TYMSHARE MANUALS

BUSINESS MANAGEMENT LIBRARY

April 1971

TYMSHARE, INC.
525 UNIVERSITY AVENUE, SUITE 220
PALO ALTO, CALIFORNIA 94301

INTRODUCTION

This manual describes 14 Tymshare business applications and personnel management programs. A brief description of each program's capabilities is given, followed by operating instructions and a sample run. The programs are arranged by category. Programs marked with an asterisk are described in greater detail in a separate Tymshare manual.

All of these programs are called by typing the program name in the EXECUTIVE, followed by a Carriage Return. For example,

– RETRIEVE ↵

or

– #MORTGAGE ↵

To allow the user to use the library programs more easily, a standard set of commands has been added to many of the programs. These programs print a colon (:) when they are ready

to accept commands. Each command may be abbreviated to its first three letters and must be followed by a Carriage Return.

The standard commands and their action are:

Command	Action
CAPABILITIES	Prints the program capabilities. Describes methods employed, if appropriate.
EXPERT	Decreases the amount of conversation the program uses. Used by some programs to identify users familiar with the program.
HELP or ?	Prints a complete list of program commands with a brief description of each.
INSTRUCTIONS	Prints a detailed set of operating instructions.
QUIT or Q	Returns the user to the EXECUTIVE.
RUN	Begins program execution, allowing the program to request the needed data.
SAMPLE	Prints a sample run.
VERSION	Prints version number and date of last revision.

Tymshare has many other programs which are useful in business. Please consult the Tymshare STATPAK Manual for information on STATPAK, our new statistics package.

Consult your Tymshare representative for information about other statistical and management related programs on the Tymshare system.

NOTE: In all examples in this manual, information typed by the user is underlined. The symbols used to indicate Carriage Return, Line Feed, and ALT MODE/ESCAPE typed by the user are:

Carriage Return ↵
Line Feed ↓
ALT MODE/ESCAPE ⊕

SECTION 1

GENERAL FINANCE

#ANNUITY
CALCULATION OF ANNUITIES

Description

This program performs calculations for both payment and withdrawal annuities. Payment annuities include any case in which the principal amount is increased by equal payments, such as Christmas club accounts that pay interest. Withdrawal annuities include any case in which the principal is decreased by regular payments, such as loans, mortgages, and planned income programs.

Any three relevant unknowns may be entered, and #ANNUITY calculates the fourth.

For a payment annuity, the user enters any three of the following:

1. Number of periods (N).
2. Total amount at end of N periods (A).
3. Interest rate per period in percent (I).
4. Amount of payment for each period (R).

For a withdrawal annuity, the user enters any three of the following:

1. Number of periods (N).
2. Original principal amount (P).
3. Interest rate per period in percent (I).
4. Amount withdrawn each period (R).

For withdrawal annuities, #ANNUITY also lists the total interest paid and a table of withdrawals, principal, and interest.

Instructions

The program is called by typing #ANNUITY ↵ in the EXECUTIVE. First, the program asks if instructions are needed. This question may be answered YES or NO followed by a Carriage Return. Then, #ANNUITY requests the annuity type, payment or withdrawal, and the unknown variable. The options are described above. When calculating interest, #ANNUITY requests the number of periods in a year in order to calculate the annual interest rate.

After each calculation, #ANNUITY asks ANOTHER CASE?. The following alternatives are available.

Code	Option
1	Another case, same unknown variable.
2	Another case, different unknown variable.
3	Another case, another annuity type.
4	Total interest paid over N periods.
5	Table of withdrawals, principal, and interest.
6	Stop the program.

The additional output requested by code 4 or 5 is available only for withdrawal calculations.

