Box 259N Route 4
Fredericksburg, Virginia 22405
November 13, 1984

Hewlett Packard Company
Attention: Users' Library
1000 N.E. Circle Blvd.
Corvallis, Oregon 97330

Dear Sirs:

Enclosed is my latest contribution to the HP Users' Library. Please consider this program as my entry to the present contest. I have submitted numerous programs in the past but never had one ready during a contest. Fortunately, this time I can enter.

A couple of programs are already in the library for schedule planning, and I purchased them before undertaking the present program. After reviewing them, I felt that the HP-41C was capable of supporting a more complete program for schedule planning. The requirements for such a program are more than just being able to compute the schedule statistics for a fairly complex project. Data entry should be very easy, and verification of the entry is important, especially on a long-running program. Projects are dynamic; they often last many months and individual jobs finish ahead of schedule or fall behind. Thus schedules need to be updated, and the activities on the critical path have to be redetermined so that a project can finish as timely as possible. A good program should have the ability to perform this task easily - without the user having to reenter all the data. Finally, since a schedule may have to be updated a month or more from when it was originally developed, a method of saving the input data is required so that the computer's memory can be freed up for other tasks. The current program was designed to meet these additional requirements and to be very user friendly.

You will find that the submittal package includes two page 8s, containing the sample problem solution, done in two formats. One format is the same as the User Instructions, that is, Input, Function, and then Display. The other is in the library's current format of Display, Input, and then Function. Use which ever one you prefer and remove the other.

Sincerely yours,

Gary Goodman

41-03134

PROGRAM SUBMITTAL

☒ New Program ☐ Revision to Program

Model No. ☐ 67 ☐ 97 ☒ 41

Program Title Project Planning and Scheduling (PERT Method)

No. Lines 299 & 90 Bytes 574 & 212 Category No. D150 (Primary)

Primary Category Name Forcasting/Planning

Abstract-75 Word Maximum This program uses PERT methods to calculate the project length and schedule statistics (early start, late start, total float & free float) for each of the activities comprising the project. Extended memory is used to store the activity data. An editing capability allows the user to review or change this data to update the schedule as activities are completed or slippages occur. Maximum project size is 108 activities, more with additional extended memory modules. A utility program is included to transfer project files to/from cards.

Necessary Accessories: Printer, one Memory Module (minimum), Extended Functions Module

Name Gary Goodman

Company (If Applicable)

Address Box 259N Route 4

City Fredericksburg State/Province Virginia Postal Code 22405

Phone Number (703) 775-3260 Country U.S.A.

Hewlett Packard Dealer

Acceptance Choice: ☐ FOUR $6.00 PROGRAMS ☒ ONE POINT CERTIFICATE

Please use the checklist below to insure submittal of all proper program documents

☒ SIGNED PROGRAM SUBMITTAL PAGE

☒ Program Description I

☒ Program Description II

☒ User Instructions

☒ Program Listing(s)

☒ MASS STORAGE MEDIA

☒ Registers, Status...

☐ Keyboard, Card Labeling (optional)

ACKNOWLEDGMENT AND AGREEMENT

To the best of my knowledge, I have the right to contribute this program material without breaching any obligation concerning nondisclosure of proprietary or confidential information of other persons or organizations. I am contributing this program material on a nonconfidential nonobligatory basis to Hewlett-Packard Company ("HP") for inclusion in its program library, and I agree that HP may use, duplicate, modify, publish, and sell the program material, and authorize others to do so without obligation or liability of any kind. HP may publish my name and address, as the contributor, to facilitate user inquiries pertaining to this program material.

Signature Gary Goodman Date Nov. 12, 1984

HP USE ONLY	No. Pages: 23	Mag Cards Cassette: 4	Price:

Program Title: Project Planning and Scheduling (PERT Method)

Contributor's Name: Gary Goodman

Address: Box 259N Route 4

City: Fredericksburg **State:** Virginia **Zip Code:** 22405

Program Description:

This program uses PERT methods to calculate the project length and schedule statistics (early start, late start, total float, and free float) for each of the activities comprising the project.

The input data consists of the "from" and "to" node numbers which designate each activity, and the activity's expected duration. The data is kept in an ASCII file in extended memory. This file is then used by the program to compute the total project length, schedule statistics, and critical path. The program has a simple edit facility which allows the user to review, change, delete, or add new data to reflect the completion or slippage of activities as the project progresses. Thus an updated schedule can be computed without the necessity of reentering all the data.

The program can accomodate a project containing up to 108 activities, or even more with the addition of Extended Memory Modules.

A utility program is provided to transfer project files between extended memory and magnetic cards.

(Continued on next page)

Necessary Accesories: Printer, one Memory Module (minimum), and the Extended Functions/Memory Module.

Operating Limits and Warnings: Node numbers are limited to values between 0 and 99. Activity durations must be integers, and total project length must not exceed 998.

Reference: Elwood S. Buffa, Modern Production Management, 4th edition, John Wiley & Sons, Inc., New York, N.Y., 1973.

Acknowledgement is given to Edwin R. Schmeckpeper for his excellent program: Critical Path Method for Project Planning and Scheduling, HP Users' Library 41-01309-1.

Overview

The PERT method of schedule planning is based on the development of a network representation of the required activities like that shown in the PERT network diagram in Program Description II.

In the diagram, the arrows represent the required activities coded by the letters. The numbers near the arrows are the estimated activity durations. The numbered circles are called nodes and are used to define the beginning and end points of each activity. The directions of the arrows represent the order in which activities follow each other; their length has no significance. The network diagram also shows the required precedence relationships of the activities and which activities can proceed simultaneously.

The dashed lines represent dummy activities. No activity is associated with them, and their duration is zero. Dummy activities serve two purposes. The first is to insure the correct representation in the arrow diagram of the actual precedence requirements. For example, in the diagram, activity **t** has s as its immediate predecessor, while activity **r** has **n** and **s** as its immediate predecessors. The other purpose is to provide a unique set of node numbers for each activity. In the network diagram, activities l and **m** can both start at the completion of **k**, and both are immediate predecessors to activity **s**. The dummy activity, **11→12**, provides different nodes for activities l and **m**.

A project may have multiple starting nodes but only one ending node, signifying the completion of the project.

The node numbering, as shown in the diagram, has been done in a particular way. The numbers for every arrow are progressive, and no backtracking through the network is allowed. This is a requirement of the computational algorithm used and also prevents the occurence of closed loops.

The results of the PERT algorithm are the determination of the scheduling data for each activity so that the project may be completed in the shortest possible time, subject to the constraints of the precedence relationships. These data are the early start time(**ES**) and late start time(**LS**), respectively. For each activity there is also an early finish time(**EF**) and a late finish time(**LF**), calculated by adding the activity's duration(D) to its corresponding start time. The project length is the minimum time required to complete the project and is given by the difference between the latest of the early finish times and the earliest of the early start times.

Some other very useful results may be obtained from these data. The difference between the late start time and the early start time for an activity is the "slack", or total float time(**TF**). If this difference is zero, the activity is said to be critical. The total float time is the amount of slippage that may occur in the scheduled completion (early finish) for the activity without delaying the completion of the project. Note that slack does not "belong" to a particular activity but is shared with other activities along the same path; i.e., slippages in other activities on the path use up some of the total float time. Activities which are critical lie along the critical path. These activities must begin and end on schedule. The critical path is the longest time path through the network. Free float time(**FF**) is another quantity of interest. It is the amount of time between an activity's early finish time and the next activity's early start time. People working on an activity with free float, will be "free" from the time their activity completes until the next activity can be started.

Note: This entire discussion of the PERT method of schedule planning assumes that all activities which can proceed simultaneously will do so. In other words, enough resources (e.g., people) must be available to do the work in parallel, as shown by the schedule. A short discussion and further references to the problem of scheduling with limited resouces is given by Buffa (see reference).

Processing

The program, "PERT", prompts for the file name which is to contain the project activity data (the "from" node, "to" node, and duration). If the file exists, the program assumes that you wish to review or edit the file and then recompute the schedule statistics. If the file is new, the program prompts you for the number of activities in the project so that it can allocate sufficient file space. (The actual number of activities can be less than you allocate space for.) The last record of the file contains a single "*" signifying the end of data.

