19,ia)=flt(i+19,leg)
      f=fi
10    continue
      if(f.eq.0)leg=leg-1
      if(f.ne.0)call legadd(leg,next,prev)
C:TYPE*,IA,PREV(IA),NEXT(IA),LEG,PREV(LEG),NEXT(LEG)
      if(kall.eq.1)go to 200
      if(leg.ge.99)go to 110
180   leg=leg+1
      do 800 l=1,lrec
800   flt(l,leg)=flt(l,id)
      mdapt(leg)=mdapt(id)
      maapt(leg)=maapt(id)
      ida(leg)=ida(id)
      prev(leg)=prev(id)
      call delete(prev(id),ia)
C     prev(leg)=prev(ia)
      next(leg)=next(id)
      CALL DELETE(NEXT(IA),ID)
C:TYPE*,ID,PREV(ID),NEXT(ID),LEG,PREV(LEG),NEXT(LEG)
      f=0
      fi=0
      do 900 l=1,7
      f=f+dfn(l,1)
```

```
                fi=fi+dfn(l,2)
                flt(l+19,id)=dfn(l,2)
                flt(l+19,leg)=dfn(l,1)
                la=l-ida(id)
                if(la.lt.1)la=7
                afreq(l,id)=dfn(la,2)
                afreq(l,leg)=dfn(la,1)
      900       continue
                if(fi.gt.0)go to 30
                f=fi
                do 40 i=1,7
                afreq(i,id)=afreq(i,leg)
      40        flt(i+19,id)=flt(i+19,leg)
                prev(id)=prev(leg)
      30        continue
                if(f.eq.0)leg=leg-1
                if(f.ne.0)call legadd(leg,next,prev)
      C:TYPE*,ID,PREV(ID),NEXT(ID),LEG,PREV(LEG),NEXT(LEG)
      200       addflt=leg-addflt
                return
      110       TYPE 3,ia,id,leg,(prev(l),next(l),l=1,leg)
                go to 200
      3         format(3i3,7i10/8i10)
                end
                subroutine legadd(leg,next,prev)
                implicit integer(a-z)
                dimension next(leg),prev(leg)
                k=next(leg)
                if(k.le.0)go to 10
      20        j=mod(k,100)
                call add(prev(j),leg)
                k=k/100
                if(k.gt.0)go to 20
      10        k=prev(leg)
                if(k.le.0)go to 30
      40        j=mod(k,100)
                call add(next(j),leg)
                k=k/100
                if(k.gt.0)go to 40
      30        return
                end
```

PADIS.FOR

```
00100          implicit integer(a-z)
00200          parameter mx=99
00300          dimension ndc(mx),nac(mx),eqp(mx),tt(mx),dis(mx),gt(mx)
00400          1,d1(6,mx),a1(6,mx),fc(mx),et(mx),sc(mx),sc2(mx),
00500          2op(mx),ed(mx),dd(mx),dad(mx)
00550          open(unit=1,access='seqin',file='FLT',DEVICE='PT49')
00570          open(unit=20,access='seqout',file='PR',DEVICE='PT49')
00580          open(unit=21,access='seqout',file='DIS',DEVICE='PT49')
00600    10    read(1,1,end=1000)h0,ca,fl,fls,nfr,nseq
00700          if(h0.eq.'0')go to 20
00800          type 1,h0,ca,fl,fls,nfr,nseq
00900          go to 10
01000    20    WRITE(21,1)H0,CA,FI,FIS,NFR,NSEQ
01050          do 30 l=1,nseq
01100    1     format(a1,a2,a5,a1,2i2)
01200          read(1,2)hs,ndc(l),nac(l),eqp(l),tt(l),dis(l),gt(l),(d1(i,l)
01300          1i=1,4),(a1(i,l),i=1,3),fc(l),a1(4,l),et(l),
01400          2car,flr,sc(l),d1(5,l),d1(6,l),sc2(l),op(l),ed(l),dd(l),
01450          3DLAT,DLON,DSGN,ALAT,ALON,ASGN,a1(5,l)
01500          3,a1(6,l),dad(l)
01600    2     format(a1,i6,i6,a3,i5,i5,i5,A4,A3,I4,2A4,A3,I4,i1,a4
01650          9,a1,a2,a5,a5,a5,a2,i1
01700          1,a1,2a4,2(I6,I5,A1),a5,a2,i1)
01800          if(hs.eq.' '.and.car.eq.ca.and.flr.eq.fl)go to 40
01900          type 2,hs,ndc(l),nac(l),eqp(l),tt(l),dis(l),gt(l),(d1(i,l),
02000          1i=1,3),(a1(i,l),i=1,3),fc(l),a1(4,l),et(l),
02100          2car,flr,sc(l),d1(4,l),d1(5,l),sc2(l),op(l),ed(l),dd(l),a1(5,l)
02200          3,a1(6,l),dad(l)
02300          go to 30
02400    40    continue
02450          DIS(L)=IDIST(DLAT,DLON,DSGN,ALAT,ALON,ASGN)
02475          WRITE(21,4)NDC(L),NAC(L),EQP(L),TT(L),DIS(L),GT(L),D1(2,L)
02487          1,A1(2,L)
02493    4     FORMAT(1X,2I6,A3,3I5,2A3)
02500    30    continue
02550          sq=0
02600          do 100 i=1,nseq
02700          if(ndc(i).le.0)go to 100
02800          do 200 j=i,nseq
02900          if(nac(j).le.0)go to 200
02910          ttmax=0
02920          ltt=0
02930          ttsum=0
02940          disum=0
02950          gtsum=0
02960          fcval=0
02965          dadsum=0
02970          sc2sum=0
02980          edval=' '
02990          ddval=' '
03000          do 300 l=i,j
03100          if(tt(l).le.ttmax)go to 50
03200          ltt=l
03300          ttmax=tt(l)
03400    50    ttsum=ttsum+tt(l)
03500          disum=disum+dis(l)
03600          if(l.ne.j)gtsum=gtsum+gt(l)
03700          fcval=max0(fcval,fc(l))
03800          sc2sum=sc2sum+sc2(l)
03900          if(ed(l).ne.' ')edval=ed(l)
04000          if(dd(l).ne.' ')ddval=dd(l)
```

```
04100            dadsum=dadsum+dad(l)
04110            IF(I.GE.J)GOTO 300
04120            ID=1
04130            IF(A1(3,I).LT.D1(3,I+1)+30)ID=0
04140            DADSUM=DADSUM+ID
04200    300     continue
04250            sq=sq+1
04275            nstop=j-i
04300            write(20,3)ca,fl,fls,sq,nir,nstop,nac(i),nac(j),
04400            1egp(ltt),ttsum,disum,gtsum,(d1(ii,i),ii=1,6),(a1(ii,j),ii=1,6)
04500            2,fcval,et(ltt),sc(ltt),sc2sum,op(ltt),edval,ddval,dadsum
04600    3       format(1x,a2,a5,a1,i2,i2,i2,2i6,a3,3i5,A4,A3,I4,A4,a5,a2,A4,A3,
04650            9I4,a4,a5,a2
04700            1,i1,a1,a5,i1,a1,2a4,i1)
04800    200     continue
04900    100     continue
05000            go to 10
05100    1000    continue
05200            close(unit=1)
05300            close(unit=20)
05400            stop
05500            end
05600            FUNCTION IDIST(LAT,LON,LSGN,MAT,MON,MSGN)
05610            DATA PI/3.1415926535/,init/0/
05615            if(init.ne.0)goto 10
05617            init=1
05620            STR=PI/(180.*3600.)
05670            RTM=10800./PI
05685            rtm=rtm*1.151
05692    c       Statute Miles
05700    10      ALT=FLOAT(LAT-648000)*STR
05800            AMT=FLOAT(MAT-648000)*STR
05900            ALN=FLOAT(LON)*STR
06000            IF(LSGN.EQ.'-')ALN=-ALN
06100            AMN=FLOAT(MON)*STR
06200            IF(MSGN.EQ.'-')AMN=-AMN
06300            DN=ABS(ALN-AMN)*10.
06400            IF(DN.GT.PI)DN=PI+PI-DN
06500            COSDIS=COS(ALT)*COS(AMT)+SIN(ALT)*SIN(AMT)*COS(DN)
06600            DIS=ACOS(COSDIS)
06700            IDIST=DIS*RTM
06800            RETURN
06900            END
```

USHFLR.FOR

```
      DIMENSION IUP(3),IEP(3),IUC(3)
      DATA JUP,JUC/2*0/
      OPEN(UNIT=1,ACCESS='SEQIN',FILE='USHFIL.DAT')
      OPEN(UNIT=20,ACCESS='SEQOUT',FILE='USHCON.DAT')
10    READ(1,1,END=1000)IUP,IAL,IEP,ISEQ,IUC
      IF(IUP(1).EQ.IUC(1))GO TO 10
      IF(IUP(1).EQ.JUP.AND.IUC(1).EQ.JUC)GO TO 10
1     FORMAT(I6,I2,I4,1X,A2,I6,I2,I4,I9,I6,I2,I4)
      JUP=IUP(1)
      JUC=IUC(1)
      WRITE(20,2)JUP,JUC
2     FORMAT(2I6)
      GO TO 10
1000  CLOSE(UNIT=1)
      CLOSE(UNIT=20)
      STOP
      END
```

EUHFLR.FOR

```
00050          DIMENSION IUP(3),IEP(3),IUC(3)
00075          DATA JUP,JUC/2*0/
00100          OPEN(UNIT=1,ACCESS='SEQIN',FILE='EUHFLR.DAT')
00200          OPEN(UNIT=20,ACCESS='SEQOUT',FILE='EUHCON.DAT')
00300    10    READ(1,1,END=1000)IUP,IAI,IEP,ISEQ,IUC
00400          IF(IEP(1).EQ.IUC(1))GO TO 10
00500          IF(IEP(1).EQ.JUP.AND.IUC(1).EQ.JUC)GO TO 10
00600    1     FORMAT(I6,I2,I4,1X,A2,I6,I2,I4,I4,I6,I2,I4)
00700          JUP=IEP(1)
00800          JUC=IUC(1)
00900          WRITE(20,2)JUP,JUC
00950    2     FORMAT(2I6)
01000          GO TO 10
01100    1000  CLOSE(UNIT=1)
01200          CLOSE(UNIT=20)
01300          STOP
01400          END
```

PDUPDU.FOR

```
00100            implicit integer(a-z)
00200            parameter mx=1
00300            double precision fo
00400            dimension ndc(2),nac(2),ttp(mx),tt(mx),fls(mx),et(mx)
00500            1,d1(5,mx),a1(6,mx),fc(mx),et(mx),sc(mx),sc2(mx),
00600            2op(mx),ed(mx),dd(mx),dad(mx)
00700            i=1
00800            j=1
00900            ltt=1
01000            ext='PDU'
01100            open(unit=1,access='seqin',file='ARRUS',DEVICE='PT49')
01200            REWIND 1
01300            OPEN(UNIT=22,ACCESS='SEQOUT',FILE='BADPDU.DAT',DEVICE='PT49')
01400            OPEN(UNIT=21,ACCESS='SEQIN',FILE='USHCOW')
01500            OPEN(UNIT=20,ACCESS='SEQOUT',FILE='PDUPDU.DAT',DEVICE='PT49')
01600    2       FORMAT(2I6)
01700            read(1,3,end=1000)ca,fl,fls,sq,nfr,nstop,NDC,NAC,
01800            1egp(ltt),ttsum,disum,gtsum,(d1(ii,i),ii=1,5),(a1(ii,j),ii=1,6)
01900            2,fcval,et(ltt),sc(ltt),sc2sum,op(ltt),edval,ddval,dadsum
02000            TYPE 3,ca,fl,fls,sq,nfr,nstop,NDC,NAC,
02100            1egp(ltt),ttsum,disum,gtsum,(d1(ii,i),ii=1,5),(a1(ii,j),ii=1,6)
02200    C       2,fcval,et(ltt),sc(ltt),sc2sum,op(ltt),edval,ddval,dadsum
02300    10      CONTINUE
02400    C:dapt=d1(2,i)
02500    C:      aapt=a1(2,j)
02600            READ(21,2,END=1000)DAPT,AAPT
02700    C       TYPE 2,DAPT,AAPT
02800    C       encode(10,1,fo)aapt,ext
02900    1       format(15,1h.,a3)
03000    C       open(unit=20,access='seqout',file=fo,DEVICE='PT49')
03100    40      IF(DAPT-NDC(1))60,50,90
03200    50      IF(AAPT-NAC(1))60,20,90
03300    20      write(20,3)ca,fl,fls,sq,nfr,nstop,NDC,NAC,
03400            1egp(ltt),ttsum,disum,gtsum,(d1(ii,i),ii=1,5),(a1(ii,j),ii=1,6)
03500            2,fcval,et(ltt),sc(ltt),sc2sum,op(ltt),edval,ddval,dadsum
03600    3       format(1x,a2,a5,a1,i2,i2,i2,2(I5,I1),a3
03700            9,3i5,a4,a3,3a5,a4,a3,a4,a4,a5,a2
03800            1,i1,a1,a5,i1,a1,2a4,i1)
03900    30      read(1,3,end=1000)ca,fl,fls,sq,nfr,nstop,NDC,NAC,
04000            1egp(ltt),ttsum,disum,gtsum,(d1(ii,i),ii=1,5),(a1(ii,j),ii=1,6)
04100            2,fcval,et(ltt),sc(ltt),sc2sum,op(ltt),edval,ddval,dadsum
04200            GO TO 40
04300    90      write(22,3)ca,fl,fls,sq,nfr,nstop,NDC,NAC,
04400            1egp(ltt),ttsum,disum,gtsum,(d1(ii,i),ii=1,5),(a1(ii,j),ii=1,6)
04500            2,fcval,et(ltt),sc(ltt),sc2sum,op(ltt),edval,ddval,dadsum
04600            GO TO 30
04700    60      CONTINUE
04800    C:close(unit=20)
04900    C       TYPE 3,ca,fl,fls,sq,nfr,nstop,NDC,NAC,
05000    C       1egp(ltt),ttsum,disum,gtsum,(d1(ii,i),ii=1,5),(a1(ii,j),ii=1,6)
05100    C       2,fcval,et(ltt),sc(ltt),sc2sum,op(ltt),edval,ddval,dadsum
05200            go to 10
05300    1000    continue
05400            close(unit=1)
05500            close(unit=20)
05600            stop
05700            end
```

```
00100           implicit integer(a-z)
00200           parameter mx=1
00300           double precision fo
00400           dimension ndc(2),nac(2),ttsum(mx),tt(mx),dis(mx),gt(mx)
00500           1,d1(5,mx),a1(5,mx),fc(mx),et(mx),sc(mx),sc2(mx),
00600           op(mx),ed(mx),dd(mx),dad(mx)
00700           i=1
00800           j=1
00900           ltt=1
01000           ext='PDE'
01100           open(unit=1,access='seqin',file='ARREU',DEVICE='PT49')
01200           REWIND 1
01300           OPEN(UNIT=22,ACCESS='SEQOUT',FILE='BADPDE.DAT',DEVICE='PT49')
01400           OPEN(UNIT=21,ACCESS='SEQIN',FILE='EUHCON')
01500           OPEN(UNIT=20,ACCESS='SEQOUT',FILE='PDEPDE.DAT',DEVICE='PT49')
01600   2       FORMAT(2I6)
01700           read(1,3,end=1000)ca,fl,fls,sq,nfr,nstop,NDC,NAC,
01800           1eqp(ltt),ttsum,disum,gtsum,(d1(ii,i),ii=1,5),(a1(ii,j),ii=1,
01900           2,fcval,et(ltt),sc(ltt),sc2sum,op(ltt),edval,ddval,dadsum
02000           TYPE 3,ca,fl,fls,sq,nfr,nstop,NDC,NAC,
02100           1eqp(ltt),ttsum,disum,gtsum,(d1(ii,i),ii=1,5),(a1(ii,j),ii=1,
02200   C       2,fcval,et(ltt),sc(ltt),sc2sum,op(ltt),edval,ddval,dadsum
02300   10      CONTINUE
02400   C:dapt=d1(2,i)
02500   C:      aapt=a1(2,j)
02600           READ(21,2,END=1000)DAPT,AAPT
02700   C       TYPE 2,DAPT,AAPT
02800           encode(10,1,fo)aapt,ext
02900   1       format(I5,1h.,a3)
03000   C       open(unit=20,access='seqout',file=fo,DEVICE='PT49')
03100   40      IF(DAPT-NDC(1))60,30,90
03200   50      IF(AAPT-NAC(1))60,20,90
03300   20      write(20,3)ca,fl,fls,sq,nfr,nstop,NDC,NAC,
03400           1eqp(ltt),ttsum,disum,gtsum,(d1(ii,i),ii=1,5),(a1(ii,j),ii=1
03500           2,fcval,et(ltt),sc(ltt),sc2sum,op(ltt),edval,ddval,dadsum
03600   3       format(1x,a2,a5,a1,i2,i2,i2,2(I5,I1),a3
03700           9,3i5,a4,a3,3a5,a4,a3,a4,a4,a5,a2
03800           1,i1,a1,a5,i1,a1,2a4,i1)
03900   30      read(1,3,end=1000)ca,fl,fls,sq,nfr,nstop,NDC,NAC,
04000           1eqp(ltt),ttsum,disum,gtsum,(d1(ii,i),ii=1,5),(a1(ii,j),ii=1,
04100           2,fcval,et(ltt),sc(ltt),sc2sum,op(ltt),edval,ddval,dadsum
04200           GO TO 40
04300   90      write(22,3)ca,fl,fls,sq,nfr,nstop,NDC,NAC,
04400           1eqp(ltt),ttsum,disum,gtsum,(d1(ii,i),ii=1,5),(a1(ii,j),ii=1,
04500           2,fcval,et(ltt),sc(ltt),sc2sum,op(ltt),edval,ddval,dadsum
04600           GO TO 30
04700   60      CONTINUE
04800   C:close(unit=20)
04900   C       TYPE 3,ca,fl,fls,sq,nfr,nstop,NDC,NAC,
05000   C       1eqp(ltt),ttsum,disum,gtsum,(d1(ii,i),ii=1,5),(a1(ii,j),ii=1,
05100   C       2,fcval,et(ltt),sc(ltt),sc2sum,op(ltt),edval,ddval,dadsum
05200           go to 10
05300   1000    continue
05400           close(unit=1)
05500           close(unit=20)
05600           stop
05700           end
```

OPUOPU.FOR