Example

In a Christmas club account, a depositor pays $50 every month for ten months. The bank pays 4.5% annual interest, compounded monthly. The example below calculates the amount the depositor receives at the end of ten months.

```
-#ANNUITY

A N N U I T Y   C A L C U L A T I O N
DO YOU WISH INSTRUCTIONS (YES OR NO)? NO
WHICH ANNUITY TYPE (1=PAYMENT, 2=WITHDRAWAL)     ? 1
WHICH VARIABLE IS UNKNOWN (N,A,I,R)? A
WHAT ARE N(INTEGER),I(PCT),R($)? 10,.375,50
AMOUNT AT END OF N PERIODS = A = 508.52243
ANOTHER CASE (TYPE CODE NUMBER)? 3
WHICH VARIABLE IS UNKNOWN (N,P,I,R)? R
WHAT ARE N(INTEGER),P($),I(PCT)  ? 10,500,1.5
WITHDRAWAL EACH PERIOD = R = 54.217089
ANOTHER CASE (TYPE CODE NUMBER)? 5
```

The percent per period is entered. The annual interest of 4.5% divided by 12 periods becomes .375%.

PERIOD	PRINCIPAL	INTEREST	PRINC BAL	INT TO DATE
0			500	
1	46.72	7.4970889	453.28	7.4970889
2	47.42	6.7970889	405.86	14.294178
3	48.13	6.0870889	357.73	20.381267
4	48.85	5.3670889	308.88	25.748356
5	49.58	4.6370889	259.3	30.385445
6	50.33	3.8870889	208.97	34.272534
7	51.08	3.1370889	157.89	37.409623
8	51.85	2.3670889	106.04	39.776712
9	52.63	1.5870889	53.41	41.363801
10.00001	53.41	.8	0	42.163801

ANOTHER CASE (TYPE CODE NUMBER)? 6↵

-

#CASHFLOW
DISCOUNTED CASH FLOW

Description

This program calculates the rate of return on an investment, given a series of cash flows. The user specifies the number of payment periods per year and the total cash flow for each period. #CASHFLOW uses the Newton-Raphson method of polynomial equation solution to calculate the simple annual interest returned on the investment.

Instructions

The program is called from the EXECUTIVE by typing #CASHFLOW↵. #CASHFLOW first asks for the total number of periods and the number of periods per year. The net cash flows may be typed individually or in sets. The program requests that the user indicate the desired method of input. The user must type a comma or a Carriage Return between input values and follow the last value by a Carriage Return. #CASHFLOW begins computations after the last value is typed. Net cash outflows are entered as negative numbers.

When the cash inflows are discounted to within .000001% of the initial investment, the program prints the correct rate of return which is accurate to all decimal places shown.

If the rate of return is highly unreasonable, some of the cash flows may have been entered incorrectly. For example, omitting the negative sign on the initial cash outflow causes a very large rate of return.

Example

The following program is run twice for the same data. The data is entered first in sets and then individually. The cash flows are entered for 24 periods, 12 periods in a year.

```
-#CASHFLOW↵

                *** DISCOUNTED CASH FLOW PROGRAM ***

DO YOU WISH INSTRUCTIONS? (TYPE YES OR NO) ? NO↵

GIVE TOTAL NO. OF PERIODS, AND PERIODS PER YEAR: ? 24,12↵

TYPE '0' FOR INDIVIDUAL PERIOD FLOWS OR
     '1' FOR SETS OF CASH FLOWS: ? 1↵

GIVE A SERIES OF 'P,B' WHERE P = NUMBER OF FLOWS IN
THE SERIES AND B = THE CASH FLOW FOR THOSE PERIODS:
? 1,-1000↵
? 12,60↵
? 12,50↵
```

```
RETURN =      29.63823 %

FINISHED? ? NO↵

GIVE TOTAL NO. OF PERIODS, AND PERIODS PER YEAR: ? 24,12↵

TYPE '0' FOR INDIVIDUAL PERIOD FLOWS OR
     '1' FOR SETS OF CASH FLOWS: ? 0↵

TYPE IN YOUR CASH FLOWS
? -1000↵
? 60,60,60,60,60,60,60,60,60,60,60,60↵
? 50,50,50,50,50,50,50,50,50,50,50,50↵


RETURN =      29.63823 %

FINISHED? ? YES↵

-
```

#COSTSAVE

Description

#COSTSAVE calculates the present value of the cost savings realized by making a product as opposed to buying it. When calculating the cost of buying a product, #COSTSAVE considers the true cost of the capital and the effect of corporate taxes. Included in the cost of manufacturing are the true cost of the investment, fixed costs, depreciation, and state and federal taxes. #COSTSAVE prints the expenses and net cash flows per year and the total cash flows.