If the file is new, the program will prompt you to enter the data in the form: **"from" node** [ENTER] **"to" node** [ENTER] **duration** (only integer values, please). Your entry will be formatted on the display in the form **ff-tt D=ddd.** This gives you a chance to verify the data before it is accepted. A bad entry can be cancelled and you will be prompted to reenter the data. Once the data are accepted, they are compressed to save file space and inserted in the file as a 7 digit integer in the form **fftt ddd.** If the printer is used during this phase, which is highly recommended, the input data will be echoed to the printer when it is accepted, helping to insure that all the activities have been entered before sorting and computation begin. A null entry signals the program that all the data has been entered.

If the file is old, the program will review each entry on the display, as described above, and give you a chance to accept it or delete it. If you delete it, you will be prompted to reenter corrected data, which will be inserted in the file. You may choose to ignore this prompt and proceed on to the next record, in which case the record remains deleted. When all the records have been reviewed, the program prompts you for new data. The printer may be used in the review/edit phase as described in the input phase above.

After all the data has been entered, the computer sorts the file. This insures that the records are ordered according to the precedence relationships set forth in the scheduling requirements. For a project containing many activities, the sort may take over an hour; the printer may be turned off to save batteries. You will be BEEPed when the printer is again required.

After the records have been sorted, the process of computing the early start and late finish times is begun.

Up to this point, only five data registers have been needed. The remaining processing requires six registers plus an additional register for each node, from zero to the highest numbered node in the project. The program finds the number of this node in the "to" node portion of the last data record in the file and then automatically sets the SIZE if necessary.

These new registers will contain the early start time and late finish time for the node, compressed in the form **est.lft.** The program initializes these registers to 1.999, the earliest possible early start time and the latest possible late finish time. (The project is assumed to start on day 1.)

The process of computing the early start time for each node is quite straightforward. The early start time for a node is the latest of the early finish times, **Max.(ES + D)**, for the node's immediate predecessors. The program reads the activity file in forward (normal) order, computing the early finish time for each activity. It then compares this time with the early start time previously stored under the "to" node, and if larger, substitutes the new value. As each activity is being processed, the early start time for its "to" node is being computed. This process is repeated until all records have been read and the early start time for the last node has been computed.

Remember, the early start time for a node is the same as the latest early finish time for all its immediate predecessors. Thus the early start time for the last node is the same as the earliest possible finish time for the project, and it is also the late finish time for its immediate predecessors. Thus for the last node, **lft = est.**

The late finish time for each node is computed in a similar manner (only in reverse) to the early start time. The late finish time for a node is the earliest of the late start times, **Min.(LF - D)**, for the node's immediate succeeders. After setting the late finish time of the last node equal to its early start time, the program reads the file backwards from the last record, computing the late start time for each activity. It then compares this time with the late finish time previously stored under the "from" node, and if smaller, substitutes the new value. At the end of this process, the late finish times for all the nodes will be computed.

When these computations are finished, the computer will BEEP you to signal that it is ready to start printing the results (and to remind you to turn the printer back on, if you had turned it off).

The tabulated results are calculated as follows:

PROJECT LENGTH = **est** of last node - **1,**

ES = **est** of "from" node,

LS = **lft** of "to" node - **D,**

TF = **LS** - **ES,**

FF = **est** of "to" node - **ES** - **D.**

User Options

A. During an Edit/Review session, the user may skip reviewing the entire file and jump directly into the sort phase. This is useful when only the beginning activities in a file are modified.

B. If the file is already sorted by node number, the user may skip the sort phase altogether, thus saving considerable time. It is the user's responsibility to insure that the file is sorted. This is useful when activities are modified without changing their order.

Saving Project Files on Magnetic Cards

The utility program, "XFER", allows you to transfer project files between extended memory and magnetic cards. Project files may be quite long and will probably need to be modified as activities are completed or slippages occur so an updated schedule can be produced. "XFER" saves the project file on magnetic cards, thereby freeing up extended memory. When a new schedule needs to be produced, "XFER" can restore the file.

As soon as "XFER" is loaded into memory, it will prompt you for the project file name that you wish to save on cards. A null entry signals the program that you wish to restore a project file to extended memory.

If you are saving a file, the program will prompt you to enter the required number of unprotected cards, and after the file has been saved on the cards, it will prompt you to verify them (using the card reader's VER function). The file name, file size, and required SIZE are recorded on the cards along with all of the activity data. After the file is saved, you may manually purge it to free up space in extended memory.

If you are restoring a file, the program will prompt you to enter only the first track of the saved file. The first track contains the file name, file size, and memory size needed to restore the file. You will be prompted to verify the file name and size. (At this point, you may manually rename the file or alter its size.) You verify that you are ready to restore the file by keying [R/S]. The program will then prompt you to enter all the data cards, including the first track. The file is now restored and is ready for use by the "PERT" program.

Beyond its basic requirements of two data registers, "XFER" computes the number of registers needed to transfer the files and sets the SIZE if necessary.

Definitions of Inputs and Outputs

PROJECT ? Input, the name of the file for the project's activity data.

N? Input, the maximum number of activities for this project.

F ↑ T ↑ D Input, "from" node, [ENTER], "to" node, [ENTER], duration for each activity.

PROJECT: Output, same as in PROJECT ? above.

LENGTH: Output, the calculated minimum length of time required to complete the project.

JOB Output, the activity, specified by its "from" and "to" nodes.

D Output, the activity's estimated duration.

ES Output, the calculated early start time for the activity.

LS Output, the calculated late start time for the activity.

TF Output, the calculated total float time for the activity, the amount of slack available in the starting time.

FF Output, the calculated free float time for the activity, the amount of time available between an activity's early finish time and the following activity's early start time.

CP Output, in place of TF and FF to indicate the activity lies on the critical path, i.e., has no slack.

Time Requirements

Approximate running time from last entry until printing starts:

$T \simeq .0075N^2 + .1N$ minutes, with the printer off.

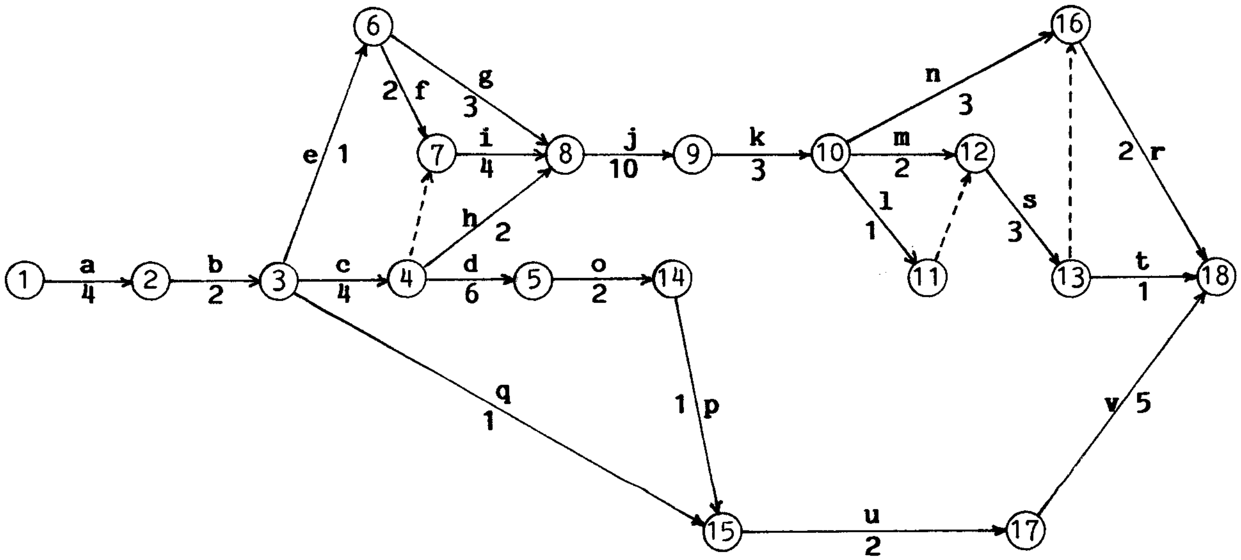
The program runs about 13% slower if the printer is on.