```
00100            implicit integer(a-z)
00200            parameter mx=1
00300            double precision fo
00400            dimension nac(2),rac(2),eqp(mx),tt(mx),dis(mx),gt(mx)
00500            1,d1(5,mx),a1(6,mx),fc(mx),et(mx),sc(mx),sc2(mx),
00600            2op(mx),ed(mx),dd(mx),dad(mx)
00700            i=1
00800            j=1
00900            ltt=1
01000            ext='OPU'
01100            OPEN(unit=1,access='seqin',file='DEPUS',DEVICE='PT49')
01150            REWIND 1
01200            OPEN(UNIT=22,ACCESS='SEQOUT',FILE='BADOPU.DAT',DEVICE='PT49')
01300            OPEN(UNIT=21,ACCESS='SEQIN',FILE='USHCON')
01350            OPEN(UNIT=20,ACCESS='SEQOUT',FILE='OPUOPU.DAT',DEVICE='PT49')
01400    2       FORMAT(2I6)
01500            read(1,3,end=1000)ca,fl,fls,sq,nfr,nstop,NDC,NAC,
01600            1eqp(ltt),ttsum,disum,gtsum,(d1(ii,i),ii=1,5),(a1(ii,j),ii=1,6)
01700            2,fcval,et(ltt),sc(ltt),sc2sum,op(ltt),edval,ddval,dadsum
01725            TYPE 3,ca,fl,fls,sq,nfr,nstop,NDC,NAC,
01750            1eqp(ltt),ttsum,disum,gtsum,(d1(ii,i),ii=1,5),(a1(ii,j),ii=1,6)
01775            2,fcval,et(ltt),sc(ltt),sc2sum,op(ltt),edval,ddval,dadsum
01800    10      CONTINUE
01900    C:dapt=d1(2,i)
02000    C:      aapt=a1(2,j)
02100            READ(21,2,END=1000)AAPT,DAPT
02150    C       TYPE 2,AAPT,DAPT
02200            encode(10,1,fo)aapt,ext
02300    1       format(I5,1h.,a3)
02400    C       OPEN(unit=20,access='seqout',file=fo,DEVICE='PT49')
02500    40      IF(AAPT-NAC(1))60,50,90
02600    50      IF(DAPT-NDC(1))60,20,90
02700    20      write(20,3)ca,fl,fls,sq,nfr,nstop,NDC,NAC,
02800            1eqp(ltt),ttsum,disum,gtsum,(d1(ii,i),ii=1,5),(a1(ii,j),ii=1,6)
02900            2,fcval,et(ltt),sc(ltt),sc2sum,op(ltt),edval,ddval,dadsum
03000    3       format(1x,a2,a5,a1,i2,i2,i2,2(I5,I1),a3
03100            9,3i5,a4,a3,3a5,a4,a3,a4,a4,a5,a2
03200            1,i1,a1,a5,i1,a1,2a4,i1)
03300    30      read(1,3,end=1000)ca,fl,fls,sq,nfr,nstop,NDC,NAC,
03400            1eqp(ltt),ttsum,disum,gtsum,(d1(ii,i),ii=1,5),(a1(ii,j),ii=1,6)
03500            2,fcval,et(ltt),sc(ltt),sc2sum,op(ltt),edval,ddval,dadsum
03600            GO TO 40
03700    90      write(22,3)ca,fl,fls,sq,nfr,nstop,NDC,NAC,
03800            1eqp(ltt),ttsum,disum,gtsum,(d1(ii,i),ii=1,5),(a1(ii,j),ii=1,6)
03900            2,fcval,et(ltt),sc(ltt),sc2sum,op(ltt),edval,ddval,dadsum
04000            GO TO 30
04100    60      CONTINUE
04112    C:close(unit=20)
04125    C       TYPE 3,ca,fl,fls,sq,nfr,nstop,NDC,NAC,
04150    C       1eqp(ltt),ttsum,disum,gtsum,(d1(ii,i),ii=1,5),(a1(ii,j),ii=1,6)
04175    C       2,fcval,et(ltt),sc(ltt),sc2sum,op(ltt),edval,ddval,dadsum
04200            go to 10
04300    1000    continue
04400            close(unit=1)
04500            close(unit=20)
04600            stop
04700            end
```

CPEOPE.FOR

```
00100           implicit integer(a-z)
00200           parameter mx=1
00300           double precision fo
00400           dimension ndc(2),nac(2),eqp(mx),tt(mx),dis(mx),gt(mx)
00500           1,d1(5,mx),a1(5,mx),fc(mx),et(mx),sc(mx),sc2(mx),
00600           2op(mx),ed(mx),dd(mx),dad(mx)
00700           i=1
00800           j=1
00900           ltt=1
01000           ext='OPE'
01100           open(unit=1,access='seqin',file='DEPEU',DEVICE='PT49')
01150           REWIND 1
01200           OPEN(UNIT=22,ACCESS='SEQOUT',FILE='BADOPE.DAT',DEVICE='PT49')
01300           OPEN(UNIT=21,ACCESS='SEQIN',FILE='EUHCON')
01350           OPEN(UNIT=20,ACCESS='SEQOUT',FILE='CPEOPE.DAT',DEVICE='PT49')
01400   2       FORMAT(2I6)
01500           read(1,3,end=1000)ca,fl,fls,sq,nfr,nstop,NDC,NAC,
01600           1eqp(ltt),ttsum,disum,gtsum,(d1(ii,i),ii=1,5),(a1(ii,j),ii=1,6
01700           2,fcval,et(ltt),sc(ltt),sc2sum,op(ltt),edval,ddval,dadsum
01725           TYPE 3,ca,fl,fls,sq,nfr,nstop,NDC,NAC,
01750           1eqp(ltt),ttsum,disum,gtsum,(d1(ii,i),ii=1,5),(a1(ii,j),ii=1,6
01775           2,fcval,et(ltt),sc(ltt),sc2sum,op(ltt),edval,ddval,dadsum
01800   10      CONTINUE
01900   C:dapt=d1(2,i)
02000   C:      aapt=a1(2,j)
02100           READ(21,2,END=1000)AAPT,DAPT
02150           TYPE 2,AAPT,DAPT
02200           encode(10,1,fo)aapt,ext
02300   1       format(I5,1h.,a3)
02400   C       open(unit=20,access='seqout',file=fo,DEVICE='PT49')
02500   40      IF(AAPT-NAC(1))60,50,90
02600   50      IF(DAPT-NDC(1))60,20,90
02700   20      write(20,3)ca,fl,fls,sq,nfr,nstop,NDC,NAC,
02800           1eqp(ltt),ttsum,disum,gtsum,(d1(ii,i),ii=1,5),(a1(ii,j),ii=1,6
02900           2,fcval,et(ltt),sc(ltt),sc2sum,op(ltt),edval,ddval,dadsum
03000   3       format(1x,a2,a5,a1,i2,i2,i2,2(I5,I1),a3
03100           2,3i5,a4,a3,3a5,a4,a3,a4,a4,a5,a2
03200           1,i1,a1,a5,i1,a1,2a4,i1)
03300   30      read(1,3,end=1000)ca,fl,fls,sq,nfr,nstop,NDC,NAC,
03400           1eqp(ltt),ttsum,disum,gtsum,(d1(ii,i),ii=1,5),(a1(ii,j),ii=1,6
03500           2,fcval,et(ltt),sc(ltt),sc2sum,op(ltt),edval,ddval,dadsum
03600           GO TO 40
03700   90      write(22,3)ca,fl,fls,sq,nfr,nstop,NDC,NAC,
03800           1eqp(ltt),ttsum,disum,gtsum,(d1(ii,i),ii=1,5),(a1(ii,j),ii=1,6
03900           2,fcval,et(ltt),sc(ltt),sc2sum,op(ltt),edval,ddval,dadsum
04000           GO TO 30
04100   60      CONTINUE
04112   C:close(unit=20)
04125           TYPE 3,ca,fl,fls,sq,nfr,nstop,NDC,NAC,
04150           1eqp(ltt),ttsum,disum,gtsum,(d1(ii,i),ii=1,5),(a1(ii,j),ii=1,6
04175           2,fcval,et(ltt),sc(ltt),sc2sum,op(ltt),edval,ddval,dadsum
04200           go to 10
04300   1000    continue
04400           close(unit=1)
04500           close(unit=20)
04600           stop
04700           end
```

USEUHP.FOR

```
      DIMENSION IUP(3),IEP(3),IUC(3)
      DATA JUP,JUC/2*0/
      OPEN(UNIT=1,ACCESS='SEQIN',FILE='HFILE2.DAT')
      OPEN(UNIT=20,ACCESS='SEQOUT',FILE='USEUHP.DAT')
10    READ(1,1,END=1000)IUP,IAL,IEP,ISEQ,IUC
      IF(IUP(1).EQ.IUC(1))GO TO 10
      IF(IUP(1).EQ.JUP.AND.IEP(1).EQ.JUC)GO TO 10
1     FORMAT(I6,I2,I4,1X,A2,I6,I2,I4,I9,I6,I2,I4)
      JUP=IUP(1)
      JUC=IEP(1)
      WRITE(20,2)JUP,JUC
2     FORMAT(2I6)
      GO TO 10
1000  CLOSE(UNIT=1)
      CLOSE(UNIT=20)
      STOP
      END
```

USPEUP.FOR

```
      implicit integer(a-z)
      parameter mx=1
      double precision fo
      dimension ndc(2),nac(2),eqp(mx),tt(mx),dis(mx),gt(mx)
     1,d1(5,mx),a1(6,mx),fc(mx),et(mx),sc(mx),sc2(mx),
     2op(mx),ed(mx),dd(mx),dad(mx)
      i=1
      j=1
      ltt=1
      open(unit=1,access='seqin',file='SUSEU',DEVICE='PT49')
      REWIND 1
      OPEN(UNIT=22,ACCESS='SEQOUT',FILE='DDUSEU.DAT',DEVICE='PT49')
      OPEN(UNIT=21,ACCESS='SEQIN',FILE='USEUHP')
      OPEN(UNIT=20,ACCESS='SEQOUT',FILE='USPEUP.DAT',DEVICE='PT49')
2     FORMAT(2I6)
      read(1,3,end=1000)ca,fl,fls,sq,nfr,nstop,NDC,NAC,
     1eqp(ltt),ttsum,disum,gtsum,(d1(ii,i),ii=1,5),(a1(ii,j),ii=1,6)
     2,fcval,et(ltt),sc(ltt),sc2sum,op(ltt),edval,ddval,dadsum
      TYPE 3,ca,fl,fls,sq,nfr,nstop,NDC,NAC,
     1eqp(ltt),ttsum,disum,gtsum,(d1(ii,i),ii=1,5),(a1(ii,j),ii=1,6)
C     2,fcval,et(ltt),sc(ltt),sc2sum,op(ltt),edval,ddval,dadsum
10    CONTINUE
      READ(21,2,END=1000)DAPT,AAPT
C     TYPE 2,DAPT,AAPT
40    IF(DAPT-NDC(1))60,50,90
50    IF(AAPT-NAC(1))60,20,90
20    write(20,3)ca,fl,fls,sq,nfr,nstop,NDC,NAC,
     1eqp(ltt),ttsum,disum,gtsum,(d1(ii,i),ii=1,5),(a1(ii,j),ii=1,6)
     2,fcval,et(ltt),sc(ltt),sc2sum,op(ltt),edval,ddval,dadsum
3     format(1x,a2,a5,a1,i2,i2,i2,2(I5,I1),a3
     9,3i5,a4,a3,3a5,a4,a3,a4,a4,a5,a2
     1,i1,a1,a5,i1,a1,2a4,i1)
30    read(1,3,end=1000)ca,fl,fls,sq,nfr,nstop,NDC,NAC,
     1eqp(ltt),ttsum,disum,gtsum,(d1(ii,i),ii=1,5),(a1(ii,j),ii=1,6)
     2,fcval,et(ltt),sc(ltt),sc2sum,op(ltt),edval,ddval,dadsum
      GO TO 40
90    write(22,3)ca,fl,fls,sq,nfr,nstop,NDC,NAC,
     1eqp(ltt),ttsum,disum,gtsum,(d1(ii,i),ii=1,5),(a1(ii,j),ii=1,6)
     2,fcval,et(ltt),sc(ltt),sc2sum,op(ltt),edval,ddval,dadsum
      GO TO 30
60    CONTINUE
C:close(unit=20)
C     TYPE 3,ca,fl,fls,sq,nfr,nstop,NDC,NAC,
C     1eqp(ltt),ttsum,disum,gtsum,(d1(ii,i),ii=1,5),(a1(ii,j),ii=1,6)
C     2,fcval,et(ltt),sc(ltt),sc2sum,op(ltt),edval,ddval,dadsum
      go to 10
1000  continue
      close(unit=1)
      close(unit=20)
      stop
      end
```

EUPUSP.FOR

```
00100          implicit integer(a-z)
00200          parameter mx=1
00300          double precision fo
00400          dimension ndc(2),nac(2),eqp(mx),tt(mx),dis(mx),gt(mx)
00500          1,d1(5,mx),a1(6,mx),fc(mx),et(mx),sc(mx),sc2(mx),
00600          2op(mx),ad(mx),dq(mx),dad(mx)
00700          i=1
00800          j=1
00900          ltt=1
01100          open(unit=1,access='seqin',file='DEPUE',DEVICE='PT49')
01150          REWIND 1
01200          OPEN(UNIT=22,ACCESS='SEQOUT',FILE='LDLUUS.DAT',DEVICE='PT49')
01300          OPEN(UNIT=21,ACCESS='SEQIN',FILE='USEUHP')
01350          OPEN(UNIT=20,ACCESS='SEQOUT',FILE='EUPUSP.DAT',DEVICE='PT49')
01400   2      FORMAT(2I6)
01500          read(1,3,end=1000)ca,fl,fls,sq,nfr,nstop,NDC,NAC,
01600          1eqp(ltt),ttsum,disum,gtsum,(d1(ii,i),ii=1,5),(a1(ii,j),ii=1,6)
01700          2,fcval,et(ltt),sc(ltt),sc2sum,op(ltt),edval,ddval,dadsum
01725          TYPE 3,ca,fl,fls,sq,nfr,nstop,NDC,NAC,
01750          1eqp(ltt),ttsum,disum,gtsum,(d1(ii,i),ii=1,5),(a1(ii,j),ii=1,6)
01775          2,fcval,et(ltt),sc(ltt),sc2sum,op(ltt),edval,ddval,dadsum
01800   10     CONTINUE
01900   C:dapt=d1(2,i)
02000   C:     aapt=a1(2,j)
02100          READ(21,2,END=1000)AAPT,DAPT
02150          TYPE 2,AAPT,DAPT
02400   C      open(unit=20,access='seqout',file=fo,DEVICE='PT49')
02500   40     IF(AAPT-NAC(1))60,50,90
02600   50     IF(DAPT-NDC(1))60,20,90
02700   20     write(20,3)ca,fl,fls,sq,nfr,nstop,NDC,NAC,
02800          1eqp(ltt),ttsum,disum,gtsum,(d1(ii,i),ii=1,5),(a1(ii,j),ii=1,6)
02900          2,fcval,et(ltt),sc(ltt),sc2sum,op(ltt),edval,ddval,dadsum
03000   3      format(1x,a2,a5,a1,i2,i2,i2,2(I5,I1),a3
03100          9,3I5,a4,a3,3a5,a4,a3,a4,a4,a5,a2
03200          1,i1,a1,a5,i1,a1,2a4,i1)
03300   30     read(1,3,end=1000)ca,fl,fls,sq,nfr,nstop,NDC,NAC,
03400          1eqp(ltt),ttsum,disum,gtsum,(d1(ii,i),ii=1,5),(a1(ii,j),ii=1,6)
03500          2,fcval,et(ltt),sc(ltt),sc2sum,op(ltt),edval,ddval,dadsum
03600          GO TO 40
03700   90     write(22,3)ca,fl,fls,sq,nfr,nstop,NDC,NAC,
03800          1eqp(ltt),ttsum,disum,gtsum,(d1(ii,i),ii=1,5),(a1(ii,j),ii=1,6)
03900          2,fcval,et(ltt),sc(ltt),sc2sum,op(ltt),edval,ddval,dadsum
04000          GO TO 30
04100   60     CONTINUE
04112   C:close(unit=20)
04125          TYPE 3,ca,fl,fls,sq,nfr,nstop,NDC,NAC,
04150          1eqp(ltt),ttsum,disum,gtsum,(d1(ii,i),ii=1,5),(a1(ii,j),ii=1,6)
04175          2,fcval,et(ltt),sc(ltt),sc2sum,op(ltt),edval,ddval,dadsum
04200          go to 10
04300   1000   continue
04400          close(unit=1)
04500          close(unit=20)
04600          stop
04700          end
```

UEHNC.FOR

```
      DIMENSION IUP(3),IEP(3),IUC(3)
      OPEN(UNIT=1,ACCESS='SEQIN',FILE='HFILE.DAT')
      OPEN(UNIT=20,ACCESS='SEQOUT',FILE='UEHFIL.DAT')
10    READ(1,1,END=1000)IUP,IAL,IEP,ISEQ,IUC
      IF(IUP(1).EQ.IUC(1))GO TO 10
cx    IF(IUP(1).EQ.JUP.AND.IUC(1).EQ.JUC)GO TO 10
1     FORMAT(I6,I2,I4,1X,A2,I6,I2,I4,I9,I6,I2,I4)
      WRITE(20,2)IUP(1),IEP(1),IUC(1)
2     FORMAT(1X,2I5,'U',I5)
      GO TO 10
1000  CLOSE(UNIT=1)
      CALL EHFIL
      OPEN(UNIT=20,FILE='UEHFIL.DAT')
      CALL SORT('UEHFLS.DAT=UEHFIL.DAT/REC:17/KEY:2:16/ALPHA')
      CLOSE(UNIT=20)
3000  CALL UEHFSR
      STOP
      END
      SUBROUTINE EHFIL
      DIMENSION IUP(3),IEP(3),IUC(3)
      OPEN(UNIT=1,ACCESS='SEQIN',FILE='HFILE2.DAT')
10    READ(1,1,END=1000)IUP,IAL,IEP,ISEQ,IUC
      IF(IEP(1).EQ.IUC(1))GO TO 10
cx    IF(IEP(1).EQ.JUP.AND.IUC(1).EQ.JUC)GO TO 10
1     FORMAT(I6,I2,I4,1X,A2,I6,I2,I4,I9,I6,I2,I4)
      WRITE(20,2)IUP(1),IEP(1),IUC(1)
2     FORMAT(1X,2I5,'E',I5)
      GO TO 10
1000  CLOSE(UNIT=1)
      CLOSE(UNIT=20)
      RETURN
      END
      SUBROUTINE UEHFSR
      DIMENSION IR(4),JR(4)
      DATA JR/4*' '/
      OPEN(UNIT=21,ACCESS='SEQIN',FILE='UEHFLS.DAT')
      OPEN(UNIT=22,ACCESS='SEQOUT',FILE='UEHnc.DAT')
10    READ(21,1,END=20)IR
1     FORMAT(1X,2A5,A1,A5)
      DO 30 J=1,4
      IF(IR(J).NE.JR(J))GO TO 40
30    CONTINUE
      GO TO 10
40    DO 50 J=1,4
50    JR(J)=IR(J)
      WRITE(22,1)JR
      GO TO 10
20    CLOSE(UNIT=21)
      CLOSE(UNIT=22)
      RETURN
      END
```

SCENAR.FOR