Instructions

This program is conversational and has the standard set of commands. Each command or item of data entered by the user should be followed by a Carriage Return. The program is called from the EXECUTIVE by typing #COSTSAVE↵. When the colon appears, the user types RUN followed by a Carriage Return.

The program asks for the name of the company and the name of the product. It then asks 11 questions concerning the cost of making and buying the product. The program calculates the present value of the cost to make and the cost to buy, and prints a table of cash flows.

The user may then change the answers to any of the questions and recompute the cost savings. If any variables are modified, the user may request a new table of cash flows and a new calculation of cost savings. If the user does not wish to make further modifications, he may type QUIT↵ after the colon and return to the EXECUTIVE.

Example

```
-#COSTSAVE↵

:RUN↵
PLEASE ANSWER THE FOLLOWING QUESTIONS . . .

ENTER THE NAME OF YOUR COMPANY  : AMERICANIC CORP.↵

ENTER THE NAME OF THE PRODUCT YOU ARE CONSIDERING
MAKING OR BUYING  : DISK↵

FOR THE FOLLOWING ITEMS PLEASE ENTER ALL COSTS IN DOLLARS :

  1. ENTER THE COST TO BUY A DISK FOR YOUR PLANT  : 14000↵

  2. ENTER THE COST TO MANUFACTURE A DISK IN YOUR PLANT
(INCLUDE DIRECT MATERIALS AND LABOR BUT NOT OVERHEAD)  : 4000↵

  3. ENTER THE INITIAL INVESTMENT (COST OF THE EXTRA
MACHINERY THAT WOULD BE NEEDED IF YOU WERE GOING TO
MANUFACTURE DISKS)  : 6123000↵

  4. ENTER THE INVESTMENT LIFE IN YEARS  : 6↵
```

```
  5. ENTER THE SALVAGE VALUE OF THIS INVESTMENT  : 600000↵

  6. ENTER THE ANNUAL FIXED COSTS (E.G. SUPERVISION
AND MAINTENANCE) INVOLVED IN MAKING DISKS  : 90000↵

  7. ENTER YOUR CORPORATE TAX RATE (PERCENT)  : 48↵

  8. ENTER THE LOCAL TAX RATE ON THE EXTRA
INVESTMENT (DOLLARS PER THOUSAND)  : 3.112↵

  9. ENTER YOUR COST OF CAPITAL (PERCENT)  : 8.5↵

 10. ENTER YOUR ESTIMATED YEARLY DEMAND FOR DISKS  : 400↵

    SUM-OF-THE-YEARS-DIGITS METHOD WILL BE USED TO
DEPRECIATE THE INVESTMENT.:

 11. ENTER THE DEPRECIATION LIFE IN YEARS :6↵

THE PRESENT VALUE OF THE COST TO MAKE IS  $7,802,399.16
THE PRESENT VALUE OF THE COST TO  BUY IS $13,260,045.84

AMERICANIC CORP. SHOULD MAKE DISKS AT A SAVINGS OF  $5,457,646.68

                    ------ THE FLOWS ------

        ------- IF BUY ------        ------ IF MAKE ------      -- NET --
YEAR:  EXPENSE:     CASH FLOW:      EXPENSE:     CASH FLOW:     CASH FLOW
 0          .00           .00    6123000.00    6123000.00   -6123000.00
 1   5600000.00    2912000.00    3282144.04     128714.90    2783285.10
 2   5600000.00    2912000.00    3015051.76     252826.92    2659173.08
 3   5600000.00    2912000.00    2748777.94     377364.53    2534635.47
 4   5600000.00    2912000.00    2483322.57     502327.74    2409672.26
 5   5600000.00    2912000.00    2218685.66     627716.54    2284283.46
 6   5600000.00    2912000.00    1354867.20     441530.94    2470469.06
-------------------------------------------------------------------------

WOULD YOU LIKE TO SEE THE EFFECT OF CHANGING A VARIABLE  : YES↵
HOW MANY QUESTIONS DO YOU WISH TO CHANGE? 2↵
ENTER THE QUESTION NUMBERS (1-11)  : 4,11↵

ENTER THE NEW ANSWERS  : 8,7↵
DO YOU WISH TO SEE THE FLOWS  : NO↵

THE PRESENT VALUE OF THE COST TO MAKE IS  $8,840,944.52
THE PRESENT VALUE OF THE COST TO  BUY IS $16,421,300.80

AMERICANIC CORP. SHOULD MAKE DISKS AT A SAVINGS OF  $7,580,356.28

DO YOU WISH TO MAKE MORE CHANGES  : NO↵

:QUIT↵
```