Sample Problem House Construction Project

Precedence Chart Showing Sequence of Activities and Required Times to Finish a House

Job No.	Description	Immediate Predecessors	Time (days)
a	Excavate and pour footings	-	4
b	Pour concrete foundation	a	2
c	Erect wooden frame including rough roof	b	4
d	Lay brickwork	c	6
e	Install basement drains and plumbing	b	1
f	Pour basement floor	e	2
g	Install rough plumbing	e	3
h	Install rough wiring	c	2
i	Install heating and ventilating	c,f	4
j	Fasten plaster board and plaster (including drying)	g,h,i	10
k	Lay finishing flooring	j	3
l	Install kitchen fixtures	k	1
m	Install finish plumbing	k	2
n	Finish carpentry	k	3
o	Finish roofing and flashing	d	2
p	Fasten gutters and downspouts	o	1
q	Lay storm drains for rain water	b	1
r	Sand and varnish flooring	n,s	2
s	Paint	l,m	3
t	Finish electrical work	s	1
u	Finish grading	p,q	2
v	Pour walks and complete landscaping	u	5

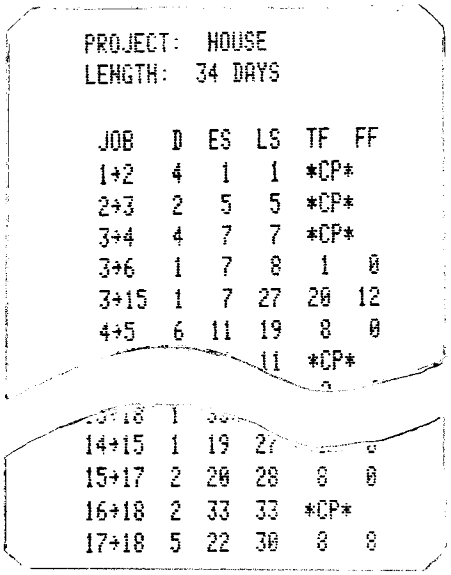
PERT Network Diagram



SOLUTION:

Display	Keystrokes (SIZE 005)	Comments
	[XEQ] "PERT"	
PROJECT ?	**HOUSE** [R/S]	Project file name
N?	25 [R/S]	Number of activities
F ↗ T ↗ D	1 [ENTER] 2 [ENTER] 4 [R/S]	Input data for activity **a**
1-2 D=4	[R/S]	Echo of input, if ok, accept
F ↗ T ↗ D	2 [ENTER] 3 [ENTER] 2 [R/S]	Input data for activity **b**
2-3 D=2	[R/S]	Echo of input, if ok, accept
F ↗ T ↗ D	3 [ENTER] 4 [ENTER] 4 [R/S]	Input activity **c**
3-4 D=4	[R/S]	Echo and accept
F ↗ T ↗ D	4 [ENTER] 5 [ENTER] 0 [R/S]	BAD ENTRY for activity **d**
4-5 D=0	[←] [←] [R/S]	Delete entry
F ↗ T ↗ D	4 [ENTRY] 5 [ENTER] 6 [R/S]	Correct entry for activity **d.**
4-5 D=6	[R/S]	Echo and accept
F ↗ T ↗ D	4 [ENTER] 7 [ENTER] 0 [R/S]	Input for dummy activity **4→7**
4-7 D=0	[R/S]	Echo and accept
•		
•		
•		
F ↗ T ↗ D	17 [ENTER] 18 [ENTER] 5 [R/S]	Input for final activity **v**
17-18 D=5	[R/S]	Echo and accept
F ↗ T ↗ D	[R/S]	Null entry signals end of data.

Wait about 8 minutes for calculations to complete, signaled by (**BEEP**).

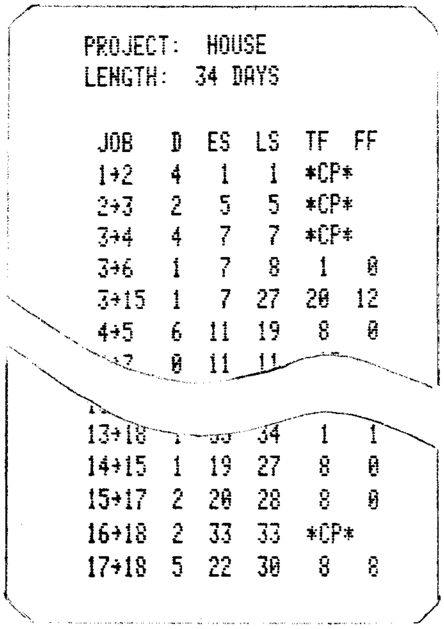


```
PROJECT:  HOUSE
LENGTH:  34 DAYS

 JOB    D  ES  LS  TF  FF
 1→2    4   1   1  *CP*
 2→3    2   5   5  *CP*
 3→4    4   7   7  *CP*
 3→6    1   7   8   1   0
 3→15   1   7  27  20  12
 4→5    6  11  19   8   0
                11  *CP*

14→15   1  19  27
15→17   2  20  28   8   0
16→18   2  33  33  *CP*
17→18   5  22  30   8   8
```

SOLUTION:

Keystrokes (SIZE 005)	**Display**	**Comments**
[XEQ] "PERT"	PROJECT ?	Prompt for project file name
HOUSE [R/S]	N?	Prompt for number of activities
25 [R/S]	F ↑ T ↑ D	Prompt for activity data
1 [ENTER] 2 [ENTER] 4 [R/S]	1-2 D=4	Echo of input data for activity a
[R/S]	F ↑ T ↑ D	Accept & prompt for next activity
2 [ENTER] 3 [ENTER] 2 [R/S]	2-3 D=2	Echo for activity **b**
[R/S]	F ↑ T ↑ D	Accept & prompt for next activity
3 [ENTER] 4 [ENTER] 4 [R/S]	3-4 D=4	Echo for activity c
[R/S]	F ↑ T ↑ D	
4 [ENTER] 5 [ENTER] 0 [R/S]	4-5 D=0	BAD ENTRY for activity **d**
[←] [←] [R/S]	F ↑ T ↑ D	Delete entry and reprompt.
4 [ENTRY] 5 [ENTER] 6 [R/S]	4-5 D=6	Correct entry for activity **d.**
[R/S]	F ↑ T ↑ D	
4 [ENTER] 7 [ENTER] 0 [R/S]	4-7 D=0	Echo for dummy activity 4→7
[R/S]	F ↑ T ↑ D	
•		
•		
•		
17 [ENTER] 18 [ENTER] 5 [R/S]	17-18 D=5	Echo for final activity **v**
[R/S]	F ↑ T ↑ D	Accept & prompt for next activity
[R/S]		Null entry signals end of data.

Wait about 8 minutes for calculations to complete, signaled by (**BEEP**).



```
PROJECT:  HOUSE
LENGTH:  34 DAYS

 JOB    D  ES  LS  TF  FF
 1→2    4   1   1  *CP*
 2→3    2   5   5  *CP*
 3→4    4   7   7  *CP*
 3→6    1   7   8   1   0
 3→15   1   7  27  20  12
 4→5    6  11  19   8   8
 4→7    0  11  11

13→18   1  33  34   1   1
14→15   1  19  27   8   0
15→17   2  20  28   8   0
16→18   2  33  33  *CP*
17→18   5  22  30   8   8
```

USER INSTRUCTIONS

Produce Schedule Statistics

SIZE: (HP-41C) ≥ 005

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
1	Load program "PERT".		[GTO]..	PACKING
2	Connect the printer and set it to manual mode. (The printer is required for step 11 and is recommended for all steps.)			
3	Initialize the program.		[XEQ] "PERT"	PROJECT ?
4	Input the name of the project (maximum 6 characters).		[R/S]	
	Depending on whether the file is new or old, there are two possible responses.			
4a	New file: Prompt for number of activities.			N?
4b	Old file: The activity data for the first record is displayed.			ff-tt D=ddd
	Go to step 10.			
5	Input the maximum number of activities for the project including all dummy activities.	N	[R/S]	F ↗ T ↗ D
6	Repeat steps 7 and 8 for each activity to be added to the file. At end go to step 9.			

USER INSTRUCTIONS

SIZE:
(HP-41C)

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
7	Input: "from" node number,	ff	[ENTER]	ff
	"to" node number,	tt	[ENTER]	tt
	estimated duration of activity.	ddd	[R/S]	
	The activity is echoed for review.			ff-tt D=ddd
8	Review entry. You may accept it or cancel it.			
8a	Accept activity as entered. (Activity data will be printed.)		[R/S]	
	Display will prompt for next activity. Return to step 7.			F ↗ T ↗ D
8b	Cancel bad entry.		[←] [←] [R/S]	
	Display will prompt for reentry. Return to step 7.			F ↗ T ↗ D
9	Signal end of data entry.		[R/S]	⊢ (goose)
	Go to step 13.			
10	Review and Edit: Repeat steps 11 and 12 for each record which is echoed at the display.			
11	Review activity data. You may keep it, delete it outright, or edit it by first deleting it and then substituting corrected data.			