```
      implicit integer(a-z)
      parameter mx=1
      double precision fo
      DIMENSION CN(44),NDCI(2),NACJ(2)
      IGO=1
      open(unit=1,access='seqin',file='UsFN')
      REWIND 1
      OPEN(UNIT=20,ACCESS='SEQIN',FILE='UEHnc')
      OPEN(UNIT=22,ACCESS='SEQTUT',FILE='IFCNUE.DAT'
     1,device='PT49')
10    CONTINUE
20    read(1,3,end=1500)
     9ca,fl,fls,sq,nfr,nstop,ndci,nacj,
     1eqpltt,ttsum,disum,gtsum,dwac,dap,dlt,dgt,dfr,awac
     2,aap,alt,agt,afr
     2,fcval,etltt,scltt,sc2sum,opltt,edval,ddval,dadsum
C:TYPE 3,
C:    9ca,fl,fls,sq,nfr,nstop,ndci,nacj,
C:1eqpltt,ttsum,disum,gtsum,dwac,dap,dlt,dgt,dfr,awac
3     format(1x,a2,a5,a1,i2,i2,i2,2(I5,I1),a3
     9,3i5,a4,a3,2A4,I7,a4,a3,2A4,I7
     1,i1,a1,a5,i1,a1,2a4,i1)
      IF(IGO)100,1000,140
100   CALL PPCN(NACJ,NDCI,CA,FL,FLS,DLT,DFR,ALT,AFR,IGO)
      GO TO 10
140   CALL PPCN(NDCI,NACJ,CA,FL,FLS,DLT,DFR,ALT,AFR,IGO)
      GO TO 10
1500  close(unit=1)
      IF(IGO)1000,1000,1400
1400  IGO=-1
      open(unit=1,access='seqin',file='EuUS')
      go to 10
1000  continue
      close(unit=22)
      close(unit=20)
      OPEN(UNIT=22,FILE='IFCNUE.DAT')
     call sort(
    1'PT49:SICNUE.DAT=IFCNUE.DAT/RECORD:34/KEY:10:7/KEY:19:4/KEY:2:8'
      call rcnue
      call pncon
      stop
      end
      SUBROUTINE PPCN(NCUS,NCEU,CAIF,IFLT,IFLTS,IFTIMD,IFRD,
     1IFTIMA,IFRA,IGO)
      implicit integer(a-z)
      parameter mx=1
      double precision fo
      dimension CN(5),NCUS(2),NCEU(2)
      DATA IF20/-1/
      IF(IF20.EQ.-1)READ(20,3,END=1500)CN
      IF20=0
210   IF(NCUS(1)-CN(2))230,250,20
250   IF(NCEU(1)-CN(3))230,240,20
230   REWIND 20
20    read(20,3,end=1500)CN
3     FORMAT(A1,2I5,A1,I5)
310   IF(NCUS(1)-CN(2))1000,350,20
350   IF(NCEU(1)-CN(3))1000,240,20
240   CONTINUE
      NCUN=CN(5)
      UE=CN(4)
```

```
06600            IF(UE.EQ.'E')GO TO 10
06700            NCIF=NCUS(1)
06800            APTIF=NCUS(2)
06900            JGO=IGO
07000      10    IF(JGO)30,2000,60
07100      30    AD='PD'
07200            IFTIME=IFTIMA
07300            IFR=IFRA
07400            GO TO 50
07500      40    AD='CP'
07600            IFTIME=IFTIMD
07700            IFR=IFRD
07800      50    WRITE(22,1)CAIF,IFLT,IFLTS,AD,UE,NCIF,APTIF,NCCN,IFTIME,IFR
07900      C:TYPE 1,CAIF,IFLT,IFLTS,AD,UE,NCIF,APTIF,NCCN,IFTIME,IFR
08000      1     FORMAT(1X,A2,A5,A1,A2,A1,I5,I1,I5,A4,I7)
08100            GO TO 20
08200      60    NCIF=NCEU(1)
08300            APTIF=NCEU(2)
08400            JGO=-IGO
08500            GO TO 60
08600      2000  INIT=10
08700            STOP 10
08800      1500  REWIND 20
08900            IF20=-1
09000      1000  continue
09100            RETURN
09200            end
09300            subroutine rcnue
09400            DIMENSION l(7),m(7)
09500            open(unit=20,access='seqout',file='rcncue',device='PT49')
09600            OPEN(UNIT=1,ACCESS='SEQIN',FILE='SICNUE',DEVICE='PT49')
09700            data m/7*' '/
09800      10    read(1,1,end=20)l
09900      1     format(6a5,a4)
10000            do 30 i=1,7
10100            if(l(i).ne.m(i))go to 40
10200      30    continue
10300            go to 10
10400      40    do 50 i=1,7
10450      50    m(i)=l(i)
10500            write(20,1)l
10600            go to 10
10700      20    close(unit=1)
10800            close(unit=20)
10900            return
11000            end
11100            subroutine pncon
11200            IMPLICIT INTEGER(A-Z)
11300            parameter nm=99
11400          DIMENSION IFP(7,nm),caif(nm),iflt(nm),iflts(nm),aptif(nm)
11500            1,iftime(nm)
11600            OPEN(UNIT=1,ACCESS='SEQIN',FILE='RcncUN.DAT',DEVICE='PT49')
11700            OPEN(UNIT=21,ACCESS='SEQOUT',FILE='CPDCON',DEVICE='PT49')
11800      60    READ(1,1,END=20)CAIF(1),IFLT(1),IFLTS(1),AD,UE,NCIF,APTIF(1)
11900            1,NCCN,IFTIME(1),(IFP(j,1),j=1,7)
12000      10    do 30 i=2,nm
12100            READ(1,1,END=20)CAIF(i),IFLT(i),IFLTS(i),ADn,UEn,NCIFn,APTIF(i
12200            1,NCCNn,IFTIME(i),(IFP(j,i),j=1,7)
12300            if(adn.ne.ad.or.uen.ne.ue.or.ncifn.ne.ncif.or.nccnn.ne.nccn
12400            1) go to 40
12500      30    continue
```

```
11900              it=0
12000              ix=nm
12100      [illegible]  MAT=ALLOC(7,MX)
12200              [illegible] 2,MX,II,NCIF,NCCN,(CAIF(J),IFLT(J),IFLTS(J),J=1,MX7)
12300      2       FORMAT(1X,I2,I3,2I3,7(A2,A5,A1))
12400              CALL S1CON(CAIF,IFLT,IFLTS,AD,UE,NCIF,APTIF,NCCN,IFTIME,IFR,mx)
12500              [illegible] (it) 80,80,90
12600      [illegible]  caif(1)=caif(it)
12700              iflt(1)=iflt(it)
12800              iflts(1)=iflts(it)
12900              aptif(1)=aptif(it)
13000              iftime(1)=iftime(it)
13100              do 70 i=1,7
13200              ifr(i,1)=ifr(i,it)
13300      70      continue
13400              ad=adn
13500              ue=uen
13600              ncif=ncifn
13700              nccn=nccnn
13800              GO TO 10
13900      40      mx=i-1
14000              it=i
14100              go to 50
14200      1       FORMAT(1X,A2,A5,A1,A2,A1,I5,I1,I3,I4,7I1)
14300      20      CLOSE(UNIT=1)
14400              it=-1
14500              mx=i-1
14600              go to 50
14700      80      continue
14800              CLOSE(UNIT=21)
14900              OPEN(UNIT=21,FILE='OPDCON',DEVICE='PT49')
15000              CALL SORT(
15100          1'scenar.DAT=PT49:OPDCON.DAT/RECORD:57/KEY:2:8')
15200              STOP
15300              END
15400              SUBROUTINE S1CON(CAIF,IFLT,IFLTS,AD,UE,NCIF,APTIF,NCCN,IFTIME
15500              1,IFR,mx)
15600              IMPLICIT INTEGER(A-Z)
15700              parameter nm= 99
15800              DOUBLE PRECISION FO
15900              DIMENSION IFR(7,mx),APT(3,nm),OPTIME(3,nm),caif(mx),iflt(mx)
16000              9,ontime(nm),iflts(mx),ttsum(nm),disum(nm),ma(nm),eqpltt(nm)
16100              1,aptif(mx),iftime(mx),nstop(nm),ca(nm),dapt(nm),aapt(nm)
16200              2,actgt(nm),sc2sum(nm),gtsum(nm),CAPT(NM)
16300              DATA OPD/' '/
16400              do 110 i=1,mx
16500      110     APT(1,i)=APTIF(i)
16600              ENCODE(5,3,OPDUE)AD,UE
16700      3       FORMAT(A2,A1)
16800              IF(OPD.EQ.OPDUE)GO TO 200
16900              ENCODE(10,4,FO)OPDUE,OPDUE
17000      4       FORMAT(A3,A3,'.DAT')
17100              OPEN(UNIT=20,ACCESS='SEQIN',FILE=FO)
17200              OPD=OPDUE
17300      200     CONTINUE
17400              IF(AD.EQ.'PD')GO TO 120
17500              [illegible]=NCIF
17600              [illegible]=NCCN
17700              [illegible] TO 50
17800      120     [illegible]=NCCN
17900              [illegible]=NCIF
```

```
70      CONTINUE
        IF(UL.EQ.'T')GO TO 40
        do 130 i=1,mx
130     CALL CONNECT(NCIF,APT(1,i),AD,NAPT,ONTIME(i),OFTIME(1,i))
        GO TO 150
40      CONTINUE
        do 140 i=1,mx
140     CALL CONNEC(NCIF,APT(1,i),AD,NAPT,ONTIME(i),OFTIME(1,i))
150     CONTINUE
C:TYPE*,(ONTIME(I),I=1,MX),NAPT,((OFTIME(I,J)
C:1,APT(I,J),I=1,NAPT),J=1,MX)
        call onecon(dapt,aapt,ad,ue,caif,iftime,oftime,ontime,IFR,
        1ca,nstop,eqpltt,ttsum,disum,gtsum,SC2SUM,actgt,NDCI,NACJ
        2,NAPT,APT,MA,mx,CAPT)
        do 10 k=1,mx
      IF(NSTOP(K).LT.0)TYPE5,CAIF(k),IFLT(k),IFLTS(k),NCIF,APT(1,K),NCC
        1,CAPT(K)
C:TYPE 5,CAIF(k),IFLT(k),IFLTS(k),NCIF,NCCN,TTSUM(k),DISUM(k)
C:9,GT,MA(k)
C:1,EQPLTT(k),NSTOP(k),CA(k),DAPT(k),AAPT(k),ACTGT(k),SC2SUM(k)
        IF(NSTOP(k).LT.0)GO TO 10
        GT=GTSUM(k)+ACTGT(k)
        WRITE(21,5)CAIF(k),IFLT(k),IFLTS(k),NCIF,APT(1,K),NCCN,CAPT(K)
        8,TTSUM(k),DISUM(k)
        9,GT,MA(k)
        1,EQPLTT(k),NSTOP(k),CA(k),DAPT(k),AAPT(k),ACTGT(k),SC2SUM(k)
5       FORMAT(1X,A2,A5,A1,2(I5,I1),3I5,I1,A3,I2,A2,2A3,I5,I2)
10      continue
        RETURN
        end
        SUBROUTINE CONNEC(NCITY,NAP,OPD,NAPTS,ONL,OFL)
        IMPLICIT INTEGER(A-Z)
        PARAMETER NO=40
        DIMENSION APT(NO),CITY(NO),ON(2,NO),AP(3,NO),OF(3,2,NO),NAPT(NO
        DIMENSION OFL(3),NAP(3)
        DATA INIT/0/
        IF(INIT.NE.0)GO TO 200
        INIT=1
        OPEN(UNIT=23,ACCESS='SEQIN',FILE='CONNEC')
        DO 10I=1,NO
      READ(23,1,END=30)APT(I),CITY(I),AP(1,I),(OF(1,J,I),ON(J,I),J=1,2
1       FORMAT(A3,I5,I1,4I4)
        DO 70 J=2,3
70      AP(J,I)=-1
        NAPT(I)=1
        IF(OF(1,1,I).EQ.0)GO  TO 20
10      CONTINUE
30      TYPE *,CITY,ON
        STOP
20      MAX=I-1
90      READ(23,2,END=100)A1,C1,AP1,A2,C2,AP2,PD,OP
        IF(A2.EQ.' ')GO TO 90
2       FORMAT(2(A3,I5,I1),2I4)
        IF(C1.EQ.C2)GO TO 90
        TYPE*,C1,AP1,C2,AP2,PD,OP
        GO TO 90
        DO 50 I=1,MAX
        IF(C1.NE.CITY(I))GO TO 50
        IF(AP1.EQ.AP(1,I))GO TO 50
        IF(AP2.NE.AP(1,I))GO TO 50
        ANN=AP1
```

```
38600              30 CF 50
38700      60      AP1=AP2
38800              DO 110 J=2,3
38900              IF(AP(J,I).GT.-1)GO TO 110
39000              AP(J,I)=AP1
39100              CF(J,1,I)=PD
39200              CF(J,2,I)=OP
39300              NAPT(I)=J
39400              GO TO 50
39500      110     CONTINUE
39600              TYPE*,J,I,AP1,AP2,AP
39700              STOP
39800      50      CONTINUE
39900              GO TO 80
40000      100     CONTINUE
40100              CLOSE(UNIT=23)
40200      200     DO 120 I=1,MAX
40300              IF(CITY(I).EQ.NCITY.AND.AP(1,I).EQ.NAP(1))GO TO 210
40400      120     CONTINUE
40500              NAPTS=1
40600              ONL=100
40700              CFL(1)=100
40800              RETURN
40900      210     NAPTS=NAPT(I)
41000              J=2
41100              IF(OPD.EQ.'PD')J=1
41200              ONL=ON(J,I)
41300              DO 220 K=1,NAPTS
41400              NAP(K)=AP(K,I)
41500      220     CFL(K)=CF(K,J,I)
41600              END
41700              SUBROUTINE CONNECT(NCITY,NAP,OPD,NAPTS,ONL,CFL)
41800              IMPLICIT INTEGER(A-Z)
41900              PARAMETER NO=40
42000              DIMENSION APT(NO),CITY(NO),ON(2,NO),AP(3,NO),CF(3,2,NO),NAPT(NO
42100              DIMENSION CFL(3),NAP(3)
42200              DATA INIT/0/
42300              IF(INIT.NE.0)GO TO 200
42400              INIT=1
42500              OPEN(UNIT=23,ACCESS='SEQIN',FILE='CONNECT')
42600              DO 10I=1,NO
42700            READ(23,1,END=30)APT(I),CITY(I),AP(1,I),(CF(1,J,I),ON(J,I),J=1,2)
42800      1       FORMAT(A3,I5,I1,4I4)
42900              DO 70 J=2,3
43000      70      AP(J,I)=-1
43100              NAPT(I)=1
43200              IF(CF(1,1,I).EQ.0)GO  TO 20
43300      10      CONTINUE
43400      30      TYPE *,CITY,ON
43500              STOP
43600      20      MAX=I-1
43700      80      READ(23,2,END=100)A1,C1,AP1,A2,C2,AP2,PD,OP
43800              IF(A2.EQ.' ')GO TO 80
43900      2       FORMAT(2(A3,I5,I1),2I4)
44000              IF(C1.EQ.C2)GO TO 90
44100              TYPE*,C1,AP1,C2,AP2,PD,OP
44200              GO TO 90
44300              DO 50 I=1,MAX
44400              IF(C1.NE.CITY(I))GO TO 50
44500              IF(AP1.EQ.AP(1,I))GO TO 60
44600              IF(AP2.EQ.AP(1,I))GO TO 50
```

```
      APN=AP1
      GO TO 90
   50 NPN=NPN
      DO 110 J=2,3
      IF(AP(J,I).NE.-1)GO TO 110
      AP(J,I)=APN
      OP(J,1,I)=PD
      OP(J,2,I)=OP
      NAPT(I)=J
      GO TO 50
  110 CONTINUE
      TYPE*,J,I,AP1,AP2,AP
      STOP
   80 CONTINUE
      GO TO 80
  100 CONTINUE
      CLOSE(UNIT=23)
  200 DO 120 I=1,MAX
      IF(CITY(I).EQ.NCITY.AND.AP(1,I).EQ.NAP(1))GO TO 210
  120 CONTINUE
      NAPTS=1
      CNL=100
      OFL(1)=100
      RETURN
  210 NAPTS=NAPT(I)
      J=2
      IF(OPD.EQ.'PD')J=1
      CNL=CN(J,I)
      DO 220 K=1,NAPTS
      NAP(K)=AP(K,I)
  220 OFL(K)=OF(K,J,I)
      END
      subroutine onecon(dapt,aapt,ad,ue,caif,iftime,oftime,ontime,IFR
     1,ca,nstop,eqpltt,ttsum,disum,gtsum,SC2SUM,actgt,NDCI,NACJ
     2,NAPT,APT,MAS,mx,CAPT)
      implicit integer(a-z)
      parameter nm=99
      double precision fo
      dimension CN(44),OFT(3,nm),CFTIME(3,mx),APT(3,mx),init(nm),
     2dfr(7),afr(7),IFR(7,mx),caif(mx),iftime(mx),ontime(mx)
     3,ca(mx),nstop(mx),eqpltt(mx),ttsum(mx),disum(mx),gtsum(mx)
     4,sc2sum(mx),actgt(mx),mas(mx),mift(nm),ont(nm),of210(nm)
     5,nifr(nm),dapt(mx),aapt(mx),conalt(nm),CAPT(MX)
      DATA IF20/-1/
      ihmtm(i)=(i/100)*60+mod(i,100)
      mthm(i)=(i/60)*100+mod(i,60)
      SGN=-1
      if(ad.eq.'PD')SGN=1
      do 10 k=1,mx
      nifr(k)=0
      do 410 j=1,7
  410 nifr(k)=nifr(k)+ifr(j,k)
      mift(k)=ihmtm(iftime(k))
      ont(k)=mthm(mift(k)+ihmtm(ontime(k))*SGN)
      IF210(k)=MTHM(MIFT(k)+210*SGN)
      init(k)=0
      nstop(k)=-1
      DO 10 I=1,NAPT
   10 OFT(I,k)=mthm(mift(k)+ihmtm(oftime(I,k))*SGN)
      CFTIME=CFTIME*,IFTIME,OFTIME,CNTIME,NIFT,OFT,INT
      WRITE*,(IFR(K),K=1,MX)
```