#DEPREC
FOUR-METHOD DEPRECIATION ON INVESTMENT

Description

#DEPREC computes and prints a yearly or monthly table of the amount of depreciation on an investment. Four depreciation methods are calculated.

1. **Straight Line**
   With this method, the depreciation each year is the same and is obtained by dividing the cost less the salvage value by the useful life of the property.
2. **Double Declining Balance**
   A declining balance is obtained by subtracting the amount of depreciation taken each year from the cost of the property before computing the next year's depreciation. The same rate applies to a smaller, or declining, balance each year. The double declining balance method uses twice the straight line rate on the declining balance.
3. **Sum of the Year's Digits**
   The depreciation is a fraction of the cost less the salvage value, calculated in the following manner. The denominator is the sum of all the numbers from 1 to the useful life of the property. The numerator is the number of years remaining in the useful life of the property. If the property has a life of four years, the denominator is 1+2+3+4=10. The fractions each year are 4/10, 3/10, 2/10, 1/10.
4. **125% Declining Balance**
   This method is similar to the double declining balance, but it uses a rate 1.25 times the straight line rate.

When the balance becomes small, it is often advantageous to switch from a declining balance to a straight line method. Then the depreciation for each year remaining in the lifetime of the property is the remaining balance divided by the remaining years. The user can request #DEPREC to perform this switchover automatically as soon as the straight line annual rate is greater than the declining balance rate. Alternatively, the user can specify a switchover year or request no switchover at all. If there is no switchover, then a balance remains at the end of the depreciable life.

Instructions

The program is called from the EXECUTIVE by typing #DEPREC ↵. When the colon appears, the user types RUN ↵ to begin execution. The first question asked is INPUT FROM?. The user enters TEL if he wishes to enter the data from his terminal or the file name if his data is stored on a file. When the data is entered from the terminal, the user can request either detailed prompting or minimal prompting by answering the question DO YOU NEED INSTRUCTIONS TO INPUT THE DATA (Y OR N)? with YES or NO, respectively. The input from the terminal is continued by answering the prompting questions. All answers must be followed by a Carriage Return.

If the input is from a file, the file must contain 15 entries for depreciation.

1. The amount of the investment
2. The salvage value

3. The depreciable life in years

4. & 5. The month and year of the investment

6. The discount rate in decimal form (a 6.5% discount rate is entered as .065)

7. Breakdown code: 1 for annual values only; 0 for monthly breakdown

8.-11. Depreciation codes: 1 means YES; 0 means NO

   8. Straight line

   9. Double declining balance

   10. Sum of year's digits

   11. 125% declining balance

12.-14. Output codes: 1 means YES; 0 means NO

   12. Depreciation for the year

   13. Accumulative depreciation for the year

   14. Undepreciated balance

15. Switchover code: 0 means no switchover; 1 means automatic switchover; any other entry is the year of switchover, e.g., 1978.