USER INSTRUCTIONS

SIZE:
(HP-41C)

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
11a	Keep activity as displayed.		[R/S]	
	(Activity data will be printed.) If			
	more activities are on file, the			
	computer will display the next activity.			ff-tt D=ddd
	Return to step 11.			
	If all activities on file have been			
	reviewed, you will be prompted to add			
	new activities.			F ↑ T ↑ D
	Return to step 6.			
11b	Delete the activity.		[←] [←] [R/S]	
	You will be prompted to replace the			
	deleted data with corrected data.			F ↑ T ↑ D
12	You may leave the activity deleted or			
	replace it with corrected data.			
12a	Leave deleted and proceed with next			
	activity.		[R/S]	ff-tt D=ddd
	Return to step 11.			
12b	Replace with corrected data:			
	Input "from" node number,	ff	[ENTER]	ff
	"to" node number,	tt	[ENTER]	tt
	estimated duration of activity.	ddd	[R/S]	
	The activity is echoed for review.			ff-tt D=ddd
	Return to step 11.			
13	Processing: Sorting and calculations			
	may take many minutes. An estimate of			

USER INSTRUCTIONS

SIZE:
(HP-41C)

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
	the time required for processing may be calculated as shown on page 6. The printer may be turned off if desired.			
14	When the computer is ready to print, it will signal you with a BEEP. If the printer is off, you must now turn it on and press [R/S].			
	OPTIONS			
	The following options may be used when recomputing a schedule from an edited file.			
	Option A allows you to skip the review of all the records in a file and to proceed directly from step 11 to step 13.			
	To choose Option A:		[XEQ] "A"	⁊- (goose)
	Go to step 13.			
	Option B allows you to skip the review of all the records in a file and also to skip the lengthy sorting process. This option is useful if you know the			

USER INSTRUCTIONS

				SIZE: (HP-41C)
STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
	file is still in sorted order after your editing. (If your records are not sorted, this option will give you wrong answers.) It may be used anytime following the start of step 11.			
	To choose Option B:		[XEQ] "B"	
	Go to step 14.			

Error Messages

NO ROOM

During step 5 means there is not enough room in extended memory to hold the project file. You can purge unneeded files and restart the program at step 3.

During step 13 means that there are not enough registers available in main memory to continue. The number of registers needed is 7 more than the highest numbered node. You may clear other programs from memory, then key [XEQ] "PACK", [GTO] "PERT", [XEQ] "B" to continue.

END OF FL

During step 8 means that you have tried to enter more activities than you had originally specified in step 5. If this is caused by mistakenly having entered some activities more than once, reexecute the program, review the entries, delete erroneous entries and add the remaining entries to the end. If the error is caused by miscounting the number of activities, you must purge the project file and restart the program at step 3. An alternative solution is to use "XFER" to first save the project file on magnetic cards and then to restore it using a larger file size. See the User Instructions for "XFER".

PROGRAM LISTING

□ 67 □ 97 ☒ 41C

STEP/LINE | KEY ENTRY | KEY CODE (67/97 only) | COMMENTS

```
01◆LBL "PER           INITIALIZATION
T"
02 "PROJECT
?"
03 AON
04 STOP
05 AOFF
06 ASTO 04
07 ADV
08 CF 29
09 FIX 0
10 SF 25
11 0                 Check to see if
12 SEEKPTA           project file
13 FS?C 25           already exists.
14 GTO 01
15 " N?"             If file does not
16 PROMPT            exist, prompt for
17 8                 number of activ-
18 *                 ities, compute
19 9                 required file
20 +                 size, create
21 7                 file, put an "*"
22 /                 at the end of the
23 CLA               file, and set the
24 ARCL 04           record pointer at
25 CRFLAS            the beginning of
26 "*"               the file.
27 APPREC
28 0
29 SEEKPT
30◆LBL 01            GET NEXT RECORD
31 SF 25
32 GETREC
33 FC?C 25           At End-of-File,
34 GTO A             go to Sort.
35◆LBL 02            ECHO ON DISPLAY
36 CF 22
37 ANUM              If no more activ-
38 FC?C 22           ity records, go
39 GTO 03            to Prompt for New
40 XEQ 15            Activity Data.
41 CLA
42 ARCL Z
43 "├-"              Format echo
44 ARCL Y            display
45 "├ D="
46 ARCL X
```

```
47 1                 Leave 1 in X-reg.
48 PROMPT
49 X=0?              Test for delete
50 DELREC
51 X=0?              If deleted,prompt
52 GTO 03            for replacement,
53 FS? 21            otherwise echo to
54 PRA               printer.
55 GTO 01
56◆LBL 03              PROMPT FOR NEW
57 " F ↑ T              ACTIVITY DATA
↑ D"
58 PROMPT
59 FC?C 22
60 GTO 01
61 1 E3
62 /
63 +
64 1 E2              Code entered
65 /                 activity data and
66 +                 insert it in the
67 1 E5              project file.
68 *
69 CLA
70 ARCL X
71 INSREC
72 GTO 02
73◆LBL A              INITIALIZE SORT
74 ADV                    PROCESS
75 "*"
76 POSFL
77 RCLPT             No. of activities
78 INT
79 DSE X
80 1 E3
81 /
82 ISG X
83 STO 01            Outer loop cntl.
84◆LBL 04
85 RCL 01
86 INT
87 STO 02            Inner loop cntl.
88 SEEKPT
89 GETREC            Get sort record
90 DELREC
91 0
92 SEEKPT
93 ANUM
```

Note: Refer to "HP-41C OWNER'S HANDBOOK AND PROGRAMMING GUIDE" for specific information on keystrokes. The Function Index is found at the very back of the Handbook. Refer to Appendix E in 67 or 97 "OWNER'S HANDBOOK AND PROGRAMMING GUIDE" for exact keystrokes.

PROGRAM LISTING

□ 67 □ 97 ☒ 41C

STEP/ LINE	KEY ENTRY	KEY CODE (67/97 only)	COMMENTS
94	◆LBL 05		
95	GETREC		Comparison rec.
96	ANUM		If sort record is
97	X<>Y		less than compar-
98	X<Y?		ison rec., go to
99	GTO 06		Insert Sort Rec.
100	DSE 02		
101	GTO 05		
102	GETREC		
103	◆LBL 06		INSERT SORT
104	CLA		RECORD
105	ARCL X		
106	INSREC		
107	ISG 01		
108	GTO 04		
109	◆LBL B		CALCULATIONS
110	"*"		
111	POSFL		
112	RCLPT		No. of activities
113	INT		
114	DSE X		Record pointer
115	STO 00		for last activity
116	SEEKPT		
117	GETREC		Get last activity
118	SIZE?		
119	ANUM		
120	1 E5		
121	/		
122	FRC		
123	1 E2		
124	*		
125	INT		Highest node no.
126	7		
127	+		
128	X>Y?		
129	PSIZE		Set SIZE
130	DSE X		
131	.005		
132	+		
133	1.999		
134	◆LBL 07		Initialize all
135	STO IND Y		est.lft to 1.999
136	DSE Y		
137	GTO 07		
138	0		Initialize for
139	SEEKPT		forward pass
140	CF 22		through file.

STEP/ LINE	KEY ENTRY	KEY CODE (67/97 only)	COMMENTS
141	◆LBL 08		FORWARD PASS
142	GETREC		
143	ANUM		
144	FC?C 22		
145	GTO 09		
146	XEQ 15		
147	RCL IND 01		est of "from" node
148	+		est + D
149	RCL IND 02		est of "to" node
150	X<>Y		
151	X>Y?		
152	STO IND 02		Update est of "to" node
153	GTO 08		
154	◆LBL 09		lft OF LAST NODE
155	RCL IND 02		
156	INT		est
157	STO 05		Project Length +1
158	.1		
159	%		
160	+		
161	STO IND 02		est.lft = est.est
162	◆LBL 10		BACKWARD PASS
163	RCL 00		Record pointer
164	X<0?		If finished,
165	GTO 11		go to Print.
166	SEEKPT		
167	GETREC		
168	ANUM		
169	XEQ 15		
170	1 E3		
171	/		
172	CHS		.ddd = D x 10^{-3}
173	RCL IND 02		est.lft of "to" node
174	FRC		
175	+		.lft' = .lft-.ddd
176	RCL IND 01		est of "from" node
177	INT		
178	+		est.lft'
179	RCL IND 01		est.lft of "from" node
180	X<>Y		

Note: Refer to "HP-41C OWNER'S HANDBOOK AND PROGRAMMING GUIDE" for specific information on keystrokes. The Function Index is found at the very back of the Handbook. Refer to Appendix E in 67 or 97 "OWNER'S HANDBOOK AND PROGRAMMING GUIDE" for exact keystrokes.