```
50700              IF(IF20.EQ.-1)READ(20,3,END=150)CN
50800              IF20=0
50900      100     IF(AD.EQ.'PD')GO TO 210
51000              IF(NACJ-CN(9))230,220,420
51100      220     IF(NDCI-CN(7))230,240,420
51200      210     IF(NDCI-CN(7))230,250,420
51300      230     IF(NACJ-CN(9))230,240,420
51400      230     REWIND 20
51500      420     read(20,3,end=1500)CN
51600      C(1):cn,fl,fls,sg,nfr,nstop,ndci,nacj,
51700      C(11):  1epltt,ttsum,disum,gtsum,dwac,dap,dlt,dgt,dfr,awac
51800      C(27):  2,aap,alt,agt,afr
51900      C(37):  2,fcval,etltt,scltt,sc2sum,opltt,edval,ddval,dadsum
52000      3       format(1x,a2,a5,a1,i2,i2,i2,2(I5,I1),a3
52100              9,3i5,a4,a3,2I4,7i1,a4,a3,2I4,7i1
52200              1,i1,a1,a5,i1,a1,2a4,i1)
52300      C:TYPE 3,CN
52400              IF(AD.EQ.'PD')GO TO 310
52500              IF(NACJ-CN(9))1000,320,420
52600      320     IF(NDCI-CN(7))1000,240,420
52700      310     IF(NDCI-CN(7))1000,350,420
52800      350     IF(NACJ-CN(9))1000,240,420
52825      240     IF(CN(17).LT.CN(28).OR.CN(44).GT.0)GOTO 260
52850              TYPE 5
52875      5       FORMAT(' ARRIVAL TIME PRECEDES DEPARTURE TIME:')
52900              TYPE 3,CN
52925              GOTO 420
52950      260     do 20 k=1,mx
53000              IF(INIT(k).EQ.10)GOTO 20
53100      c       IF(CN(5).LT.NIFR(k))GO TO 20
53200      c-frequency check
53300              APTC=CN(9-SGN)
53400              APTOD=CN(9+SGN)
53500              MA=0
53600              DO 30 I=1,NAPT
53700              IF(APTC.EQ.APT(I,k))GO TO 40
53800      30      MA=1
53900              TYPE 4,APTC,AD,UE,CAIF(k),IFTIME(k),(IFR(i,k),i=1,7),(APT(i,k)
54000              1,i=1,napt)
54100      4       FORMAT('OUNK A/P:',I1,', 210 MIN'
54200              1,' USED.',A2,A1,A2,I4,7I1,3I8)
54300              TYPE 3,CN
54400              GT=OF210(k)
54500              GO TO 50
54600      40      gt=oft(I,k)
54700              if(I.EQ.1.AND.caif(k).eq.cN(1))gt=ont(k)
54800      C:TYPE *,GT
54900      C,(CN(J),J=4,10),(CN(J),J=12,14),(CN(J),J=17,25)
55000      C       1,(CN(J),J=28,37),CN(40)
55100      50      CONTINUE
55200              if(ad.eq.'PD')go to 100
55300              PT=CN(28)
55400              CT=CN(17)
55500              if(gt.lt.PT)go to 10
55600      110     KB=0
55700      70      IF(SGN)120,20,130
55800      130     IF(CHECKD(IFR(1,k),CN(30),KB).GT.0)GO TO 20
55900              GO TO 140
56000      120     if(Check3(ifr(1,k),CN(39),KB).gt.0)go to 20
56100              IF(INIT(k).EQ.2)GO TO 20
56200              INIT(k)=1
```

```
            IF(AP.EQ.'PD')GO TO 190
            IF(GT.GE.PT)INIT(k)=2
   90       CA(k)=CN(1)
            NSTOP(k)=CN(6)
            CAPT(k)=APTOD
            EQPLTT(k)=CN(11)
            DISUM(k)=CN(12)
            DISUM(k)=CN(13)
            GTSUM(k)=CN(14)
            DAPT(k)=CN(16)
            AAPT(k)=CN(27)
            SC2SUM(k)=CN(40)
            MAS(k)=MA
            ACTGT(k)=(XD*1440+IHMTM(PT)-MIFT(k))*SGN
   CT:TYPE *,MAS,ATF,XD,PT,CT,INIT
            IF(INIT(k).EQ.1)GO TO 20
            if(caif(k).eq.ca(k))go to 2000
            CONALT(k)=MTHM(IHMTM(CT)+SGN*120)
   c        120 minute on line connection
            GO TO 20
   190      IF(GT.LE.PT)INIT(k)=2
            GO TO 90
   C:LOOK for best online within 30 minutes
   90       CONTINUE
   CTC:TYPE *,GT,(CN(J),J=4,10),(CN(J),J=12,14),(CN(J),J=17,25)
   CT       1,(CN(J),J=28,37),CN(40)
            IF(CN(17).LT.CONALT(k))GO TO 2000
            IF(CN(1).NE.CAIF(k))GO TO 20
            GO TO 90
   60       IF(INIT(k).GT.0)GO TO 20
            XD=SGN
            GO TO 70
   100      PT=CN(17)
            CT=CN(28)
            if(gt.Gt.PT)go to 60
            GO TO 110
   2000     INIT(k)=10
            GO TO 20
   20       continue
            go to 420
   1500     REWIND 20
            IF20=-1
   1000     continue
            RETURN
            end
            function kheckd(ifr,fr,XD)
            implicit integer(a-z)
            dimension ifr(7),fr(7)
            kheckd=1
            do 10 i=1,7
            J=XD+I
            IF(J.LT.1)J=J+7
            IF(J.GT.7)J=J-7
            if(ifr(i).eq.1.and.fr(J).eq.1)kheckd=0
   10       continue
   CC:TYPE *,IFR,FR,KHECKD
            return
            end
```

```
00100          IMPLICIT INTEGER(A-Z)
00200          PARAMETER NACTY=23,NOAC=16,NWRITE=135000,thresh=.02
00300          INTEGER BB
00400          REAL*8 PRECISION NAME,NAME,LBLM
00500          REAL IDPF,QMK,QST,QTT,QMA,QAC,EACT,ENOR,FLCS,IACT,IDPF
00600          COMMON/C1/USPT(31),EUPT(31),T1(31),U1(31),E1(31),T2(31)
00700          COMMON/C2/HDTA(5),HDTA1(10,6),JDTA1(10,6)
00800          COMMON/C3/CYCD(10,2),CYFLG(10),ACTY(NACTY),CDPN(25,2),IDPF(2)
00900          COMMON/C4/QMK(8),QST(8),QTT(4),QMA(2),QAC(NOAC),CARTY(5),PIDR(64)
01000          COMMON/C5/CTFL(16),DTFL(16),LBL(10),LTLCNT(10),PDR(64),IDPF(2)
01100          COMMON/C6/DIST(11),TTIME(11),GTIME(10)
01200          COMMON/C7/CTY(6),CTYFG(10),CFG(25),CANC(2)
01300          COMMON/C8/LSTOP(25),CONCT(10),ONCT(10),ONFL(10)
01400          DIMENSION WLGC(2,2),WLG(2),IDFLG(10),JACT(10),NAME(10)
01500          DATA ITY/5/,BB/'  '/
01600          DATA LBL/11,12,13,14,15,23,24,25,26,27/
01700          DATA NAME/'C1D.DAT','C2D.DAT','C3D.DAT','C4D.DAT','C5D.DAT',
01800     X    'C6D.DAT','C7D.DAT','C8D.DAT','C9D.DAT','C10D.DAT'/
01900          DATA QMK/1.0,0.8,0.6,0.115,0.09,0.071,0.053,0.027/
02000          DATA QST/1.0,0.28,0.17,0.09,0.04,0.028,0.020,0.0130/
02100          DATA QTT,QMA/-0.002,-0.007,-0.024,-0.001/
02200          DATA ACTY/'747','L10','D10','707','DC8','SSC','D1C','VCX',
02300     X    'D9S','74L','74B','D8V','L16','D8A','74M',
02400     X    'D1M','70M','703','D8L'
02500          1,'72S','DC9','727'/
02600          DATA QAC/1.7,1.43,1.43,1.00,1.00,1.00,1.43,1.00,
02700     X    1.0,1.43,1.7,1.0,1.0,1.0,1.7
02800          1,1.43,1.0,1.0,1.0/
02900          DATA CARTY/'PA','TW','NA','DL','BN'/
03000          DATA CTY/5850,63761,51122,18502,51412,66314/
03100          DATA CANC/'AC','CP'/
03200          OPEN(UNIT=21,DEVICE='DSK',FILE='PBTNM.DAT',ACCESS='SEQIN')
03300  C
03400  C
03500  C.....READ IN CITY CODES
03600  C
03700  C
03800          DO 1 I=1,31
03900          READ(21,2)U1(I),E1(I)
04000   2      FORMAT(5X,I5,5X,I5)
04100   1      CONTINUE
04200          CLOSE(UNIT=21)
04300  C
04400  C
04500  C.....OPEN FILES
04600  C
04700  C
04800          OPEN(UNIT=9,DEVICE='DSK',FILE='WATERS.LOG',MODE='IMAGE',
04900     X    ACCESS='SEQOUT')
05000          OPEN(UNIT=20,DEVICE='DSK',FILE='UEDSC.DAT',ACCESS='SEQIN')
05100          OPEN(UNIT=21,DEVICE='DSK',FILE='SCENAR.DAT',ACCESS='SEQIN')
05200          OPEN(UNIT=22,DEVICE='DSKB',FILE='SCEDRT.DAT',ACCESS='SEQOUT',
05300          NWRITE=0
05400  C
05500  C
05600  C.....READ INTERNATIONAL FLIGHT
05700  C
05800  C
05900          ANDC=0
06000          NLFTB=0
06100          READ(20,3,END=100)(HDTA(I),I=1,4)
```

```
05900     7       FORMAT(1X,A2,I6,I2,I2)
06000   DO 7 I=1,NDTA
06100             I1=NDTA(1)
06200             I2=NDTA(2)
06300             I3=NDTA(3)
06400     7       J4=NDTA(4)
06500             CONFLG=0
06600             DO 6 K=1,J4
06700     6       READ(20,7)(HDTA1(K,I),I=1,6)
06800     7       FORMAT(1X,2(I5,I1),A3,3I5)
06900             DO 8 I=1,10
07000             CYCD(I,1)=0
07100             CYCD(I,2)=0
07200     8       CYFLG(I)=0
07300             DO 9 I=1,J4
07400             CYCD(I,1)=HDTA1(I,1)
07500     9       CYCD(I,2)=HDTA1(I,2)
07600             J5=J4+1
07700             CYCD(J5,1)=HDTA1(J4,3)
07800             CYCD(J5,2)=HDTA1(J4,4)
07900             I=0
08000     3003    I=I+1
08100   C
08200   C
08300   C.....CHECK FOR U.S. OR FOREIGN ORIGIN
08400   C
08500   C
08600             DO 3000 I=1,31
08700             IF(CYCD(I,1).EQ.U1(I))GO TO 3001
08800             IF(CYCD(I,1).EQ.E1(I))GO TO 3002
08900     3000    CONTINUE
09000             GO TO 3003
09100     3001    USEPT=1
09200             DO 3005 I=1,31
09300             EUPT(I)=E1(I)
09400     3005    USPT(I)=U1(I)
09500             GO TO 3007
09600     3002    USEPT=2
09700             DO 3006 I=1,31
09800             USPT(I)=E1(I)
09900     3006    EUPT(I)=U1(I)
10000     3007    CONTINUE
10100   C
10200   C
10300   C.....CHECK FOR WATERWAY SEGMENT AND CITIES NOT IN MODEL
10400   C
10500   C
10600             IFLG=0
10700             I2FLG=0
10800             DO 10 I=1,J5
10900             DO 11 J=1,31
11000             IF(CYCD(I,1).EQ.USPT(J))GO TO 12
11100     11      CONTINUE
11200             DO 13 K=1,31
11300             IF(CYCD(I,1).EQ.EUPT(K).AND.IFLG.EQ.0)GO TO 13
11400             IF(CYCD(I,1).EQ.EUPT(K).AND.IFLG.EQ.1)GO TO 14
11500     13      CONTINUE
11600             CYFLG(I)=1
11700             I2FLG=1
11800             GO TO 10
11900     12      I1=I-1
```

```
      IF(CYFLG(I1).EQ.4)CYFLG(I1)=5
      IF(CYFLG(I1).EQ.5)GO TO 900
130   IF(CYFLG(I1).NE.4)CYFLG(I1)=1
      CYFLG(I)=2
      WLG(1)=I1
      WLG(2)=I
      WLGC(1,1)=CYCD(I1,1)
      WLGC(1,2)=CYCD(I1,2)
      WLGC(2,1)=CYCD(I,1)
      WLGC(2,2)=CYCD(I,2)
      IFLG=1
      GO TO 10
14    CYFLG(I)=3
      GO TO 10
15    CYFLG(I)=0
10    CONTINUE
      IF(CYFLG(1).EQ.4)CYFLG(1)=6
      IF(CYFLG(J5).EQ.4)CYFLG(J5)=6
      NUSP=WLG(1)
      NEUP=WLG(2)
c:type*,nusp,neup
      DO 16 I=1,10
16    IDFLG(I)=0
      IF(I2FLG.EQ.0)GO TO 20
      DO 17 I=1,J4
      IF(I.NE.1)GO TO 18
      IF(CYFLG(1).EQ.0.AND.CYFLG(2).EQ.4)IDFLG(1)=1
      IF(CYFLG(1).EQ.6)IDFLG(1)=6
      GO TO 17
18    IF(I.NE.J4)GO TO 19
      I1=J4+1
      IF(CYFLG(I).EQ.4.AND.CYFLG(I1).LE.3)IDFLG(I)=9
      IF(CYFLG(I1).EQ.6)IDFLG(I1)=6
      GO TO 17
19    I2=I+1
      IF(CYFLG(I).LE.3.AND.CYFLG(I2).LE.3)IDFLG(I)=0
      IF(CYFLG(I).EQ.4.AND.CYFLG(I2).LE.3)IDFLG(I)=9
      IF(CYFLG(I).LE.3.AND.CYFLG(I2).EQ.4)IDFLG(I)=1
      IF(CYFLG(I).EQ.4.AND.CYFLG(I2).EQ.4)IDFLG(I)=9
17    CONTINUE
      I1FLG=0
      IF(IDFLG(J5).NE.6)GO TO 20
      J3=0
      IF(IDFLG(J5).EQ.6)I1FLG=1
25    J3=J3+1
      J2=J5-J2
      IF(IDFLG(J2).NE.9)GO TO 24
      I1FLG=I1FLG+1
      GO TO 25
24    J4=J4-I1FLG
      IDFLG(J4+1)=0
20    IAC=HDTA1(NUSP,5)
      J8=0
      I5=1
      J=1
      DO 201 J10=1,9
201   JACT(J10)=0
      DO 210 J10=1,11
      LIST(J10)=0
210   TTIMT(J10)=0
      DO 211 J10=1,10
```

```
  211     GTIME(J10)=0
C
C
C.....RECALCULATE INTERNATIONAL SEGMENTS
C
C
          I5=1
          DO 21 I=1,J4
          IF(I.EQ.2.AND.IDFLG(1).EQ.6.AND.IDFLG(2).EQ.9)GO TO 219
          IF(I.EQ.1.AND.IDFLG(1).EQ.6)GO TO 219
          IF(IDFLG(I5).NE.0)GO TO 22
          IF(NUSP.EQ.I5)NUSP1=J
c:type*,nusp,i5,nusp1,j,i
          DO 23 K=1,4
   23     JDTA1(J,K)=HDTA1(I5,K)
          JDTA1(J,5)=0
          IF(HDTA1(I5,5).NE.IAC)JDTA1(J,5)=1
          JDTA1(J,6)=HDTA1(I5,7)
          JDTA1(J,7)=0
          JDTA1(J,8)=0
          IF(HDTA1(I5,8).GE.120)JDTA1(J,7)=1
          ITIME=HDTA1(I5,6)
          JTIME=HDTA1(I5,8)
          GO TO 29
   22     J1FLG=0
          DO 26 I1=1,10
          I2=I5+I1
          IF(IDFLG(I2).NE.9)GO TO 27
          J1FLG=J1FLG+1
   26     CONTINUE
   27     JDTA1(J,1)=HDTA1(I5,1)
          JDTA1(J,2)=HDTA1(I5,2)
          I2=I5+J1FLG
          JDTA1(J,3)=HDTA1(I2,3)
          JDTA1(J,4)=HDTA1(I2,4)
          ITIME=0
          JTIME=0
          IDIST=0
          JDTA1(J,5)=0
          JDTA1(J,7)=0
          DO 28 I3=I5,I2
          IDIST=IDIST+HDTA1(I3,7)
          ITIME=ITIME+HDTA1(I3,6)
          JTIME=JTIME+HDTA1(I3,8)
          IF(HDTA1(I3,5).NE.IAC)JDTA1(J,5)=1
   28     IF(HDTA1(I3,8).GE.120)JDTA1(J,7)=1
          JDTA1(J,6)=IDIST
          JDTA1(J,8)=J1FLG
          I5=I5+J1FLG
          I6=J4-J1FLG
   29     CONTINUE
          J8=J8+1
          GTIME(J8)=JTIME
          DIST(J8)=JDTA1(J,6)
          ITIME(J8)=ITIME
          DO 202 J9=1,NACTY
          IF(ACTY(J9).EQ.HDTA1(I5,5))JACT(J)=J9
  202     CONTINUE
          J=J+1
          IF(I6.EQ.I)GO TO 2110
  219     I5=I5+1
```

```
23900     21      CONTINUE
23900     2110    CONTINUE
24000   C
24100   C
24200   C.....CHECK FOR ONLINE CONNECTION ON INTERNATIONAL FLIGHT
24300   C.....DETERMINE NUMBER OF STOPS
24400   C
24500   C
24600             DO 4011 I=1,10
24700             ONFL(I)=0
24800             CONCT(I)=0
24900     4011    ONCT(I)=0
25000             TAFLG=0
25100             BFLG=0
25200             AFLG=0
25300             J6=J-1
25400             DO 400 I=1,J6
25500             IF(JDTA1(I,5).EQ.1)ONFL(I)=1
25600             IF(ONFL(I).EQ.1.AND.I.LT.NUSP1)BFLG=1
25700             IF(ONFL(I).EQ.1.AND.I.GT.NUSP1)AFLG=1
25800     400     CONTINUE
25900             IF(BFLG.EQ.0)GO TO 4003
26000             DO 4001 I=1,NUSP1
26100             IF(ONFL(I).EQ.0)GOTO 4010
26200     4001    CONTINUE
26300     4010    BFLG1=I
26400             NFLG=I-1
26500             DO 4002 I=1,NFLG
26600             CONCT(I)=BFLG1-I+JDTA1(I,8)
26700             ONCT(I)=BFLG1-I-1+JDTA1(I,8)
26800             IF(ONCT(I).LT.0)ONCT(I)=0
26900             IF(CONCT(I).LT.0)CONCT(I)=0
27000     4002    CONTINUE
27100     4003    IF(AFLG.EQ.0)GO TO 4007
27200             AFLG=1
27300             N1FLG=NUSP1+1
27400             J1=0
27500             DO 4004 I=N1FLG,J6
27600             IF(ONFL(I).EQ.1)GO TO 4005
27700     4004    CONTINUE
27800     4005    AFLG1=I
27900             NFLG=I
28000             N1FLG=NFLG-1
28100             DO 4006 I=NFLG,J6
28200             CONCT(I)=I-N1FLG+JDTA1(I,8)
28300             ONCT(I)=I-N1FLG-1+JDTA1(I,8)
28400             IF(ONCT(I).LT.0)ONCT(I)=0
28500             IF(CONCT(I).LT.0)CONCT(I)=0
28600     4006    CONTINUE
28700     4007    CONTINUE
28800   C
28900   C
29000   C.....CALCULATE NEW WATERWAY SEGMENT CITIES
29100   C.....REDEFINE CITY PAIRINGS
29200   C
29300   C
29400             J7=J
29500             NUSP1=NUSP1+1
29600             N=J7-NUSP1
29700             N1=NUSP1*N
29800             DO 209 J9=1,N1
```