An example file might contain:

```
1000,500,7,6,1971,.065,1,0,1,0,1,1,0,0,1
```

Example 1

```
-#DEPREC↵

:RUN↵
PROGRAM EXECUTION.....03/26  17:15


INPUT FROM ? TEL↵
DO YOU NEED INSTRUCTIONS TO INPUT THE DATA (Y OR N)? Y↵

WHAT IS THE AMOUNT OF THE INVESTMENT ? 1000↵

WHAT IS THE SALVAGE VALUE ? 0↵

WHAT IS THE DEPRECIABLE LIFE (IN YEARS) ? 6↵

ENTER MONTH,YEAR IN WHICH THE INVESTMENT
WAS MADE (FOR EXAMPLE:  6,1969) ? 6,1971↵
```

```
WHAT IS THE DISCOUNT RATE (IN DECIMAL NOTATION) FOR COMPUTING
THE PRESENT VALUE OF THE ANNUAL DEPRECIATION ? 0
TYPE A 1 FOR AN ANNUAL SUMMARY ONLY,
AND A 0 FOR A MONTHLY BREAKDOWN ? 1

WHICH OF THE FOLLOWING DEPRECIATION SCHEMES DO YOU WISH
TO COMPUTE (1=YES, 0=NO)
 STRAIGHT LINE ? 0
 DOUBLE DECLINING BALANCE ? 1
 SUM-OF-YEARS-DIGITS ? 0
 125% DECLINING BALANCE ? 1
WHICH OF THE FOLLOWING OUTPUTS DO YOU WANT DISPLAYED
(1=YES;0=NO)
DEPRECIATION FOR THE YEAR ? 1
CUM DEPRECIATION FOR THE YEAR ? 0
UNDEPRECIATED BALANCE FOR YEAR ? 0

YOU HAVE OPTIONS ON SWITCHOVER FROM 200% & 125% DECL. BALANCE TO
STRAIGHT LINE.  TO PREVENT SWITCHOVER TYPE 0; TO SPECIFY THE
YEAR OF SWITCHOVER, TYPE THE YEAR; TO OBTAIN AUTOMATIC SWITCH-
OVER WHEN THE ANNUAL STRAIGHT LINE DEPRECIATION BECOMES
GREATER THAN DOUBLE DECLINING BALANCE, TYPE 1.  WHICH DO YOU
WANT ? 1


DATE                 200/DB        125/DB

DEP FOR 1971           167           104
DEP FOR 1972           278           187
DEP FOR 1973           185           158
DEP FOR 1974           123           158
DEP FOR 1975            99           158
DEP FOR 1976            99           158
DEP FOR 1977            49            79
PRESENT VALUE OF DEPRECIATION,
BEGINNING OF  1971
WITH DISCOUNT RATE OF  0%

                      1000          1000


:QUIT

-
```

Example 2

```
-#DEPREC↵

:RUN↵

PROGRAM EXECUTION.....03/26   17:26


INPUT FROM ? TEL↵
DO YOU NEED INSTRUCTIONS TO INPUT THE DATA (Y OR N)? N↵
INPUT THE 15 PARAMETERS
? 3450,0,3,1,1970,.065,1,1,1,0,0,1,0,0,1↵

DATE                STRLINE        200/DB

DEP FOR 1970          1054           2108
DEP FOR 1971          1150            894
DEP FOR 1972          1150            413
DEP FOR 1973            96             34
PRESENT VALUE OF DEPRECIATION,
BEGINNING OF  1970
WITH DISCOUNT RATE OF  6.5%

                      3304           3292


:QUIT↵

-
```

Example 3

```
-EDITOR↵
*APPEND↵
3456,0,3,1,1970,.05,1,0,1,0,0,1,0,0,1↵
6555,0,4,6,1969,.065,1,1,1,0,0,1,1,0,1↵
*WRITE DEPDATA↵
 NEW FILE↵
77 CHARACTERS
*QUIT↵
```