PROGRAM LISTING

☐ 67 ☐ 97 ☒ 41C

STEP/LINE | KEY ENTRY | KEY CODE (67/97 only) | COMMENTS

```
181 X<Y?
182 STO IND                Update .lft of
01                         "from" node
183 1
184 ST- 00
185 GTO 10
186♦LBL 11                 - PRINT RESULTS -
187 BEEP
188 SF 21
189 "PROJECT
:  "
190 ARCL 04                Project name
191 PRA
192 FIX 0
193 "LENGTH:
 "
194 DSE 05
195 ARCL 05                Project length
196 "⊦ DAYS"
197 PRA
198 ADV
199 " JOB
D  ES  L"                  Print column
200 "⊦S   TF               headings
 FF"
201 PRA
202 0                      Reset pointer
203 SEEKPT
204 CF 22
205♦LBL 12                 PRINT ACTIVITY
206 GETREC                     LINE
207 ANUM
208 FC?C 22                When done, go to
209 GTO 16                 End of Program.
210 XEQ 15
211 2
212 R↑                     "from" node
213 XEQ 14
214 125
215 ACCHR                  "→"
216 CLX
217 9
218 R↑                     "to" node
219 CLA
220 ARCL X
221 X<=Y?
222 "⊦ "
223 ACA
```

STEP/LINE | KEY ENTRY | KEY CODE (67/97 only) | COMMENTS

```
224 3
225 RCL 03                 duration (D)
226 XEQ 14
227 4
228 RCL IND
01
229 INT                    est of activity
230 STO 00
231 XEQ 14
232 4
233 RCL IND
02
234 FRC
235 1 E3
236 *                      lft of activity
237 RCL 03
238 -                      lst = lft-D
239 ST- 00                 est-lst
240 XEQ 14
241 4
242 RCL 00                 Total Float (-TF)
243 X≠0?                   If not critical,
244 GTO 13                 go to Calc. Float
245 "   *CP*"
246 ACA
247 PRBUF
248 GTO 12
249♦LBL 13                  CALCULATE FLOAT
250 CHS
251 XEQ 14
252 4
253 RCL IND
02
254 INT                    est of "to" node
255 RCL 03                 D
256 -
257 RCL IND                est of "from"
01                         node
258 INT
259 -                      Free Float (FF)
260 XEQ 14
261 PRBUF
262 GTO 12                 - - - - - - - - -

00
```

PROGRAM LISTING

□ 67 □ 97 ☒ 41C

STEP/ LINE	KEY ENTRY	KEY CODE (67/97 only)	COMMENTS
263	♦LBL 14		COLUMN ALIGNMENT
264	CLA		SUBROUTINE
265	ARCL X		Input:
266	CLX		R_y: no. of cols.
267	ALENG		R_x: number
268	-		
269	SKPCHR		Output:
270	ACA		Number in print
271	RTN		buffer_ _ _ _ _ _
272	♦LBL 15		DECODE SUBROUTINE
273	6		
274	STO 01		
275	STO 02		
276	CLX		
277	1 E5		
278	/		Input:
279	INT		R_x: ffttddd
280	ST+ 01		
281	LASTX		Output:
282	FRC		R_z: ff
283	1 E2		R_y: tt
284	*		R_x: ddd (D)
285	INT		R_1: ff node addr.
286	ST+ 02		R_2: tt node addr.
287	LASTX		
288	FRC		
289	1 E3		
290	*		
291	STO 03		
292	RTN		
293	♦LBL 16		END OF PROGRAM
294	ADV		
295	ADV		
296	ADV		
297	ADV		
298	ADV		
299	END		- - - - - - - - - -

STATUS

SIZE	:	n + 7, where n is the highest numbered node. "PERT" sets the SIZE automatically. However, a minimum SIZE of 005 is necessary to start.
Program Size	:	574 Bytes (82 Registers)
ASCII File Size	:	(8N + 9)/7, where N is the number of activities. "PERT" automatically creates the proper sized file.

DATA REGISTERS

00	Record pointer; Used for total float calculation
01	"from" node address; Outer loop control for sort
02	"to" node address; Inner loop control for sort
03	Activity duration (D)
04	Project file name
05	Project length
06	est.lft for node 0
07	est.lft for node 1
.	
.	
.	
n+6	est.lft for node n

FLAGS USED

21	Set	:	Printing required
	Clear	:	Printing not desired
22	Set	:	Number entered; Record contains activity data
	Clear	:	Null entry; Record contains "*"
25	Set	:	Project file exists; Not End-of-File
	Clear	:	File must be created; End-of-File
29	Clear	:	No decimal point or commas

USER INSTRUCTIONS

Transfer Project File To or From Magnetic Cards

SIZE: ≥002
(HP-41C)

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
1	Load program "XFER" Program starts automatically after loading. (If the program is already in memory, key [XEQ] "XFER".)			FILE TO CRDS PROJECT ?
2	Do you wish to save a project file on magnetic cards or to restore it to extended memory?			
2a	To save a project: Enter the project's name.	name	[R/S]	RDY 01 OF nn
2b	To restore a project:		[R/S]	CRDS TO FILE INSERT TRK 1
	Go to step 6.			
3	Insert the requested number of tracks using unclipped cards. After all the cards have been written:			VERIFY
4	Reinsert the cards to verify that the data has been written properly. After all the cards have been verified:		[←]	DONE
5	You may now purge the project file from extended memory and clear "XFER" from main memory.			

USER INSTRUCTIONS

SIZE:
(HP-41C)

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
6	Insert track 1 of the card set			
	containing the saved project.			project name
7	You may restore the file to its			
	original form or you may rename it			
	and/or increase its file size.			
7a	To restore file to its original form:		[R/S]	CARD
7b	To change its name and/or file size,			
	enter its new name in the			
	ALPHA-register and/or its new file			
	size in the X-register.		[R/S]	INSERT CARDS
8	Insert all the cards containing the			
	project file (including track 1).			DONE
9	You may now clear "XFER" from main			
	memory.			

PROGRAM LISTING

□ 67 □ 97 ☒ 41C

STEP/LINE KEY ENTRY KEY CODE (67/97 only) COMMENTS

```
 01♦LBL "XFE              INITIALIZATION
R"
 02 "FILE TO
 CRDS"
 03 AVIEW
 04 CF 23
 05 PSE
 06 "PROJECT
 ?"
 07 AON
 08 PROMPT            Enter project
 09 AOFF              name to save on
 10 FC? 23            cards.  Otherwise,
 11 GTO 02            restore file.
 12 ASTO 00            FILE TO CARDS
 13 FLSIZE
 14 STO 01
 15 "*"
 16 POSFL
 17 SIZE?
 18 RCLPT             No. of activities
 19 2
 20 +
 21 INT               Compute SIZE
 22 X>Y?
 23 PSIZE
 24 DSE X
 25 1 E3              .001 x Number
 26 /                 of last register
 27 ST+ 01
 28 2
 29 +
 30 0                 Initialize
 31 SEEKPT            record pointer.
 32♦LBL 01             GET NEXT RECORD
 33 GETREC
 34 CLX
 35 ANUM              Transfer file to
 36 STO IND           main memory
Y
 37 ISG Y
 38 GTO 01
 39 RCL 01
 40 FRC
 41 WDTAX             Write data cards
 42 "VERIFY"
 43 AVIEW
 44 "DONE"
 45 ASTO X
 46 PSE
 47 VER               Verify cards
```

```
 48♦LBL 02                CARDS TO FILE
 49 "CRDS TO
 FILE"
 50 AVIEW
 51 PSE
 52 1 E-3
 53 "INSERT
TRK 1"
 54 AVIEW             Read R0 and R1
 55 RDTAX             from track 1
 56 FIX 0
 57 CF 29
 58 CLA
 59 ARCL 00           File name
 60 RCL 01            File size
 61 PROMPT
 62 SIZE?
 63 RCL 01
 64 FRC
 65 1 E3
 66 *                 No. of last reg.
 67 1
 68 +                 Compute SIZE
 69 X>Y?
 70 PSIZE
 71 R↑                File size
 72 CRFLAS
 73 "INSERT
CARDS"
 74 AVIEW             Read all data
 75 RDTA              cards
 76 RCL 01
 77 FRC
 78 2
 79 +                 Loop control
 80♦LBL 03
 81 CLA
 82 ARCL IND          Transfer regis-
 X                    ters to ASCII
 83 APPREC            file
 84 ISG X
 85 GTO 03
 86 "*"               Append a "*" at
 87 APPREC            the end of the
 88 "DONE"            file
 89 PROMPT
 90 .END.
```

Note: Refer to "HP-41C OWNER'S HANDBOOK AND PROGRAMMING GUIDE" for specific information on keystrokes. The Function Index is found at the very back of the Handbook. Refer to Appendix E in 67 or 97 "OWNER'S HANDBOOK AND PROGRAMMING GUIDE" for exact keystrokes.