```
      DO 207 J10=21,31
207   ODPR(J9,J10)=0
206   CONTINUE
      DO 30 I=1,NQAC
      IF(IAC.EQ.ACTY(I))GO TO 31
30    CONTINUE
      GO TO 901
31    INAC=I
      DO 44 I=1,10
      CYCD(I,1)=0
44    CYCD(I,2)=0
      J6=J7-1
      DO 45 I=1,J6
      CYCD(I,1)=JDTA1(I,1)
45    CYCD(I,2)=JDTA1(I,2)
      CYCD(J7,1)=JDTA1(J6,3)
      CYCD(J7,2)=JDTA1(J6,4)
      DO 453 I=1,10
453   CTYFG(I)=0
      DO 451 I=1,J7
      CYC=CYCD(I,1)+CYCD(I,2)
      DO 452 I2=1,6
      IF(CYC.EQ.CTY(I2))CTYFG(I)=1
452   CONTINUE
451   CONTINUE
C
C
C.....CHECK FOR BUSY AIRPORTS
C
C
      DO 3010 I=1,31
      F1(I)=0
      F2(I)=0
3010  CONTINUE
      DO 3011 I=1,NUSP1
      F1(I)=CYCD(I,1)
3011  CONTINUE
      F1PTR=NUSP1
c:type*,f1ptr,(f1(i),i=1,f1ptr)
      J=0
      DO 3012 I=NEUP1,J7
      J=J+1
      F2(J)=CYCD(I,1)
3012  CONTINUE
      F2PTR=J
c:type*,j,neup1,j7,(cycd(i,1),i=1,j7)
      I=0
      IF(CTYFG(NUSP1).EQ.1)JCTFLG=1
      IF(CTYFG(NEUP1).EQ.1)JCTFLG=1
C
C
C.....QSI CALCULATIONS
C
C.....INTERNATIONAL FLIGHT
C
C
      DO 32 J=1,NUSP1
      DO 33 K=NEUP1,J7
      I=I+1
      DO 320 K2=21,64
320   ODPR(I,K2)=0
```

```
      K3=J
      K4=NUSP1
      K5=K-NEUP1
      K6=K4+K5
      DO 321 K7=K3,K6
      L2=33+K7
      L3=44+K7
      L4=55+K7
      ODPR(I,L2)=DIST(K7)
      ODPR(I,L3)=TTIME(K7)
      ODPR(I,L4)=GTIME(K7)
      IF(K7.EQ.K6)ODPR(I,L4)=0
      L1=21+K7
      ODPR(I,L1)=JACT(K7)
  321 CONTINUE
      CFG(I)=0
      IF(JCTFLG.EQ.1)CFG(I)=1
      IF(CFG(I).EQ.1)GO TO 380
      J1=J
      K2=K
      DO 381 J2=J1,K2
      IF(CTYFG(J2).EQ.1)CFG(I)=1
  381 CONTINUE
  380 CONTINUE
      ODPR(I,9)=0
      ODPR(I,1)=CYCD(J,1)
      ODPR(I,2)=CYCD(J,2)
      ODPR(I,3)=CYCD(K,1)
      ODPR(I,4)=CYCD(K,2)
      ODPR(I,5)=ODPR(I,1)
      ODPR(I,6)=ODPR(I,2)
      ODPR(I,7)=ODPR(I,3)
      ODPR(I,8)=ODPR(I,4)
      ODPR(I,17)=T2
      ODPR(I,10)=IAC
      ODPR(I,11)=T3
      L1=K-1
      DO 35 L=J,L1
   35 ODPR(I,9)=ODPR(I,9)+JDTA1(L,6)
      ITMFLG=0
      INFLG=0
      NSTOPS=0
      INDEX=0
C
C
C.....ONLINE CONNECTION...CHECK FOR TYPES OF QSI
C
C
      DO 360 L=J,L1
      IF(JDTA1(L,7).GT.0)ITMFLG=1
  360 CONTINUE
      IF(BFLG.EQ.0.AND.AFLG.EQ.0)GO TO 361
      IF(BFLG.EQ.0.AND.AFLG.EQ.1)GO TO 364
      IF(BFLG.EQ.1.AND.AFLG.EQ.0)GO TO 365
      IF(J.LT.BFLG1.AND.K.GT.AFLG1)GO TO 363
      IF(J.GE.BFLG1.AND.K.LE.AFLG1)GO TO 361
      IF(J.LT.BFLG1.AND.K.LE.AFLG1)GO TO 3551
      IF(J.GE.BFLG1.AND.K.GT.AFLG1)GO TO 3541
  364 IF(K.LE.AFLG1)GO TO 361
 3541 K10=AFLG1
      DO 368 L=J,K10
```

```
  366     NSTOPS=NSTOPS+JDTA1(I,8)
          N1=(AFLG1-J)-1+NSTOPS
          IF(N1.GT.1)GOTO 363
          ON1=ONCT(K)
  370     IF(ON1.GT.1)GO TO 363
          IF(N1.EQ.1.AND.ON1.EQ.1)GO TO 363
          INDEX=5
          IF(N1.EQ.0.AND.ON1.EQ.0)INDEX=4
          GO TO 40
  365     IF(J.GE.BFLG1)GO TO 361
  3651    K10=BFLG1
          DO 367 L9=K10,L1
          NSTOPS=NSTOPS+JDTA1(L9-1,8)
  367     CONTINUE
          N1=(K-BFLG1)-1+NSTOPS
          IF(N1.GT.1)GO TO 363
          ON1=ONCT(J)
          GO TO 370
  363     INDEX=8
          GO TO 40
  361     DO 36 L=J,L1
  36      NSTOPS=NSTOPS+JDTA1(L,8)
          N1=(K-J)-1+NSTOPS
          IF(N1.GT.2)INDEX=8
          IF(INDEX.EQ.8)GO TO 40
          IF(N1.EQ.2)INDEX=3
          IF(N1.EQ.1)INDEX=2
          IF(N1.EQ.0)INDEX=1
  40      FAC1=0.0
          FAC2=0.0
          FAC3=0.0
          FAC4=0.0
          IF(CFG(I).EQ.1)FAC1=QTT(1)
          IF(CFG(I).EQ.1)FAC2=QTT(2)
          IF(IMAFLG.EQ.1)FAC3=QMA(1)
          IF(IMAFLG.EQ.1)FAC4=QMA(2)
  41      QDPR(1)=T3*(QMK(INDEX)+FAC1+FAC3)*QAC(INAC)
          QDPR(2)=T3*(QST(INDEX)+FAC2+FAC4)*QAC(INAC)
          ODPR(I,12)=0
          ODPR(I,14)=0
          ODPR(I,15)=INDEX
          ODPR(I,16)=ITMFLG
          ODPR(I,18)=BB
          ODPR(I,20)=BB
          ODPR(I,32)=USEPT
          ODPR(I,19)=T1
          DO 42 L2=1,5
          IF(T1.EQ.CARTY(L2))GO TO 43
  42      CONTINUE
          ODPR(I,13)=2
          GO TO 3333
  43      ODPR(I,13)=1
  3333    CONTINUE
          IF(ODPR(I,5).le.0.OR.ODPR(I,7).le.0)GO TO 33
          IF(QDPR(1).EQ.0.)GOTO 3900
          if(qdpr(1).lt.thresh)go to 33
  3900    CONTINUE
C         WRITE(22,39)(ODPR(I,L),L=1,64),(QDPR(I2),I2=1,2)
C 39      FORMAT(1X,4(I5,T1),I5,I3,I2,5I1,I6,3A2,11I2,I1,32I5,2F10.4)
          WRITE(22,39)(ODPR(I,L),L=1,9),(ODPR(I,L),L=11,14)
          1,(ODPR(I,L),L=17,20),ODPR(I,32),(QDPR(I2),I2=1,2)
```

```
c=type39,(ODPR(I,L),L=1,9),(ODPR(I,L),L=11,14)
c=1,(ODPR(I,L),L=17,20),ODPR(I,32),(ODPR(I2),I2=1,2)
 39       FORMAT(1X,4(I5,I1),I5,I2,3I1,I6,3I2,I1,3F10.4)
          WRITE(7)(ODPR(I,L),L=1,4),(ODPR(I,L),L=9,14),ODPR(I,17),
     X    ODPR(I,18),ODPR(I,32),(ODPR(I,L),L=45,53),(ODPR(I,L),L=56,63),
     X    ODPR(1)
          NWRITE=NWRITE+1
 33       CONTINUE
 32       CONTINUE
          IF(NWRITE.GT.MWRITE)GO TO 1000
          IF(WIFLG.EQ.1)GO TO 4
          IF(ENDC.EQ.1)GO TO 4
          DO 48 I=1,10
 48       IFLCNT(I)=0
485       OPEN(UNIT=25,DEVICE='DSK',FILE='CONTMP.DAT',MODE='BINARY',
     X    ACCESS='SEQOUT')
          CONFLG=0
C
C
C.....READ CONNECTION FILE
C
C
 49       READ(21,50,END=999)(CTFL(I),I=1,13)
 50       FORMAT(1X,A2,I6,2(I5,I1),3I5,I1,A3,I2,A2)
C 50      FORMAT(1X,A2,I6,I6,I1,I5,I1,I4,2I5,I1,A3,I2,A2)
c=type 50,ctfl
          if(ctfl(1).lt.t1)go to 49
          IF(CTFL(1).NE.T1)GO TO 51
          if(ctfl(2).lt.t2)go to 49
          IF(CTFL(2).NE.T2)GO TO 51
          DO 5111 M2=1,2
          DO 5113 M3=1,31
          GO TO (5114,5115),M2
 5114     IF(CTFL(5).EQ.USPT(M3))GO TO 5112
          GO TO 5113
 5115     IF(CTFL(5).EQ.EUPT(M3))GO TO 5112
 5113     CONTINUE
 5111     CONTINUE
          GO TO 49
 5112     CONTINUE
          WRITE(25)(CTFL(I),I=1,13)
          CONFLG=1
          GO TO 49
 999      ENDC=1
 51       BACKSPACE 21
          CLOSE(UNIT=25)
          IF(CONFLG.EQ.0)GO TO 200
          DO 52 I=1,J7
          IF(I.LE.5)I2=10
          IF(I.GT.5)I2=17
          I3=I+I2
          NAME=NAME(I)
          OPEN(UNIT=I3,DEVICE='DSK',FILE=NAME,MODE='BINARY',
     X    ACCESS='SEQOUT')
 52       CONTINUE
          OPEN(UNIT=10,DEVICE='DSK',FILE='CONTMP.DAT',MODE='BINARY',
     X    ACCESS='SEQIN')
 53       READ(10,END=60)(CTFL(I),I=1,13)
          DO 54 I=1,J7
          IF(CTFL(3).EQ.CYCD(I,1))GO TO 57
          CONTINUE
```

```
52600              GO TO 53
52700       55     IUNIT=LBL(I)
52800              IFICNT(I)=IFICNT(I)+1
52900              WRITE(IUNIT)(CTFL(K),K=1,13)
53000              IF(I.GT.NUSP1)GO TO 3014
53100              DO 3013 K=1,F1PTR
53200              IF(CTFL(5).EQ.F1(K))GO TO 53
53300       3013   CONTINUE
53400              K2=F1PTR+1
53500              F1(K2)=CTFL(5)
53600              F1PTR=F1PTR+1
53700              GO TO 53
53800       3014   DO 3015 K=1,F2PTR
53900              IF(CTFL(5).EQ.F2(K))GO TO 53
54000       3015   CONTINUE
54100              K2=F2PTR+1
54200              F2(K2)=CTFL(5)
54300              F2PTR=F2PTR+1
54400              GO TO 53
54500       60     DO 56 I=1,J7
54600              IUNIT=LBL(I)
54700       56     CLOSE(UNIT=IUNIT)
54800              CLOSE(UNIT=10,DISPOSE='DELETE')
55100      C
55200      C
55300      C.....QSI CALCULATIONS FOR EACH O-D
55400      C
55500      C
55600              I3=0
55700              DO 62 I=1,NUSP1
55800              OCFLG=0
55900              NAME=NAME(I)
56000              DO 63 J=NEUP1,J7
56100              I3=I3+1
56200              ODFLG=0
56300              LAME=NAME(J)
56400       152    OPEN(UNIT=24,DEVICE='DSK',FILE=NAME,MODE='BINARY',
56500           X  ACCESS='SEQIN')
56600       71     OPEN(UNIT=25,DEVICE='DSK',FILE=LAME,MODE='BINARY',
56700           X  ACCESS='SEQIN')
56800              DO 64 J1=5,8
56900       64     PDR(J1)=ODPR(I3,J1)
57000              DO 65 J1=10,13
57100       65     PDR(J1)=ODPR(I3,J1)
57200              PDR(32)=ODPR(I3,32)
57300              PDR(17)=ODPR(I3,17)
57400              DO 661 J1=22,30
57500              PDR(J1)=ODPR(I3,J1)
57600       661    CONTINUE
57700              DO 663 J1=33,64
57800       663    PDR(J1)=ODPR(I3,J1)
57900              PDR(18)=ODPR(I3,18)
58000              IJ=IFICNT(J)
58100              IF(ODFLG.EQ.1)GO TO 70
58200      C
58300      C
58400      C.....BACK END CONNECTIONS
58500      C
58600      C
58700              IF(IJ.EQ.0)GO TO 1667
58800              J1=0
```

```
166     J1=J1+1
        IF(J1.GT.IJ) GO TO 1667
        READ(25) (DTFL(M),M=1,13)
        ICTFLG=0
        IF(JCTFLG.EQ.1) GO TO 1212
        IF(CFG(I3).EQ.1) ICTFLG=1
        IF(ICTFLG.EQ.1) GO TO 1212
        CYC=DTFL(5)+DTFL(6)
        DO 1202 M1=1,6
        IF(CYC.EQ.CTY(M1)) ICTFLG=1
1202    CONTINUE
1212    CONTINUE
        PDR(1)=ODPR(I3,1)
        PDR(2)=ODPR(I3,2)
        PDR(3)=DTFL(5)
        PDR(4)=DTFL(6)
        PDR(21)=0
        PDR(18)=BB
        PDR(20)=DTFL(13)
        PDR(33)=0
        PDR(44)=0
        PDR(55)=0
        PDR(43)=DTFL(8)
        PDR(54)=DTFL(7)
        PDR(64)=DTFL(9)
        PDR(31)=0
        DO 662 J9=1,NACTY
        IF(ACTY(J9).EQ.DTFL(11)) PDR(31)=J9
662     CONTINUE
        PDR(9)=ODPR(I3,9)+DTFL(8)
        INDEX=0
        INFLG=0
        INTFLG=0
        PDR(12)=0
        PDR(14)=2
        IF(USEPT.EQ.2) PDR(14)=1
        DO 66 J3=1,5
        IF(DTFL(13).EQ.CARTY(J3)) GO TO 67
66      CONTINUE
        IF(DTFL(13).EQ.CANC(1).OR.DTFL(13).EQ.CANC(2)) PDR(14)=2
        INTFLG=1
        GO TO 689
67      PDR(14)=1
689     CONTINUE
        IF(DTFL(13).NE.ODPR(I3,19)) INTFLG=1
        IF(DTFL(13).NE.ODPR(I3,19)) GO TO 68
        IF(ODPR(I3,10).NE.DTFL(11)) INFLG=1
68      IGT=ODPR(I3,16)
        IGT1=0
        IF(DTFL(9).GE.120) IGT1=1
        IGT2=IGT+IGT1
        IGTFLG=0
        IF(IGT2.GE.1) IGTFLG=1
        PDR(16)=IGTFLG
        IMAFLG=0
        IF(DTFL(10).GT.0) IMAFLG=1
        IM=IGTFLG+IMAFLG
        OD=ODPR(I3,13)
        IF(OD.GE.3) INDEX=8
        IF(DTFL(12).GT.1) INDEX=[illegible]
        IF(OD.EQ.2.AND.DTFL(12).EQ.[illegible]) INDEX=[illegible]
```

```
      IF(INDEX.EQ.8)GO TO 100
      IF(INTFLG.EQ.1)GO TO 69
      INDEX=5
      IF(CD.EQ.1.AND.DTFI(12).EQ.0)INDEX=8
      GO TO 100
 69   INDEX=7
      IF(CD.EQ.1.AND.DTFI(12).EQ.0)INDEX=6
 100  FAC1=0.0
      FAC2=0.0
      FAC3=0.0
      FAC4=0.0
      DO 103 M3=1,NQAC
      IF(PDR(10).EQ.ACTY(M3))GO TO 104
 103  CONTINUE
 104  INAC=M3
      IF(JCTFLG.EQ.1.OR.ICTFLG.EQ.1)FAC1=QTT(1)
      IF(JCTFLG.EQ.1.OR.ICTFLG.EQ.1)FAC2=QTT(2)
 101  IF(IM.EQ.0)GO TO 102
      FAC3=QMA(1)
      FAC4=QMA(2)
 102  QDPR(1)=QDPR(I3,11)*(QMK(INDEX)+FAC1+FAC3)*QAC(INAC)
      QDPR(2)=QDPR(I3,11)*(QMK(INDEX)+FAC2+FAC4)*QAC(INAC)
      PDR(15)=INDEX
      IF(QDPR(1).EQ.0.)GOTO 1020
      if(qdpr(1).lt.thresh)go to 166
 1020 CONTINUE
      if(pdr(5).le.0.or.pdr(7).le.0)go to 166
      WRITE(22,39)(PDR(I2),I2=1,9),(PDR(I2),I2=11,14)
      1,(PDR(I2),I2=17,20),PDR(32),(QDPR(I2),I2=1,2)
      NWRITE=NWRITE+1
      GO TO 166
 1667 ODFLG=1
 1666 CLOSE(UNIT=25)
 151  READ(24,END=170)(CTFI(N2),N2=1,13)
      GO TO 71
 70   IF(OCFLG.EQ.1)GO TO 120
C
C
C.....FRONT END CONNECTIONS
C
C
      OCFLG=1
      ICTFLG=0
      IF(JCTFLG.EQ.1)GO TO 1211
      IF(CFG(I3).EQ.1)ICTFLG=1
      CYC=CTFI(5)+CTFI(6)
      DO 1201 M1=1,6
      IF(CYC.EQ.CTY(M1))ICTFLG=1
 1201 CONTINUE
 1211 CONTINUE
      PDR(21)=0
      PDR(20)=BR
      PDR(18)=CTFI(13)
      PDR(21)=0
      DO 665 J9=1,NACTY
      IF(ACTY(J9).EQ.CTFI(11))PDR(21)=J9
 665  CONTINUE
      PDR(43)=0
      PDR(34)=0
      PDR(64)=0
      PDR(33)=CTFI(8)
```

```
      PDR(44)=CTFL(7)
      PDR(57)=CTFL(9)
      PDR(1)=CTFL(3)
      PDR(2)=CTFL(6)
      PDR(3)=ODPR(I3,3)
      PDR(4)=ODPR(I3,4)
      PDR(9)=ODPR(I3,9)+CTFL(8)
      INDEX=0
      IMFLG=0
      INTFLG=0
      PDR(12)=1
      IF(USEPT.EQ.2)PDR(12)=2
      PDR(14)=0
      DO 72 J3=1,5
      IF(CTFL(13).EQ.CARTY(J3))GO TO 73
72    CONTINUE
      IF(CTFL(13).EQ.CANC(1).OR.CANC(2).EQ.CTFL(13))PDR(12)=2
      GO TO 744
73    PDR(12)=1
744   CONTINUE
      IF(CTFL(13).NE.ODPR(I3,19))INTFLG=1
      IF(INTFLG.EQ.1)GO TO 74
      IF(CTFL(11).NE.ODPR(I3,10))INFLG=1
74    IGT=ODPR(I3,16)
      IGT1=0
      IF(CTFL(9).GE.120)IGT1=1
      IGT2=IGT1+IGT
      IGTFLG=0
      IF(IGT2.GE.1)IGTFLG=1
      PDR(15)=IGTFLG
      IMAFLG=0
      IF(CTFL(10).GT.0)IMAFLG=1
      OD=ODPR(I3,15)
      IM=IGTFLG+IMAFLG
      IF(OD.GE.3)INDEX=8
      IF(CTFL(12).GT.1)INDEX=8
      IF(OD.EQ.2.AND.CTFL(12).EQ.1)INDEX=9
      IF(INDEX.EQ.8)GO TO 85
      IF(INTFLG.EQ.1)GO TO 75
      INDEX=5
      IF(OD.EQ.1.AND.CTFL(12).EQ.0)INDEX=4
      GO TO 85
75    INDEX=7
      IF(OD.EQ.1.AND.CTFL(12).EQ.0)INDEX=6
85    FAC1=0.0
      FAC2=0.0
      FAC3=0.0
      FAC4=0.0
      DO 86 M3=1,NQAC
      IF(PDR(10).EQ.ACTY(M3))GO TO 87
86    CONTINUE
87    INAC=M3
      IF(JCTFLG.EQ.1.OR.ICTFLG.EQ.1)FAC1=QTT(1)
      IF(JCTFLG.EQ.1.OR.ICTFLG.EQ.1)FAC2=QTT(2)
88    IF(IM.EQ.0)GO TO 89
      FAC3=QMA(1)
      FAC4=QMA(2)
89    QDPR(1)=ODPR(I3,11)*(QNT(INDEX)+FAC1+FAC3)*QAC(INAC)
      QDPR(2)=ODPR(I3,11)*(QNT(INDEX)+FAC2+FAC4)*QAC(INAC)
      PDR(17)=INDEX
      IF(QDPR(1).EQ.0.)GO TO 90
```

```
76900              IF(qdpr(1).lt.thresh)go to 1660
76975      1100    CONTINUE
77000              IF(pdr(5).le.0.or.pdr(7).le.0)go to 1660
77025              WRITE(22,39)(PDR(I2),I2=1,9),(PDR(I2),I2=11,14)
77050              1,(PDR(I2),I2=17,20),PDR(32),(QDPR(I2),I2=1,2)
77075              NWRITE=NWRITE+1
77100      1660    ICTFLG=1
77200              GO TO 71
77300      C
77400      C
77500      C.....CONNECTIONS BOTH ENDS
77600      C
77700      C
77800       120    IJ=IFLCNT(J)
77900              IF(IJ.EQ.0)GO TO 1209
78000              J1=0
78100       121    J1=J1+1
78200              IF(J1.GT.IJ)GO TO 1219
78300              READ(25)(DTFL(M),M=1,13)
78400              ICTFLG=0
78500              IF(JCTFLG.EQ.1)GO TO 1210
78600              IF(CFG(I3).EQ.1)ICTFLG=1
78700              CYC=CTFL(5)+CTFL(6)
78800              CYC1=DTFL(5)+DTFL(6)
78900              DO 1200 M1=1,4
79000              IF(CYC.EQ.CTY(M1))ICTFLG=1
79100              IF(CYC1.EQ.CTY(M1))ICTFLG=1
79200       1200   CONTINUE
79300       1210   CONTINUE
79400              PDR(20)=DTFL(13)
79500              PDR(18)=CTFL(13)
79600              PDR(32)=CTFL(8)
79700              PDR(43)=DTFL(8)
79800              PDR(44)=CTFL(7)
79900              PDR(54)=DTFL(7)
80000              PDR(55)=CTFL(9)
80100              PDR(64)=DTFL(9)
80200              DO 664 J9=1,NACTY
80300              IF(ACTY(J9).EQ.DTFL(11))PDR(31)=J9
80400       664    CONTINUE
80500              PDR(1)=CTFL(5)
80600              PDR(2)=CTFL(6)
80700              PDR(3)=DTFL(5)
80800              PDR(4)=DTFL(6)
80900              PDR(9)=CDPR(I3,9)+CTFL(8)+DTFL(8)
81000              INDEX=0
81100              PDR(14)=2
81200              IF(USERT.EQ.2)PDR(14)=1
81300              DO 122 J3=1,5
81400              IF(DTFL(13).EQ.CARTY(J3))GO TO 1233
81500       122    CONTINUE
81600              IF(DTFL(13).EQ.CANC(1).OR.DTFL(13).EQ.CANC(2))PDR(14)=2
81700              GO TO 1234
81800       1233   PDR(14)=1
81900       1234   PDR(12)=1
82000              IF(USFPC.EQ.2)PDR(12)=0
82100       123    DO 124 J3=1,5
82200              IF(CTFL(13).EQ.CARTY(J3))GO TO 1235
82300       124    CONTINUE
82400              IF(CTFL(13).EQ.CANC(1).OR.CTFL(13).EQ.CANC(2))PDR(12)=0
82500              GO TO 125
```

```
      1255    PDR(12)=1
      1?      IGT=CDPR(I3,16)
              IGT1=0
              IF(CTFI(9).GT.120)IGT1=1
              IGT2=0
              IF(DTFL(9).GT.120)IGT2=1
              IGT3=IGT+IGT1+IGT2
              IGTFLG=0
              IF(IGT3.GE.1)IGTFLG=1
              PDR(16)=IGTFLG
              IMAFLG=0
              IF(CTFI(11).GT.0)IMAFLG=1
              IF(DTFL(11).GT.0)IMAFLG=1
              IM=IMAFLG+IGTFLG
              INDEX=8
      140     FAC1=0.0
              FAC2=0.0
              FAC3=0.0
              FAC4=0.0
              DO 141 M3=1,NQAC
              IF(PDR(10).EQ.ACTY(M3))GO TO 142
      141     CONTINUE
      142     INAC=M3
              IF(JCTFIG.EQ.1.OR.ICTFIG.EQ.1)FAC1=QTT(1)
              IF(JCTFIG.EQ.1.OR.ICTFIG.EQ.1)FAC2=QTT(2)
      143     IF(IM.EQ.0)GO TO 144
              FAC3=QMA(1)
              FAC4=QMA(2)
      144     QDPR(1)=CDPR(I3,11)*(QMK(INDEX)+FAC1+FAC3)*QAC(INAC)
              QDPR(2)=CDPR(I3,11)*(QST(INDEX)+FAC2+FAC4)*QAC(INAC)
              PDR(15)=INDEX
              IF(QDPR(1).EQ.0.)GOTO 1440
              if(qdpr(1).lt.thresh)go to 121
     1440     CONTINUE
              if(pdr(5).le.0.or.pdr(7).le.0)go to 121
              WRITE(22,39)(PDR(I2),I2=1,9),(PDR(I2),I2=11,14)
              1,(PDR(I2),I2=17,20),PDR(32),(QDPR(I2),I2=1,2)
              NWRITE=NWRITE+1
              GO TO 121
      1219    CLOSE(UNIT=25)
      1209    CCFIG=0
              GO TO 151
      170     CLOSE(UNIT=24)
      63      CONTINUE
      62      CONTINUE
              CLOSE(UNIT=25)
              IF(NWRITE.GT.MWRITE)GO TO 1000
              GO TO 4
C
C
C.....ERROR MESSAGES
C
C
      100     WRITE(TTY,850)T2,T1
      850     FORMAT(' ',T6,'ERROR...CANNOT DETERMINE MATERIAL FOR FLIGHT' I7
     1        ' CARRIER ',I2,/' ',T6,'CANNOT PROCESS THIS FLIGHT'
              1,' except by assuming intermediate point to be on origin side'
              goto 130
              GO TO 990
      101     WRITE(TTY,851)INC
              FORMAT(' ',T6,'ERROR...NO WEIGHT FOR AIRCRAFT TYPE ',I3,
```

```
      X  /' ',T3,'CANNOT PROCESS THIS FLIGHT')
  900     READ(21,50)(CTFL(I),I=1,13)
c=type 50,ctfl
          IF(CTFL(2).EQ.T2)GO TO 900
          BACKSPACE 21
          GO TO 4
  900     CONTINUE
          T9=CTFL(1)
          T10=CTFL(2)
          IF(T9.NE.T1)GO TO 2001
          IF(T10.GT.T2)GO TO 2001
  2002    READ(21,50,END=2006)(CTFL(I),I=1,13)
c=type 50,ctfl
          if(ctfl(1).lt.t1)go to 2002
          if(ctfl(1).gt.t1)go to 2001
          IF(CTFL(2).EQ.T2)GO TO 2005
          IF(CTFL(2).LT.T2)GO TO 2002
          IF(CTFL(2).GT.T2)GO TO 2001
  2001    BACKSPACE 21
          READ(20,3,END=1000)(HDTA(I),I=1,4)
c=type 3,hdta
          T1=HDTA(1)
          T2=HDTA(2)
          T3=HDTA(3)
          WLFLG=1
          GO TO 5
  2005    BACKSPACE 21
          CONFIG=0
          GO TO 485
  2006    ENDC=1
          GO TO 4
  1000    CONTINUE
          CLOSE(UNIT=21)
          CLOSE(UNIT=22)
          CLOSE(UNIT=9)
          CLOSE(UNIT=20)
          type *,nwrite
          STOP
          END
```

APPENDIX B

Program Listings:

ROUTE., PR3., BSCNOD.CTL,

FLT.CTL, HFILES.CTL, BSC0OD.CTL

ROUTE

```
C     ROUTE
C
C     THIS PROGRAM PRODUCES LINK LOADING DESCRIPTIONS FROM
C     ORIGIN-DESTINATION DATA
C
      SUBROUTINE ROUTE
      COMMON CARIP,FNOP,TYPEP,NFQP,SS,DIST,TIME,KPAX,NSTNS,NLINKS
     1 ,KPORIG,DIS,TIM,KPASS,INDEX,KK
      DIMENSION KPORIG(11),KPDEST(11),SS(11),DIST(11),TIME(11)
     1 ,ORIG(11),DISTA(11),TIMEA(11),KPAX(11)
      INTEGER CARI,FNO,O,D,CARIP,FNOP,OP,ORIG,SS,BLANK,TYPE,TYPEP
      IF(INDEX.NE.-1)GO TO 13
C
C     INITIALIZATION
C
      READ(5,END=61)O,N1,D,N2,KDIS,TYPE,NFQ,N3,N4,N5,FNO,CARI,KDIR
     1 ,KTIM,N6,R7,IP1,IP2,IP3,JP1,JP2,JP3,N8
      DIS=KDIS
      TIM=KTIM/60.
      IF(KDIR.EQ.1)KPASS=JP1+JP2+JP3
      IF(KDIR.EQ.2)KPASS=IP1+IP2+IP3
      WRITE(6,101)CARI,FNO,O,D,TYPE,NFQ,DIS,TIM,KPASS
101   FORMAT(1XA3,2XI5,2I6,1XA3,2XI2,2X2F6.1,I5)
13    K=0
      KK=1
      SMALLD=10.**9
      SMALLT=10.**9
      INDEX=1
      ORIG(1)=O
      DISTA(1)=DIS
      TIMEA(1)=TIM
      DO 1 I=1,10
      SS(I)=0
      DIST(I)=0.
      TIME(I)=0.
      KPAX(I)=0
      KPORIG(1)=0
      KPDEST(1)=0
1     CONTINUE
      SS(11)=0
      KPORIG(11)=0
      KPDEST(11)=0
      GO TO 3
C
C     READ ORIGIN-DESTINATION DATA
C
2     READ(5,END=61)O,N1,D,N2,KDIS,TYPE,NFQ,N3,N4,N5,FNO,CARI,KDIR
     1 ,KTIM,N6,R7,IP1,IP2,IP3,JP1,JP2,JP3,N8
      DIS=KDIS
      TIM=KTIM/60.
      IF(KDIR.EQ.1)KPASS=JP1+JP2+JP3
      IF(KDIR.EQ.2)KPASS=IP1+IP2+IP3
      WRITE(6,101)CARI,FNO,O,D,TYPE,NFQ,DIS,TIM,KPASS
      IF(FNO.NE.FNOP)GO TO 6
      IF(CARI.NE.CARIP)GO TO 6
      IF(O.NE.OP)GO TO 5
3     K=K+1
C
C     PROCESS DESTINATIONS, FOR GIVEN ORIGIN
C
      IF(K.GT.11 .OR. KK.GT.11)TYPE 301,K,KK
```

```
301     FORMAT(' ROUTE',5X'K=',I2,' KK=',I2)
        KPORIG(KK)=KPORIG(KK)+KPASS
        KPDEST(K)=KPDEST(K)+KPASS
        SMALLD=AMIN1(SMALLD,DIS)
        SMALLT=AMIN1(SMALLT,TIM)
        IF(INDEX.NE.1)GO TO 4
        DS(K)=D
        IF(K.EQ.1)GO TO 4
        DIST(K)=DISP-DIS
        TIME(K)=TIMP-TIM
4       DF=0
        DISP=DIS
        TIMP=TIM
        FNOP=FNO
        CARIP=CARI
        TYPEP=TYPE
        NFOP=NFO
        GO TO 2
C
C  PROCESS THE NEXT ORIGIN
C
5       KK=KK+1
        IF(KK.GT.11)TYPE 302,KK
302     FORMAT(' ROUTE',5X,'KK=',I2)
        K=0
        INDEX=0
        ORIG(KK)=D
        DISTA(KK)=DIS
        TIMEA(KK)=TIM
        GO TO 3
C
C  COMPUTE LINK LOADING DESCRIPTIONS OF GIVEN FLIGHT,
C  THEN RETURN TO MAIN PROGRAM.
C
6       INDEX=0
        GO TO 62
61      INDEX=1
62      IF(K+KK.GT.11)TYPE 303,K,KK
303     FORMAT(' ROUTE',5X'K+KK=',I5)
        IF(KK.LE.1)GO TO 71
        DO 7 I=1,KK-1
        DO 7 J=I+1,KK
        IF(DISTA(I).LE.DISTA(J))GO TO 7
        ITMP=ORIG(I)
        ORIG(I)=ORIG(J)
        ORIG(J)=ITMP
        TMP=DISTA(I)
        DISTA(I)=DISTA(J)
        DISTA(J)=TMP
        ITMP=KPORIG(I)
        KPORIG(I)=KPORIG(J)
        KPORIG(J)=ITMP
        TMP=TIMEA(I)
8       TIMEA(I)=TIMEA(J)
        TIMEA(J)=TMP
7       CONTINUE
71      DS(K+1)=ORIG(1)
        DIST(K+1)=SMALLD
        TIME(K+1)=SMALLT
        IF(KK.LE.1)GO TO 61
        DO 8 KI=2,KK
```

```
      SS(K+KI)=ORIG(KI)
      DIST(K+KI)=DISTA(KI)-DISTA(KI-1)
      TIME(K+KI)=TIMEA(KI)-TIMEA(KI-1)
91    KPAX(2)=KPDEST(1)
      IF(K.LE.1)GO TO 91
      DO 9 KJ=2,K
9     KPAX(KJ+1)=KPAX(KJ)+KPDEST(KJ)
91    IF(KK.LE.1)GO TO 111
      DO 10 KJJ=1,KK-1
10    KPAX(K+KJJ+1)=KPAX(K+KJJ)-KPORIG(KJJ)
111   I=(K+KK+1)/2
      DO 11 J=1,I
      KKK1J=K+KK+1-J
      ITMP=SS(J)
      SS(J)=SS(KKK1J)
      SS(KKK1J)=ITMP
      TMP=DIST(J)
      DIST(J)=DIST(KKK1J)
      DIST(KKK1J)=TMP
      TMP=TIME(J)
      TIME(J)=TIME(KKK1J)
      TIME(KKK1J)=TMP
      ITMP=KPAX(J)
      KPAX(J)=KPAX(KKK1J)
      KPAX(KKK1J)=ITMP
11    CONTINUE
      NSTNS=K+KK
      NLINKS=K+KK-1
      RETURN
      END
```