STATUS, REGISTERS, FLAGS

STATUS

SIZE : N + 2, where N is the number of activities. "XFER" sets the SIZE automatically. However, a minimum SIZE of 002 is necessary to start.

Program Size : 212 Bytes (31 Registers)

ASCII File Size : The file size is recorded on track 1 of the cards when the file is saved. You may check the size by viewing the X-register at step 7 of the User Instructions.
"XFER" automatically creates the proper sized file.

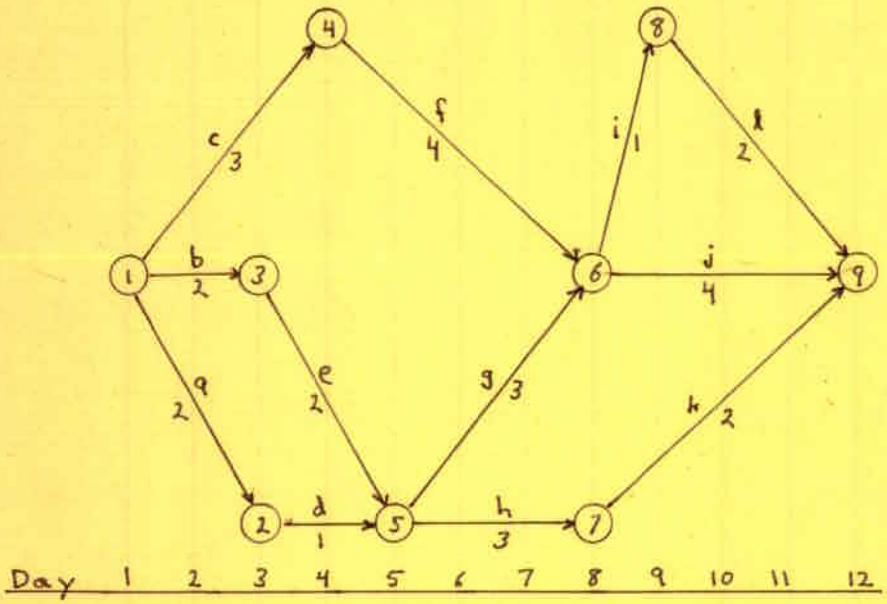
DATA REGISTERS

00	Project file name
01	File size + .001 x Number of the last register
02	ffttddd for first activity
03	ffttddd for second activity
.	
.	
.	
N+1	ffttddd for activity N

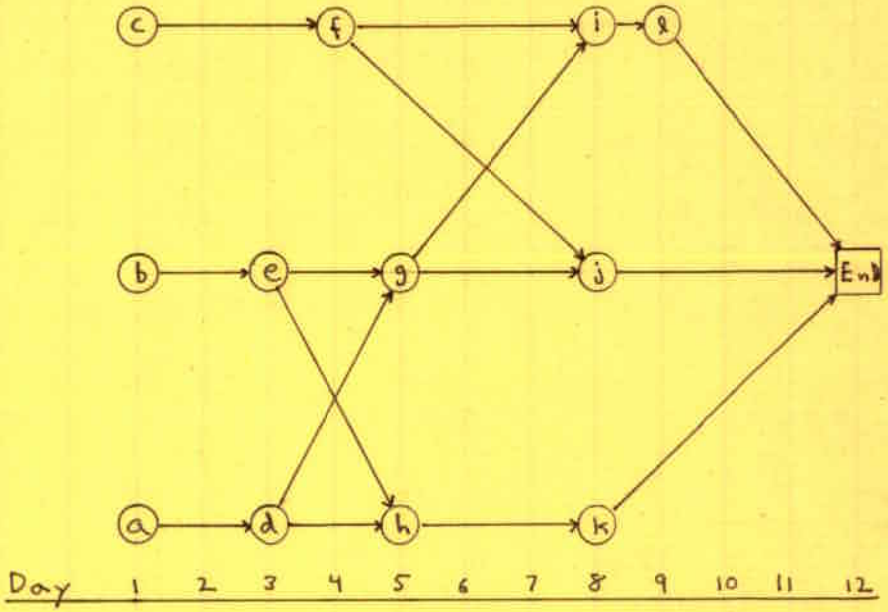
FLAGS USED

23	Set :	Project name entered, transfer file to cards
	Clear :	Null entry, transfer cards to file
29	Clear :	No decimal point or commas

PERT CHART



CPM CHART



Task	Pred.
a	e, j
b	k
c	d
d	f
e	d, h, [illegible]
f	–
g	k
h	f
i	a, g, l
j	c, h
k	f
l	b, e

CPM
Level 0

1. Initialize program. Check if project file exists and create it if it does not

2. Input, Review, Edit, and Echo data

3. Sort data file so all preceding jobs come first

4. Initialize Early Start, Late Finish and earliest early start of all succeeding jobs as 1.999999

5. Calculate Early Start for each job (forward pass)

6. Calculate Late Finish for each job (backward pass) and earliest ES of all succeeding jobs (same backward pass)

7. Calculate Total Float = LF - ES - D
   Calculate Free Float = ES of succeeding job - ES - D

8 Print table of results
   JOB DUR ES LF TF FF

Running Time for HOUSE project - ~~12~~ 11½ minutes

CPM

1 Initialize Program

1.1 Prompt for project file name

1.2 Check if project file exists. If it does exist
set the file pointer to 0

1.3 If the project file does not exist, prompt for the
required file size and create the file

1.4 Set the Alpha mode to on in preparation for the
Input, Review, & Exit phase.

CPM

2 Input, Review, Edit, and Echo data

2.1 Display the next record if it exists (not EOF) or display "JOB DUR, PRED" if at EOF. This is a prompt for entering more data

2.2 If the EOF has been reached and no new data has been entered, go to the SORT phase

2.3 if the EOF has been reached (and new data has been entered, append the record to the end of the file and Echo the input at the printer.

2.4 If (the EOF has not been reached) but the record has been changed or cancelled, delete the record. If the record has been changed, Insert the new record. Echo the input at the printer

2.5 Echo the input at the printer

2.5.1 If the printer is connected and a new or edited record is available, echo it to the printer

2.5.2 Return to 2.1

CPM

3 Sort the data file so all preceeding jobs come first

3.1 Initialize for sorting. Turn the alpha mode off and reset the file pointer to the beginning of the file

3.2 Search the file for the next preceeding job (PRED) value. If there are no more preceeding job values, Go to step 4

3.3 From this point search the file to the end to see if the preceeding job follows.

3.4 If it does not follow then it must preceed and thus be in proper order. Go to step 3.2

3.5 If it does follow, then it is not in proper order. Move the job (delete and insert) just before the record containing the current PRED value. This puts the record in proper order. Go to step 3.2

- CPM

3.2 Search the file for the next PRED value

3.2.1 Find the next comma. Commas are used to deliniate PRED values.

3.2.2 If no more commas are found, then there are no more PRED values. All records are sorted. Go to step 4

3.2.3 Increment record pointer one character past the comma

3.2.4 Get the record from the comma to the end of record.

3.2.5 Separate out the PRED value with subroutine

CPM

3.3 Search remaining part of file to see if preceeding job follows

3.3.1 Append a space to the end of PRED value

3.3.2 Search the remaining part of the file to see if the PRED value can be found in the JOB section of the following records.

3.33 If it is found check to see that it starts in the first column indicating that the preceeding job has been found and is out of sorted order. Go to step 3.5

3.3.4 If it does not start in the first column, then the PRED value has been found contained in a different job. It has not been found. Increment the file pointer one character and go to step 3.3.2.