```
C  FNANCE
C
C
C        THIS PROGRAM IS A FINANCIAL MODEL FOR ALLOCATING REVENUES
C  AND EXPENSES OF PASSENGER AVIATION.  IT HAS THREE MAIN PURPOSES:
C
C        1)  TO ALLOCATE REVENUES AMONG SPECIFIC CARRIERS AND FLIGHTS.
C        2)  TO ALLOCATE COSTS AMONG VARIOUS COST CATEGORIES FOR SPECIFIC
C          FLIGHTS, SPECIFIC CARRIERS, OR ALL CARRIERS.
C        3)  TO COMPUTE LOAD FACTORS AND REVENUE/COST RATIOS.
C
C
C        THE PROGRAM IS IN TWO PARTS: "ROUTE", WHICH READS IN FLIGHT
C  AND PASSENGER ORIGIN-DESTINATION DATA AND PRODUCES FROM THEM LINK
C  LOADING DESCRIPTIONS FOR EACH FLIGHT; AND "FNANCE", WHICH USES THE
C  LINK DESCRIPTIONS, AS WELL AS COST AND REVENUE INFORMATION CONTAINED
C  IN "DATA" STATEMENTS, TO ESTIMATE REVENUE AND COST BREAKDOWNS
C  FOR INDIVIDUAL FLIGHTS AND CARRIERS.
C
C
C
C  INPUT DATA
C
C
C        1)  FILE "DAVE.DAT", CONTAINING ORIGIN-DESTINATION DATA
C        (READ IN AS INPUT), AND SORTED ALONG THE FOLLOWING KEYS (IN
C        DESCENDING ORDER):
C
C              A)  AIRLINE
C              B)  FLIGHT NUMBER
C              C)  ORIGIN AIRPORT
C              D)  DISTANCE TO DESTINATION AIRPORT
C                (IN DECREASING ORDER OF DISTANCE)
C
C
C        2)  FORM 41 COST AND REVENUE DATA.  THESE ARE PROCESSED INTO
C  EXPENSE AND YIELD PARAMETERS ENTERED INTO THE PROGRAM AS "DATA"
C  STATEMENTS.  THESE PARAMETERS ARE USED TO DETERMINE VARIOUS COSTS
C  AND REVENUES ACCORDING TO THE FOLLOWING FORMULAS (WITH SCALING
C  CONSTANTS IGNORED):
C
C
C          UBAC -- AIRCRAFT DIRECT OPERATING COST
C             ADC=CL(1)=UBAC*(BLOCK TIME FOR FLIGHT LINK)
C
C          UBCC -- CABIN CREW COST
C             CC=CL(2)=UBCC*(BLOCK TIME FOR FLIGHT LINK)
C
C          UATS -- AIRCRAFT AND TRAFFIC SERVICES
C             ATS=CL(3)=UATS  (CARRIER SPECIFIC)
C
C          URS  -- RESERVATIONS AND SALES
C             RS=CL(4)=URS*(NUMBER OF PASSENGERS ON FLIGHT LINK)
C
C          UPC  -- PASSENGER COMMISSIONS
C             PC=CL(5)=UPC*(PASSENGER REVENUE FOR FLIGHT LINK)
C
C          UCP  -- NON-PASSENGER REVENUE/PASSENGER-REVENUE RATIO
C             NON-PASSENGER REVENUE=UCP*(PASSENGER REVENUE FOR FLIGHT LINK)
C
C          UCS  -- CARGO COMMISSIONS
```

```
C          CC=CL(6)=UCG*0.95*(NON-PASSENGER REVENUE FOR FLIGHT LINK)
C
C       UAP  -- ADVERTISING AND PUBLICITY
C          AP=CL(7)=UAP*(PASSENGER REVENUE FOR FLIGHT LINK)
C
C       UIPE -- INCIDENTAL PASSENGER EXPENSES
C          IPE=CL(8)=UIPE*(PASSENGER-MILES ON FLIGHT LINK)
C
C       UDEP -- DEPRECIATION
C          DEP=UDEP*ADC      ("ADC"--AIRCRAFT DIRECT OPERATING COSTS)
C
C       UFO  -- FUEL AND OIL
C          FO=UFO*ADC
C
C       UGA  -- GENERAL AND ADMINISTRATION
C          GA=CL(9)=UGA*((CL(1)+CL(2)+...+CL(8)))-(DEP+FO))
C
C       YLD  -- REVENUE YIELD PER PASSENGER-MILE
C          PREV=PASSENGER REVENUE=YLD*(PASSENGER-MILES ON FLIGHT LINK)
C
C       SEATS-- NUMBER OF SEATS
C
C
C
C     3)  OTHER DATA
C
C          CRN--CARRIER NAME
C          AIRCRT--AIRCRAFT TYPE
C          KCR,KCR2--CARRIER ABBREVIATION
C          NARPTS--NUMBER OF AIRPORTS UNDER CONSIDERATION
C          NUMSTA--AIRPORT 5-DIGIT IDENTIFIER
C          LTRSTA--AIRPORT 3-LETTER IDENTIFIER
C          FACTOR--NUMBER OF WEEKS IN A QUARTER
C          PAXFAC--RECIPROCAL OF "FACTOR"
C
C
C
C
      COMMON KAR,FLITE,EQUIP,NFO,SS,DIST,TIME,KPAX,NSTNS,NLINKS
     1 ,O,D,DIS,TIM,KPASS,INDEX,KK
      DOUBLE PRECISION CRN(2,5)
      REAL UBAC(5,3),UDEP(5,3),UFO(3),UBCC(5,3),
     X          UATS(3),URS(3),UPC(3),UCG(3),UIPE(3),UAP(3),UGA(3),
     X          UCP(3),YLD(3),SEATS(5,4),BLANK
      REAL ALAT(100),ALON(100),CL(10),ALFCA(11)
      REAL CF(100,10),CC(11,10),ALFF(100),ALFC(11),CFS(100,10)
      REAL CCS(11,10),PREVCS(11),TREVCS(11)
      REAL PREVC(11),PREVF(100),PREVFS(100),DISTFS(100)
      REAL TREVC(11),TREVF(100),DISTF(100),PAX(11,100,10),TREVFS(100)
      REAL DIST(11),TIME(11),DISTAN(11,100,10),BTIME(11,100,10)
      INTEGER END,STN(100),SS(11),FLITE,EQUIP,KCR2(2),TYPCAR(10)
      INTEGER NUMSTA(58),LTRSTA(58)
      INTEGER PAXF(100),PAXFS(100)
      INTEGER NSTN(11),NN(11),NFO,KAR,KCR(3),NLK(11,100),AIRCRT(6),
     X          NFREQ(11,100),IAC(11,100),NLH(11,100,10),O,D,
     X          NRH(11,100,10),NDOM(11,100,10),KPAX(11),NC(10),NFT(11),
     X          FID(11,100),KLINK(11,100),KCS(10)
C
C
C  INPUT PARAMETERS
C
C
```

```
      DATA CAN/              'NATIONAL','        ',
     7                       'TRANSNOR','LD      ',
     8                       'PAN AMER','ICAN    ',
     A                       'AIR FRAN','CE      ',
     *                       'ALL CARR','IEPS    '/
      DATA USAC/  0.0  ,     0.0  ,     0.0  ,     0.0  ,  2373.0  ,
     2          3131.0  ,     0.0  ,     0.0  , 1723.0  ,     0.0  ,
     3          3496.0  , 2087.0  , 2173.0  ,     0.0  ,     0.0  /
      DATA USEP/.0,        .0,        .0,        .0,      14.92,
     2          5.74,        .0,        .0,       6.92,        .0,
     3         13.73,      15.04,      13.16,        .0,        .0/
      DATA UFC/45.51,41.61,41.22/
      DATA USCC/    0.0,       0.0,       0.0,       0.0,     277.0,
     2                    344.4,       0.0,       0.0,     258.4,       0.0,
     3                    494.6,     552.9,     198.9,       0.0,       0.0/
      DATA UATS/4862.,3813.,3737./
      DATA URS/11.8,16.0,13.6/
      DATA     UPC/6.449,8.826,9.688/
      DATA     UCG/3.438,4.803,3.814/
      DATA  UIPE/.00421,.00389,.00518/
      DATA UAP/3.813,2.46,2.89/
      DATA     UGA/6.224,5.90,6.523/
      DATA UCP/8.156,10.42,23.85/
      DATA YLD/.0634,.0735,.0876/
      DATA SCATS/4*0.,284.,362.,2*0.,145.,0.,367.,300.,145.,2*0.,
     1  395.,0.,145.,2*0./

      DATA AIRCRT/'747  ','74L  ','707  ','703  ','DIC  ','MISC.'/
      DATA BLANK/'    '/,END/'END '/
      DATA KCR/'NA','TW','PA'/
      DATA NFT/11*0/
        DATA NARPTS/58/
        DATA NUMSTA/  5850, 7220,12200,18500,19770,23210,24090,24260
     1  ,37320,42650,51410,57560,57970,63580,63760,69830,69980,70470
     2  ,71790,78460,79630,80220,82630,87870,96460,59460,90900
     3  , 5140, 5770,10220,13430,14870,20360,21170,25310,30120,31450
     4  ,32020,35020,39500,50590,51120,52090,52530,53940,57690,61340
     5  ,63900,66730,68310,76900,83220,85960,89000,89060,94840,96370
     6  ,99960/
        DATA LTRSTA/ 'ATL','BAL','BOS','ORD','CLE','DFW','DEN','DTW'
     1  ,'IAH','MCI','LAX','MIA','MSP','MSY','JFK','PHL','PHX','PIT'
     2  ,'PDX','STL','SFO','SJU','SEA','TPA','IAD','YMX','YYZ'
     3  ,'AMS','ATH','TXL','BRU','CAI','CGN','CPH','DUB','FRA','GVA'
     4  ,'GLA','HAM','IST','LIS','LHR','LUX','MAD','MAN','MXP','MUC'
     5  ,'NCE','OSL','CDG','FCO','SNN','AKN','THR','TLV','VIE','WAW'
     6  ,'ZRH'/
        DATA FACTOR/13.0/,PAXFAC/0.077/
        DATA KCR2/'AF','  '/
        OPEN(UNIT=5,DEVICE='DSK',MODE='IMAGE',ACCESS='SEQIN')
        INDEX=-1
C
C  START PROCESSING FLIGHT DATA
C
C=
C
C  COMPUTE PASSENGER LINK LOADINGS (USING SUBROUTINE "ROUTE")
C
C
100     IF(INDEX.EQ.1)GO TO 300
        NCAR=4
        NCAR1=5
```

```
        ITOTAL=NCAR+1
      IER=0
      IERCAR=0
      IERSTN=0
      IERCOM=0
        IEREQP=0
        CALL ROUTE
        WRITE(6,891)KAR,FLITE,EQUIP,NFW,((SS(K),DIST(K),TIME(K)
     1  ,KPAX(K)),K=1,11)
        WRITE(6,101)NLINKS,NSTNS
101     FORMAT(1X'NLINKS:',I3,'  NSTNS:',I3)
C
C  THE NUMBER OF STATIONS AND LINKS HAS BEEN DETERMINED BY THE CALL TO "ROUTE"
C
C  IDENTIFY CARRIER
C
      DO 140 I=1,9
      IF(KAR.EQ.KCR(I))GO TO 150
  140 CONTINUE
        DO 141 I=1,NCAR-NCAR1
        IF(KAR.EQ.KCR2(I))GO TO 142
141     CONTINUE
      WRITE(6,885)
        WRITE(6,880)KCR
        WRITE(6,881)KCR
        WRITE(6,882)KCR
      IER=1
      IERCAR=1
      GO TO 155
142     ICARR=I+NCAR1
        IERCAR=2
        GO TO 151
  150 ICARR=I
151     IF(ICARR.GT.NCAR)TYPE 102,ICARR
102     FORMAT(' ICARR=',I3)
C
C  IDENTIFY STATIONS
        IF(NSTNS.GT.11)TYPE 103,NSTNS
103     FORMAT(' NSTNS=',I3)
C
  155 DO 180 K=1,NSTNS
        DO 181 L=1,NARPTS
        IF(SS(K).NE.NOMSTA(L))GO TO 181
        KSTN(K)=LTRSTA(L)
        GO TO 180
181     CONTINUE
        KSTN(K)='***'
  180 CONTINUE
      IF(IERSTN.EQ.0)GO TO 190
      IF(IER.EQ.0)WRITE(6,885)
      WRITE(6,870)
      IER=1
C
C
C  IDENTIFY EQUIPMENT
C
190   DO 200 K=1,5
        IF(EQUIP.EQ.-IRCRT(K))GO TO 210
200     CONTINUE
        IEREQP=1
200     IF(IER.EQ.0)WRITE(6,885)
```

```
      WRITE(6,880)AIRCRT
      GO TO 215
  210 KEQ=K
        IF(KEQ.GT.0)TYPE 104,KEQ
104     FORMAT(' KEQ=',I3)
      IF(IERCAR.NE.0)GO TO 215
      IF(SEATS(KEQ,ICARR).EQ.0.0 .AND. IERCAR.NE.2)GO TO 235
215   IF(IER.NE.0)GO TO 100
C
C  STORE FLIGHT INFORMATION
C
      I=ICARR
      NFT(I)=NFT(I)+1
        TYPCAR(I)=IERCAR/2
        IF(NFT(I).GT.100)TYPE 105,NFT(I)
105     FORMAT(' NFT=',I3)
      J=NFT(I)
      NLK(I,J)=NLINKS
      FID(I,J)=FLITE
      NFREQ(I,J)=NFQ
      IAC(I,J)=6
        IF(IEREQP.EQ.0)IAC(I,J)=KEQ
      DO 250 K=1,NLINKS
        IF(NLINKS.GT.10)TYPE 106,NLINKS
106     FORMAT(' NLINKS=',I3)
        KLINK(I,J)=KK
      NLH(I,J,K)=KSTN(K)
      NRH(I,J,K)=KSTN(K+1)
      NDOM(I,J,K)=NN(K)
        PAX(I,J,K)=KPAX(K)
        DISTAN(I,J,K)=DIST(K)
250     BTIME(I,J,K)=TIME(K)
      GO TO 100
C
C  END PROCESSING OF FLIGHT DATA
C
C
C
C  BEGIN REVENUE/COST COMPUTATION
C
C
  300 LIMMAX=54
      RPMT=0.
      SEATMT=0.
      DO 540 I=1,NCAR
      NFTX=NFT(I)
        RPMCA=0.
        SEATMA=0.
      IF(NFTX.EQ.0)GO TO 540
      RPMC=0.
      SEATMC=0.
        NPAXC=0
        PREVCS(I)=0.
        TREVCS(I)=0.
      PREVC(I)=0.
      TREVC(I)=0.
      DO 520 L=1,10
        COS(I,L)=0.
  520 CC(I,L)=0.
        WRITE(3,840)
        WRITE(3,831)(CRN(L,I),L=1,2)
```

```
      IF(TYPCAR(I).EQ.0)WRITE(3,821)
      WRITE(3,810)
      IF(TYPCAR(I).EQ.3)WRITE(3,800)
      IF(TYPCAR(I).NE.3)WRITE(3,801)
      WRITE(3,810)
      WRITE(3,355)
    LINES=10
C
C
    DO 400 J=1,NFTX
      IF(KLINK(I,J).LT.1 .OR. KLINK(I,J).GT.NLK(I,J))TYPE 997,I,J
     1 ,KLINK(I,J),NLK(I,J)
997   FORMAT(4I3)
    KEQ=IAC(I,J)
    WRITE(3,830)  FID(I,J),AIRCRT(KEQ),NFREQ(I,J)
    WRITE(3,835)
    LINES=LINES+2
    NLINKS=NLK(I,J)
    DISTF(J)=0.
      PAXF(J)=0
    SEATMF=0.
    RPMF=0.
    PREVF(J)=0.
    TREVF(J)=0.
    DO 330 L=1,10
      CFS(J,L)=0.
  330 CF(J,L)=0.
C
C  COMPUTE & PRINT LINK COSTS
C
    DO 400 K=1,NLINKS
    M=NLH(I,J,K)
    N=NKH(I,J,K)
      DST=DISTAN(I,J,K)
      TIMEBL=BTIME(I,J,K)
      RPM=0.
      SEATM=0.
      NPAX=IFIX(PAX(I,J,K)+0.5)
      NDIST=IFIX(DST+0.5)
      NTIME=IFIX(TIMEBL+0.5)
      FLOAD=1000.0
      RPM=PAX(I,J,K)*DST
      IF(KEQ.EQ.6)GO TO 333
    FLOAD=0.0
    DUM=SEATS(KEQ,I)*FACTOR*NFREQ(I,J)
    IF(DUM.EQ.0.0)GO TO 335
    FLOAD=PAX(I,J,K)/DUM
  335 FACT1=(NFREQ(I,J)-NDUM(I,J,K))*FACTOR
    SEATM=SEATS(KEQ,I)*DST*FACT1
333   IF(TYPCAR(I).NE.0)GO TO 332
    PREV=YLD(I)*RPM/1000.0
    TREV=(UCP(I)/100.0+1.0)*PREV
    CL(1)=(UBAC(KEQ,I)*TIMEBL)*FACT1/1000.0
    CL(2)=(UBCC(KEQ,I)*TIMEBL)*FACT1/1000.0
    CL(3)=UATS(I)*FACT1/1000.0
    CL(4)=URS(I)*PAX(I,J,K)/1000.0
    CL(5)=UPC(I)*PREV/100.0
    CL(6)=UCG(I)*0.95*(TREV-PREV)/100.0
    CL(7)=UAP(I)*PREV/100.0
    CL(8)=UIPE(I)*RPM/1000.0
    SUM=0.
```

```
      DO 340 L=1,8
  340 SUM=SUM+CL(L)
      CL(9)=(SUM-(UDEP(REU,I)+UFU(I))*CL(1)/100.0)*UGA(I)/100.0
      CL(10)=SUM+CL(9)
      NPREV=IFIX(PREV+0.5)
      NTREV=IFIX(TREV+0.5)
      DO 380 L=1,10
  380 NC(L)=IFIX(CL(L)+0.5)
      RATIO=0.0
      IF(NC(10).EQ.0)GO TO 332
      RATIO=TREV/CL(10)
332     IFLOAD=100.*FLOAD
C
C  PRINT LINK COSTS
C
      IF(TYPCAR(I).EQ.0)WRITE(3,790) M,N,
     *                    BTIME(I,J,K),NDIST,NPAX,        IFLOAD,NPREV,
     *NTREV,(NC(L),L=1,10),RATIO
        IF(TYPCAR(I).NE.0)WRITE(3,790)M,N,BTIME(I,J,K),NDIST,NPAX,IFLOAD
C
C  ACCUMULATE VALUES FOR FLIGHT TOTALS
C
      IF(NDOM(I,J,K).GT.0)GO TO 370
      RPMF=RPMF+RPM
      SEATMF=SEATMF+SEATM
370     PAXF(J)=PAXF(J)+PAX(I,J,K)
        IF(TYPCAR(I).NE.0)GO TO 381
      PREVF(J)=PREVF(J)+PREV
      TREVF(J)=TREVF(J)+TREV
      DO 380 L=1,10
  380 CF(J,L)=CF(J,L)+CL(L)
381     IF(K.NE.KLINK(I,J))GO TO 400
        ALFF(J)=FLOAD
        DISTF(J)=DST
        PAXFS(J)=PAX(I,J,K)
        RPMC=RPMC+RPM
        SEATMC=SEATMC+SEATM
        IF(TYPCAR(I).NE.0)GO TO 400
        PREVFS(J)=PREV
        TREVFS(J)=TREV
        DO 382 L=1,10
382     CFS(J,L)=CFS(J,L)+CL(L)
  400 CONTINUE
C
C  PRINT FLIGHT TOTALS
C
      ALFFE=1000000.
      IF(SEATMF.LE.1.000)GO TO 410
      ALFFE=RPMF/SEATMF
410     IALFFE=100.*ALFFE
        RPMCA=RPMCA+RPMF
        SEATMA=SEATMA+SEATMF
        IF(TYPCAR(I).NE.0)GO TO 422
      NPREV=IFIX(PREVF(J)+0.5)
      NTREV=IFIX(TREVF(J)+0.5)
      DO 420 L=1,10
  420 NC(L)=IFIX(CF(J,L)+0.5)
      RATIO=0.0
      IF(NC(10).EQ.0)GO TO 422
      RATIO=TREVF(J)/CF(J,10)
  422 IF(TYPCAR(I).EQ.0)WRITE(3,700)IALFFE,NPREV,NTREV,
```

```
     *        (RC(L),L=1,10),RATIO
          IF(TYPCAR(I).NE.0)WRITE(3,760)IALFFE
          IF(TYPCAR(I).NE.0)WRITE(3,761)
        LINES=LINES+LINKS+5
        LLEFT=LINMAX-LINES
        IF(J.EQ.NFTX)GO TO 425
        LREQD=NLK(I,J+1)+13
        IF(LLEFT.GE.LREQD)GO TO 435
  425   LSKIPS=LLEFT-6
        DO 427 L=1,LSKIPS
  427   WRITE(3,399)
        WRITE(3,770)
        IF(J.EQ.NFTX)GO TO 435
          WRITE(3,840)
          WRITE(3,831)(CRN(L,I),L=1,2)
          WRITE(3,832)
          IF(TYPCAR(I).EQ.0)WRITE(3,821)
          WRITE(3,810)
          IF(TYPCAR(I).EQ.0)WRITE(3,800)
          IF(TYPCAR(I).NE.0)WRITE(3,801)
          WRITE(3,810)
          WRITE(3,899)
        LINES=10
C
C  ACCUMULATE VALUES FOR CARRIER TOTALS
C
C
435       IF(TYPCAR(I).NE.0)GO TO 460
          DO 440 L=1,10
          CCS(I,L)=CCS(I,L)+CFS(J,L)
  440   CC(I,L)=CC(I,L)+CF(J,L)
        PREVC(I)=PREVC(I)+PREVF(J)
          PREVCS(I)=PREVCS(I)+PREVFS(J)
        TREVC(I)=TREVC(I)+TREVF(J)
          TREVCS(I)=TREVCS(I)+TREVFS(J)
  460   CONTINUE
C
C  PRINT FLIGHT SUMMARIES & CARRIER TOTALS
C
461       WRITE(3,750)
          WRITE(3,831)(CRN(L,I),L=1,2)
          IF(TYPCAR(I).EQ.0)WRITE(3,821)
          WRITE(3,810)
          IF(TYPCAR(I).EQ.0)WRITE(3,740)
          IF(TYPCAR(I).NE.0)WRITE(3,741)
          WRITE(3,810)
          WRITE(3,899)
        DO 500 J=1,NFTX
        KEQ=IAC(I,J)
          NPAXF=PAXFS(J)
          NPAXC=NPAXC+NPAXF
          IF(TYPCAR(I).NE.0)GO TO 490
        NDIST=DISTF(J)
        NPREV=IFIX(PREVFS(J)+0.5)
        NTREV=IFIX(TREVFS(J)+0.5)
        DO 480 L=1,10
  480   NC(L)=IFIX(CFS(J,L)+0.5)
        RATIO=0.0
        IF(NC(10).EQ.0)GO TO 490
        RATIO=TREVFS(J)/CFS(J,10)
  490     IALFF=100.*ALFF(J)
```

```
      IF(TYPCAR(I).EQ.0)WRITE(3,730)  FID(I,J),NFREQ(I,J),NDIST,
     * AIRCRT(KEQ),IALFF,NPREV,NTREV,
     *       (NC(L),L=1,10),RATIC
        IF(TYPCAR(I).NE.0)WRITE(3,630)FID(I,J),NFREQ(I,J),NDIST
     *  AIRCRT(KEQ),IALFF,NPAXF
  500 CONTINUE
C
C  PRINT CARRIER TOTALS
C
      ALFC(I)=1000000.
      IF(SEATMC.LE.1.000)GO TO 510
      ALFC(I)=RPMC/SEATMC
510     ALFCA(I)=1000000.
        IF(SEATMA.LE.1.000)GO TO 511
        ALFCA(I)=RPMCA/SEATMA
511     IF(TYPCAR(I).NE.0)GO TO 521
      NPREV=IFIX(PREVC(I)+0.5)
        NPREVS=IFIX(PREVCS(I)+0.5)
      NTREV=IFIX(TREVC(I)+0.5)
        NTREVS=IFIX(TREVCS(I)+0.5)
      DO 520 L=1,10
        NCS(L)=IFIX(CCS(I,L)+0.5)
  520 NC(L)=IFIX(CC(I,L)+0.5)
      RATIC=0.0
      IF(NC(10).EQ.0)GO TO 530
      RATIC= TREVC(I)/CC(I,10)
        RATIOS=TREVCS(I)/CCS(I,10)
521     IALFC=100.*ALFC(I)
        IALFCA=100.*ALFCA(I)
  530 IF(TYPCAR(I).EQ.0)WRITE(3,720)  IALFCA,NPREV,NTREV,
     *       (NC(L),L=1,10),RATIC
        IF(TYPCAR(I).EQ.0)WRITE(3,721)IALFC,NPREVS,NTREVS,
     *    (NCS(L),L=1,10),RATIOS
        IF(TYPCAR(I).NE.0)WRITE(3,620)IALFC,NPAXC
        WRITE(3,722)RPMC,RPMCA,SEATMC,SEATMA
C
C  ACCUMULATE VALUES FOR ALL CARRIERS
C
      SEATMT=SEATMT+SEATMC
      RPMT=RPMT+RPMC
  540 CONTINUE
C
C  CARRIER SUMMARIES
C
      NFT(ITOTAL)=0
      PREVCS(ITOTAL)=0.
      TREVCS(ITOTAL)=0.
      DO 543 L=1,10
  543 CCS(ITOTAL,L)=0.
      DO 550 I=1,NCAR
      IF(NFT(I).EQ.0)GO TO 550
      NFT(ITOTAL)=NFT(ITOTAL)+NFT(I)
      PREVCS(ITOTAL)=PREVCS(ITOTAL)+PREVCS(I)
      TREVCS(ITOTAL)=TREVCS(ITOTAL)+TREVCS(I)
      DO 545 L=1,10
  545 CCS(ITOTAL,L)=CCS(ITOTAL,L)+CCS(I,L)
  550 CONTINUE
      ALFC(ITOTAL)=1000000.
      IF(SEATMT.LE.1.000)GO TO 555
      ALFC(ITOTAL)=RPMT/SEATMT
555     WRITE(3,715)
```

```
      WRITE(3,820)
      WRITE(3,810)
      WRITE(3,710)
      WRITE(3,810)
      DO 560 I=1,ITOTAL
        IF(NFT(I).EQ.0)GO TO 560
        IF(TYPCAR(I).NE.0 .AND. .NOT.(I.EQ.ITOTAL))GO TO 570
      NPREV=IFIX(PREVCS(I)+0.5)
      NTREV=IFIX(TREVCS(I)+0.5)
      DO 560 L=1,10
  560 NC(L)=IFIX(CCS(I,L)+0.5)
      RATIO=0.0
      IF(NC(10).EQ.0)GO TO 570
      RATIO= TREVCS(I)/CCS(I,10)
  570 IF(I.EQ.ITOTAL)WRITE(3,698)
        IALFC=100.*ALFC(I)
      IF(TYPCAR(I).EQ.0 .OR. I.EQ.ITOTAL)WRITE(3,700)(CRN(M,I),M=1,2),
     *  NFT(I),              IALFC,NPREV,NTREV,
     *       (NC(L),L=1,10),RATIO
        IF(TYPCAR(I).NE.0 .AND. .NOT.(I.EQ.ITOTAL))WRITE(3,700)
     1  (CRN(M,I),M=1,2),NFT(I),IALFC
  560 CONTINUE
C
C
C
620     FORMAT(//' CARRIER TOTALS',10XI5,'%',3XI8)
630     FORMAT(1XI4,I3,'/WK',I5,3XA5,1XI5,'%',3XI7)
  700 FORMAT (1H0 2A8,1X,I3,4X,I5,'%',12(1X,I7),1X,F5.2)
  710 FORMAT (1H  T67,'A/C &',T105,'OTHER',T129,'TOTAL'/
     *  T15,'# OF', T36,'PAX' ,T43,'TOTAL',T59,'CABIN TRAFFIC  RESERV.
     #PAX     CARGO  AD. &  PAX EX- GENERAL   TOTAL REV./'/T5,'CARRIER',
     *  T17,'FLIGHTS',T28,          'L.F.    REV.    REV.  A/C OP.   CREW
     * SERVICE  & SALES  COMM.  COMM. PUBLIC. PENSES  & ADMIN   COSTS T
     #OTAL'/T129,'COSTS')
  715 FORMAT (1H1  T59,'CARRIER SUMMARIES'/T59,17(1H=))
  720 FORMAT (//' CARRIER TOTALS(ALL LEGS)', I5,'%',2(1XI9),4(1XI6)
     1 ,6(1XI7),1XF5.2)
  721 FORMAT(//'   (OVER-WATER LEGS ONLY)',I5,'%',2(1XI9),4(1XI6)
     1 ,6(1XI7),1XF5.2)
  722 FORMAT (1H ,'PAX-MILES',2XF11.0,' (OVER-WATER)',5XF11.0
     1  ,' (ALL LEGS)'/1H ,'SEAT-MILES',1XF11.0,' (OVER-WATER)',5XF11.0
     2  ,' (ALL LEGS)')
  730 FORMAT (1X I4,I3,'/WK',I5,3X,A5,1X,I5,'%',12(1X,I7),1X,F5.2)
  740 FORMAT (1H ,T2,'FLIGHT',T68,'A/C &',T105,'OTHER',T129,'TOTAL'/
     * T5,'#',T8,'FREQ',T14,'DIST.',T20,'A/C',T28,'L.F.',T36,'PAX' ,T43
     #,'TOTAL',T61,'CABIN ','TRAFFIC  RESERV.  PAX  CARGO  AD.&    PAX '
     #,'EX- GENERAL  TOTAL  ','REV./'/T15,'(M)',T36,'REV.    REV.   A/C'
     #,' OP.    CREW',' SERVICE  & SALES  COMM. COMM. PUBLIC. PENSES  &'
     #,' ADMIN  COSTS  ','TOTAL'/T129,'COSTS')
741     FORMAT(1H ,T2,'FLIGHT'/T5,'#',T8,'FREQ',T14,'DIST',T20
     1  ,'A/C',T28,'L.F.',T36,'TOTAL'/T15,'(M)',T36,'PAX')
  750 FORMAT  (1H1 T50,'FLIGHT SUMMARIES AND CARRIER TOTALS'/T50,
     *  35(1H=))
  770 FORMAT (     //' FOOT NOTES: 1.  ALL REVENUES AND COSTS ARE '
     * ,'ONE-WAY AND COVER ENTIRE PERIOD.'/
     * 13X,'2.  FOR INDIVIDUAL FLIGHT TABLES, LOAD FACTOR FOR A FLIGHT '
     * ,'IS BASED ON ALL LINKS IN ROUTE, '/
     * 13X,'WHILE FOR FLIGHT SUMMARY TABLES, LOAD FACTOR FOR A FLIGHT '
     * ,'IS THAT OF ITS OVER-WATER LEG.')
  780 FORMAT ( /' FLIGHT TOTALS',11X,I5,'%',12(1X,I7),1X,F5.2///)
  781 FORMAT (///)
```

```
  790 FORMAT (1H  A3,1X,A5,F4.1,A3,I6,I3,'@',I2(1XI7),1X,F5.2)
  800 FORMAT (1H  T88,'A/C &',T106,'OTHER',T129,'TOTAL'/T12,'TIME',T17
     *,'DIST.',T23,'#PAX',T29,'L.F.',T36,'PAX' ,T43,'TOTAL',T59,'CABIN'
     *,'  TRAFFIC  RESERV.  PAX     CARGO  AD. &  PAX EX- GENERAL  TOTA'
     *,'L REV.'/'/T3,'LINK',T11,'(HRS)',T16,'(M)',T36,'REV.    REV.   '
     *,'A/C OP.  CREW  SERVICE  & SALES  COMM.   COMM. PUBLIC. PENSES'
     *,'   ADMIN.  COSTS TOTAL'/T129,'COSTS')
  801     FORMAT(/T12,'TIME',T17,'DIST',T23,'#PAX',T29,'L.F.'/T3,'LINK'
     1  ,T11'(HRS)',T18'(M)'/)
  810 FORMAT  (1H 152(1H-))
  815 FORMAT (1H+  '(CONTINUED)')
  820 FORMAT (1H  T88,'ALL REVENUES AND COSTS IN THOUSANDS OF DOLLARS')
  821 FORMAT (1H+ T88,'ALL REVENUES AND COSTS IN THOUSANDS OF DOLLARS')
  830 FORMAT (1H 'FLIGHT : ',I4,5X,'EQUIPMENT : ',A5,5X,I2,'/WEEK')
  831 FORMAT (1H 'CARRIER : ',2A6)
  832 FORMAT (1H+ T28,'(CONTINUED)')
  835 FORMAT (1H  46(1H-))
  840 FORMAT  (1H1 T51,'FLIGHT/LINK REVENUES AND COSTS'/T51,30(1H=))
  845 FORMAT (1H 10I7)
  850 FORMAT (1H+ T115,'@DOMESTIC FLIGHTS@')
  860 FORMAT (1H+,4A6, T103,'@EQUIPMENT@')
  870 FORMAT (1H+ T92,'@STATIONS@')
  880 FORMAT (1H+,10A6, T82,'@CARRIER@')
 881     FORMAT(1H+,10I6,T82,'@CARRIER@')
 882     FORMAT(1H+,10F6.2,T82,'@CARRIER@')
  885 FORMAT (1H  T75,'ERRORS:')
  891 FORMAT  (1H ,A3,1X,I5,1X,A5,2X,I2,2X,3(/1X4(1XI6,1X2F6.1,1XI4)))
  890 FORMAT  (1H  A3,1X,A5,1X,I4,2X,I2,2X,11(A4,I2))
  898 FORMAT  (//)
  899 FORMAT (1H )
  960 FORMAT (20X,10I5)
  980 FORMAT  (2F10.3)
  970 FORMAT  (     A2,1X,A4,1X,A3,2X,I2,2X,11(A3,I2))
  989 FORMAT  (1H ,A3)
  990 FORMAT  (A3)
  998 FORMAT (1H 18A4)
  999 FORMAT  (18A4)
        CLOSE(UNIT=5)
        STOP
      END
```

BSCNOD.CTL

```
.COPY USEU.DAT=USNEU.DAT
.COPY EUUS.DAT=EUNUS.DAT
.EXE SCENAR
.COPY UEDSC.DAT=UENDIS.DAT
.EXE [3036,265]SCODFL
.COPY WATER..LEG<244>=WATERS.LEG
.R SORT
PTD2:BSCNOD.DAT<255>=SCODRT.DAT/RECORD:68/KEY:8:5/KEY:48:1/KEY:14:5/KEY
.PROTECT <255>BSCNOD.DAT
```

FLT.CTL

```
00100   .EXE USFLT,CONSEG,READLL,JETAPT,NCROM,IADATE,FLTLEG,LEGDAY,REUDLL,JE AP
00200   .EXE UEFLT,CONSEG,READLL,JETAPT,NCROM,IADATE,FLTLEG,LEGDAY,REUDLL,JEUAP
00300   .EXE EUFLT,CONSEG,READLL,JETAPT,NCROM,IADATE,FLTLEG,LEGDAY,REUDLL,JEUAP
00400   .COPY PT49:FLT.DAT=USFLT.DAT
00500   .EXE PRDIS
00600   .COPY PT49:USPR.DAT=PR.DAT
00700   .COPY PT49:USDIS.DAT=DIS.DAT
00800   .COPY PT49:FLT.DAT=UEFLT.DAT
00900   .EXE PRDIS
01000   .COPY PT49:UEPR.DAT=PR.DAT
01100   .COPY PT49:UEDIS.DAT<255>=DIS.DAT
01200   .COPY PT49:FLT.DAT=EUFLT.DAT
01300   .EXE PRDIS
01400   .COPY PT49:EUPR.DAT=PR.DAT
01500   .COPY PT49:EUDIS.DAT=DIS.DAT
01700   .R SORT
01800   PT49:UEPEU.DAT=PT49:EUPR.DAT,PT49:/TEMP,PT49:/TEMP/RECORD:107/KEY:22:5/
01900   PT49:ARREU.DAT=PT49:EUPR.DAT,PT49:/TEMP,PT49:/TEMP/RECORD:107/KEY:16:5/
02000   PT49:UEPUE.DAT=PT49:UEPR.DAT,PT49:/TEMP,PT49:/TEMP/RECORD:107/KEY:22:5/
02100   PT49:ARRUE.DAT=PT49:UEPR.DAT,PT49:/TEMP,PT49:/TEMP/RECORD:107/KEY:16:5/
02200   PT49:UEPUS.DAT=PT49:USPR.DAT,PT49:/TEMP,PT49:/TEMP/RECORD:107/KEY:22:5/
02300   PT49:ARRUS.DAT=PT49:USPR.DAT,PT49:/TEMP,PT49:/TEMP/RECORD:107/KEY:16:5/
```

HFILES, CTL

```
00100   .R SORT
00200   USHFIL.DAT=HFILE.DAT/RECORD:48/KEY:1:6/KEY:37:6
00300   EUHFIL.DAT=HFILE2.DAT/RECORD:48/KEY:16:6/KEY:37:6
00400   .EXE USHFLR
00500   .EXE UPUDPU
00600   .EXE PDUPDU
00700   .EXE EUHFLR
00800   .EXE UPEDPE
00900   .EXE PDEPDE
01000   .EXE USEUHP
01100   .EXE EUPUSP
01200   .COPY PT52:SUSEU.DAT=ARRUE.DAT
01300   .EXE USPEUP
01400   .EXE UEHNC
```

BSCØOD.CTL

```
.COPY USEU.DAT=USPEUP.DAT
.COPY EUUS.DAT=EUPUSP.DAT
.EXE SCENAR
.COPY UEDSC.DAT=UEDIS.DAT
.EXE [3000,265]SCODFL
.COPY WATER.LEG<244>=WATERS.LEG
.R SDRT
PT49:BSCOOD.DAT<255>=SCODRT.DAT/RECORD:68/KEY:6:5/KEY:48:1/KEY:14:5/KEY
.PROTECT <255>BSCOOD.DAT
```