CPM

3.5 Preceeding job follows PRED value. Move job ahead.

3.5.1 Get the record containing the preceeding job

3.5.2. Delete this record from the file

3.5.3 Reset the record pointer to the beginning of the record containing the PRED value

3.5.4 Insert the preceeding job record ahead of the Record containing the PRED value. The record pointer now points to the beginning of the inserted record.

3.5.5 Go to step 3.2

CPM

4 Initialize one register for each job to 1.999999
The integer portion will hold the early start time for the job
The first 3 decimal digits will hold the late finish for the job
The second (last) 3 decimal digits will hold the earliest
early start for the immediate succeeding jobs

4.1 Find the number of jobs by retrieving the record pointer
which will be on the last record.

4.2 Set the size so that there is one register for each job
+ enough scratch registers at the beginning of register storage
If the size is already large enough do not change it

4.3 Initialize one register for each job to 1.999999

CPM

5 Calculate the early start for each job
(Read the file in a forward direction)
The early start is the greatest of the ES + D = EF
value of all a job's predecessors

5.1 Reset the ~~file~~ primary record pointer to the beginning of the file.

5.2 Using the primary record pointer get the job name and store it. Also get the early start (ES) for the job and the duration (D). Compute the early finish (EF) for the job and store it. If no more records exist go to step 6

5.3 Increment the primary record pointer by one, ie to the next job.

5.4 Search for the next record containing this job as a predecessor. If it is not found go to step 5.2

5.5 If it is found, check to see that the JOB name and PRED value are identical. If they are not, continue searching the file. Go to step 5.4

5.6 If they are identical, compare the EF value computed in step 5.2 with the ES for record containing the job as a predecessor. If the EF is larger, store it as the new ES value for this record. Go to step 5.4.

CPM

5.2 Get the jobname, early start, and duration of the next job

5.2.1 Use the stored primary record pointer to set the file pointer
If the EOF has been reached go to step 6.

5.2.2 Get the next record

5.2.3 Separate out the job name with subroutine
Store the job name. Reset the file pointer to the beginning of the record

5.2.4 Obtain the early start (ES) for the job. The address of the early start is the primary record pointer + an offset of –.

5.2.5. Search for the space preceeding the duration and get the record from that point on.

5.2.6 Obtain the duration and add it to the early start time to give the early finish (EF) for the job.

CPM

5.4 Search for the next record containing the JOB name in the PRED value.

5.4.1 Form ", Jobname" in the ALPHA register and search the remaining records for this value

5.4.2 If it is not found, there are no more records listing this as a predecessor. Go to step 5.2

CPM

5.5 Check that the job name and the found PRED value are dentical

5.5.1 Increment the position of the found PRED value to the character following the comma. This is neccessary to continue the search to the next record containing the PRED value.

5.5.2 Get the record, remove the leading comma and separate out the PRED value using subroutine

5.5.3 Check to see if the PRED value and the jobname store at step 5.2.3 are identical. If they are not, continue searching the file. Go to step 5.4

CPM

6 Calculate the Late Finish Time (LF) for each job and find the earliest of the ES for all succeeding jobs (backward pass) Note: these will be coded in the first 3 and second 3 decimal digits respectively - LF = min LS of all succeeding jobs

6.1 Initialize LF and min ES for the backwards pass. Set LF and min ES for the last job equal to ES + D.
Note: the primary line pointer is one past the last record.

6.2 Decrement the primary record pointer to the previous record.
If the primary record pointer is zero, go to step 7.

6.3 Obtain and save the ES for this job Also compute the Late Start LS = LF - D for this job and save it

6.4 Obtain the next PRED value on this record. Save the PRED value. If there are no more PREDs, go to step 6.2

6.5 Find the job record matching the PRED value (it must exist)

6.6 Compare the LF for this record with the LS calculated in step 6.3 Store the minimum of these as the new LF. Compare the min ES for this record with the ES stored in step 6.3. Stire the minimum of these as the new min ES. Go to step 6.4

PROGRAM LISTING

Page of

□ 41C

STEP/LINE | KEY ENTRY | KEY CODE (67/97 only) | COMMENTS

```
01♦LBL "CPM
"                          INITIALIZATION
02 "PROJECT
?"
03 AON
04 STOP
05 ASTO 00                 Project Name
06 SF 25
07 0
08 SEEKPTA
09 FS?C 25                 If file exists
10 GTO 00                  Go To INPUT
11 AOFF
12 "SIZE ?"
13 PROMPT
14 CLA
15 ARCL 00
16 CRFLAS
17 AON
18♦LBL 00                  INPUT, REVIEW, & EDIT
19 CF 23
20 SF 25
21 "JOB DUR
, PRED"
22 GETREC
23 STOP
24 FC? 23  }
25 FS? 25  }               If (No more entries
26 FS? 30  }               and at EOF)
27 GTO 02  }               Go To SORT
28 ALENG
29 FC? 25                  If EOF
30 APPREC
31 FC? 25
32 GTO 01                  If EOF skip
33 FC? 23                  else
34 X=0?                    } If edited or
35 DELREC                  } cancelled
36 FS? 23                  If edited
37 INSREC
38♦LBL 01
39 FS? 55                  If (printer is
40 X=0?                    connected &
41 X<0?                    record exists)
42 PRA                     then print
43 GTO 00
```

50

```
44♦LBL 02                  SORT
45 ADV
46 AOFF                    Initialize for
47 0                       SORT
48 STO 01                  Primary Rec. ptr.
49♦LBL 03
50 RCL 01     }
51 SEEKPT     }            Find Next comma
52 ","        }
53 POSFL      }
54 X<0?                    If not found
55 GTO 05                  Go to Calculations
56 1 E-3      }            Increment
57 +          }            primary ptr
58 STO 01     }            past comma
59 SEEKPT
60 GETREC
61 44 ←ATOX ?              } Remove leading
62 XEQ 20                  } and trailing comma,
63 "⊢ "                    obtain PRED
64♦LBL 04
65 POSFL
66 X<0?
67 GTO 03
68 INT
69 LASTX
70 1 E-3
71 ST+ Z
72 +
73 X≠Y?
74 SEEKPT
75 X≠Y?
76 GTO 04
77 GETREC
78 DELREC
79 RCL 01
80 INT
81 STO 01
82 SEEKPT
83 INSREC
```

Think about this (the comma is probably not there)

00

Note: Refer to "HP-41C OWNER'S HANDBOOK AND PROGRAMMING GUIDE" for specific information on keystrokes. The Function Index is found at the very back of the Handbook. Refer to Appendix E in 67 or 97 "OWNER'S HANDBOOK AND PROGRAMMING GUIDE" for exact keystrokes

PROGRAM LISTING

Page of

□ 41C

STEP/ LINE	KEY ENTRY	KEY CODE (67/97 only)	COMMENTS	STEP/ LINE	KEY ENTRY	KEY CODE (67/97 only)	COMMENTS

```
 84 GTO 03
 85◆LBL 05
 86 SIZE?
 87 RCL 01
 88 INT
 89 11
 90 +
 91 X>Y?
 92 PSIZE
 93 .991
 94 -
 95 1.999999
 96◆LBL 06
 97 STO IND
Y
 98 DSE Y
 99 GTO 06
100 SF 28
101 CF 29
102 0
103 STO 01
104◆LBL 07
105 RCL 01
106 SF 25
107 SEEKPT
108 FC?C 25
109 GTO 09
110 GETREC
111 32
112 XEQ 20
113 ASTO 02
114 RCL 01
115 SEEKPT
116 10
117 +
118 RCL IND
X
119 STO 03
120 " "
121 POSFL
122 GETREC
123 ANUM
124 ST+ 03
125 1
126 ST+ 01
```

50

```
127◆LBL 08
128 ","
129 ARCL 02
130 POSFL
131 X<0?
132 GTO 07
133 1 E-3
134 +
135 SEEKPT
136 GETREC
137 SEEKPT
138 44
139 XEQ 20
140 ASTO X
141 RCL 02
142 X≠Y?
143 GTO 08
144 RCLPT
145 10
146 +
147 RCL IND
X
148 RCL 03
149 X>Y?
150 STO IND
Z
151 GTO 08
152◆LBL 09
153 9
154 +
155 RCL IND
X
156 INT
157 RCL 03
158 INT
159 STO 02
160 1 E3
161 /
162 .1
163 %
164 +
165 +
166 STO IND
Y
```

00

PROGRAM LISTING

Page of

□ 41C

STEP/LINE | KEY ENTRY | KEY CODE (67/97 only) | COMMENTS

```
167♦LBL 10
168 RCL 01
169 INT
170 1
171 -
172 SEEKPT
173 X=0?
174 GTO 13
175 STO 01
176 10
177 +
178 RCL IND
X
179 INT
180 STO 03
181 LASTX
182 FRC
183 1 E3
184 ST/ 03
185 *
186 INT
187 STO 04
188 " "
189 POSFL
190 GETREC
191 ANUM
192 ST- 04
193♦LBL 11
194 RCL 01
195 SEEKPT
196 ","
197 POSFL
198 INT
199 X<=Y?
200 X<0?
201 GTO 10
202 LASTX
203 1 E-3
204 +
205 STO 01
206 SEEKPT
207 GETREC
208 44
209 XEQ 20
210 "├ "
211 0
```

STEP/LINE | KEY ENTRY | KEY CODE (67/97 only) | COMMENTS

```
212♦LBL 12
213 SEEKPT
214 POSFL
215 INT
216 LASTX
217 1 E-3
218 ST+ Z
219 +
220 X≠Y?
221 GTO 12
222 10
223 +
224 RCL IND
X
225 FRC
226 1 E3
227 *
228 INT
229 LASTX
230 FRC
231 RCL 03
232 X<Y?
233 X<>Y
234 RDN
235 X<>Y
236 RCL 04
237 X<Y?
238 X<>Y
239 RDN
240 +
241 1 E3
242 /
243 X<> IND
Y
244 INT
245 ST+ IND
Y
246 GTO 11
```

Note: Refer to "HP-41C OWNER'S HANDBOOK AND PROGRAMMING GUIDE" for specific information on keystrokes. The Function Index is found at the very back of the Handbook. Refer to Appendix E in 67 or 97 "OWNER'S HANDBOOK AND PROGRAMMING GUIDE" for exact keystrokes.

PROGRAM LISTING

Page of

□ 41C

STEP/ LINE	KEY ENTRY	KEY CODE (67/97 only)	COMMENTS	STEP/ LINE	KEY ENTRY	KEY CODE (67/97 only)	COMMENTS

```
247♦LBL 13
248 BEEP
249 SF 21
250 FIX 0
251 "PROJECT
:  "
252 ARCL 00
253 PRA
254 "LENGTH:
 "
255 DSE 02
256 ARCL 02
257 "⊢ DAYS"
258 PRA
259 ADV
260 " JOB
D  ES  L"
261 "⊢S   TF
 FF"
262 PRA
263♦LBL 14
264 SF 25
265 GETREC
266 FC?C 25
267 GTO 16
268 32
269 XEQ 20
270 ACA
271 8
272 ALENG
273 -
274 RCLPT
275 INT
276 SEEKPT
277 10
278 +
279 " "
280 POSFL
281 GETREC
282 RCL Z
283 ANUM
284 STO 05
285 XEQ 30
286 RCL IND
Z
287 INT
288 STO 03
```

50

```
289 LASTX
290 FRC
291 1 E3
292 *
293 INT
294 STO 04
295 LASTX
296 FRC
297 1 E3
298 *
299 STO 02
300 4
301 RCL 03
302 XEQ 30
303 4
304 RCL 04
305 RCL 05
306 -
307 XEQ 30
308 4
309 RCL 04
310 RCL 03
311 -
312 RCL 05
313 -
314 X≠0?
315 GTO 15
316 "   *CP*"
317 ACA
318 PRBUF
319 GTO 14
320♦LBL 15
321 XEQ 30
322 4
323 RCL 02
324 RCL 03
325 -
326 RCL 05
327 -
328 XEQ 30
329 PRBUF
330 GTO 14
331 RTN
```

00

Note: Refer to "HP-41C OWNER'S HANDBOOK AND PROGRAMMING GUIDE" for specific information on keystrokes. The Function Index is found at the very back of the Handbook. Refer to Appendix E in 67 or 97 "OWNER'S HANDBOOK AND PROGRAMMING GUIDE" for exact keystrokes.

PROGRAM LISTING

Page of

□ 41C

STEP/ LINE | KEY ENTRY | KEY CODE (67/97 only) | COMMENTS | STEP/ LINE | KEY ENTRY | KEY CODE (67/97 only) | COMMENTS

```
332♦LBL 20
333 ASTO 05
334 CLA
335 ARCL 05
336 POSA
337 X<0?
338 RTN
339 AROT
340 CHS
341 ALENG
342 +
343♦LBL 21
344 ATOX
345 RDN
346 DSE X
347 GTO 21
348 RTN
349♦LBL 30
350 CLA
351 ARCL X
352 CLX
353 ALENG
354 -
355 SKPCHR
356 ACA
357 RTN
358♦LBL 16
359 ADV
360 ADV
361 ADV
362 ADV
363 ADV
364 END
```

PARSE Subroutine

Parse all characters to the left of the character specified in the x-register

PROGRAM LISTING

Page of

□ 41C

STEP/ LINE KEY ENTRY KEY CODE (67/97 only) COMMENTS

```
 01◆LBL "XFE
R-AS"
 02 "FILENAM
E?"
 03 AON
 04 STOP
 05 AOFF
 06 ASTO 00
 07 0
 08 SF 25
 09 SEEKPTA
 10 FC?C 25
 11 GTO 06
 12 .4
 13 STO 00
 14 SF 17
 15◆LBL 01
 16 CLA
 17◆LBL 02
 18 SF 25
 19 ARCLREC
 20 FC?C 25
 21 GTO 05
 22◆LBL 03
 23 6
 24 ALENG
 25 X<Y?
 26 GTO 04
 27 ISG 00
 28 ASTO IND
 00
 29 ASHF
 30 GTO 03
 31◆LBL 04
 32 FS? 17
 33 GTO 02
 34 255
 35 XTOA
 36 ISG 00
 37 ASTO IND
 00
 38 GTO 01
```

50

STEP/ LINE KEY ENTRY KEY CODE (67/97 only) COMMENTS

```
 39◆LBL 05
 40 FIX 0
 41 RCL 00
 42 INT
 43 " "
 44 ARCL 00
 45 "⊢ REG"
 46 1 E3
 47 /
 48 STO 00
 49 WDTAX
 50 AVIEW
 51 PSE
 52 "VERIFY"
 53 AVIEW
 54 "DONE"
 55 ASTO X
 56 PSE
 57 VER
 58◆LBL 06
 59 "FILESIZ
E?"
 60 PROMPT
 61 CLA
 62 ARCL 00
 63 CRFLAS
 64◆LBL 07
 65 RDTA
 66 SIZE?
 67 RCL 00
 68 1 E3
 69 *
 70 1
 71 +
 72 X>Y?
 73 PSIZE
 74 X>Y?
 75 GTO 07
 76 1
 77 ST+ 00
 78 SF 01
```

00

Note: Refer to "HP-41C OWNER'S HANDBOOK AND PROGRAMMING GUIDE" for specific information on keystrokes. The Function Index is found at the very back of the Handbook. Refer to Appendix E in 67 or 97 "OWNER'S HANDBOOK AND PROGRAMMING GUIDE" for exact keystrokes

PROGRAM LISTING

Page of

□ 41C

STEP/ LINE	KEY ENTRY	KEY CODE (67/97 only)	COMMENTS	STEP/ LINE	KEY ENTRY	KEY CODE (67/97 only)	COMMENTS

```
 79◆LBL 08
 80 SF 17
 81 CLA
 82 ARCL IND
00
 83 255
 84 POSA
 85 X<0?
 86 GTO 09
 87 CF 17
 88 -1
 89 AROT
 90 ATOX
 91◆LBL 09
 92 FC? 01
 93 APPCHR
 94 FS?C 01
 95 APPREC
 96 FC? 17
 97 SF 01
 98 ISG 00
 99 GTO 08
100 CF 01
101 "RESTORE
D"
102 AVIEW
103 END
```

40

50

51

60

70

80

90

00

Note: Refer to "HP-41C OWNER'S HANDBOOK AND PROGRAMMING GUIDE" for specific information on keystrokes. The Function Index is found at the very back of the Handbook. Refer to Appendix E in 67 or 97 "OWNER'S HANDBOOK AND PROGRAMMING GUIDE".