The user types D^C to terminate the EDITOR APPEND mode.

```
-#DEPREC ↵

:RUN

PROGRAM EXECUTION.....03/26  17:29


INPUT FROM ? DEPDATA ↵

DATE                200/DB

DEP FOR 1970         2112
DEP FOR 1971          896
DEP FOR 1972          414
DEP FOR 1973           34
PRESENT VALUE OF DEPRECIATION,
BEGINNING OF  1970
WITH DISCOUNT RATE OF  5%

                      3333

DATE                STRLINE        200/DB

DEP FOR 1969          819            1639
CUM DEP               819            1639

DEP FOR 1970         1639            2458
CUM DEP              2458            4097

DEP FOR 1971         1639            1229
CUM DEP              4097            5326

DEP FOR 1972         1639             819
CUM DEP              5736            6145

DEP FOR 1973          819             410
CUM DEP              6555            6555

PRESENT VALUE OF DEPRECIATION,
BEGINNING OF  1969
WITH DISCOUNT RATE OF  6.5%

                      6070            6212

:QUIT ↵

-
```

#LESSOR
LEASE ANALYSIS

Description

#LESSOR calculates the lessor's cash flows and rate of return on a lease with variable period-end payments. The lease rate is the compound simple interest on the purchase price minus the prepayments; the payments are assumed to be a combination of principal and interest. #LESSOR also calculates the after tax rate of return. The after tax rate of return is the real return on the amount invested from the lessor's portfolio. The principal is the purchase price minus the borrowed amount. The payments are adjusted to reflect the corporate taxes (on the payment minus the depreciation and loan interest), the loan payments on the borrowed amount, and the tax credit (for the first period).

Instructions

#LESSOR is conversational and has a standard set of commands. To call the program, the user types #LESSOR ↵ in the EXECUTIVE. When the colon appears, the user types RUN and a Carriage Return.

The program requests the data needed for each analysis. The loan and tax rates must be entered as a decimal rather than as a percent. Commas must not be used in the numbers. Each entry must be followed by a Carriage Return.

After one analysis is run and the colon appears, another analysis can be executed by typing RUN ↵.

Example

```
-#LESSOR ↵

:RUN ↵

ENTER THE PURCHASE PRICE : 33000 ↵

ENTER THE LEASE PAYMENT (ENTER ZERO IF GROUPS OF
LEASE PAYMENTS ARE TO BE GIVEN).
4700 ↵

ENTER THE NUMBER OF LEASE PAYMENTS : 8 ↵

ENTER THE METHOD OF DEPRECIATION :  1=STRAIGHT LINE
2=2XSTRAIGHT LINE ; 3=SUM-OF-THE-YEAR'S DIGITS;
4=1.5XSTRAIGHT LINE.  : 1 ↵

ENTER THE DEPRECIABLE LIFE IN YEARS  : 9 ↵

ENTER THE SALVAGE VALUE FOR TAX PURPOSES :3000 ↵

ENTER SALVAGE VALUE ACTUALLY EXPECTED : 4500 ↵
```

```
WHO GETS THE INVESTMENT TAX CREDIT?( 0=LESSEE ; 1=LESSOR) : 0
ENTER THE BORROWED AMOUNT : 26000
ENTER THE INTEREST RATE FOR BORROWING : .087
ENTER THE NUMBER OF PERIODS FOR THE BORROWING  : 8
ENTER THE FIXED PAYMENT TO PAY OFF THE LOAN : 3300
ENTER THE NUMBER OF PERIODS PER YEAR FOR THIS LEASE: 2,4 OR 12. : 4
ENTER YOUR TAX RATE 0=.480 1=.528 OR INPUT A DECIMAL RATE. .496
ENTER THE NUMBER OF PRE-PAYMENTS : 0
```

SUMMARY OF INPUT DATA

INVESTMENT	$33,000
LEASE PAYMENT	$4,700
PAYMENTS	QUARTERLY
# OF LEASE PAYMENTS	8
DEPRECIATION	STRAIGHT LINE
DEPRECIABLE LIFE	9
TAX SALVAGE	$3,000
ACTUAL SALVAGE	$4,500
BORROWED AMT	$26,000
RATE	.0870
PERIODS	8
FIXED PAYMENT	$3,300.00
PERIODS PER YEAR	4
PRE-PAYMENT	$.00
TAX RATE	.496

INVESTMENT TAX CREDIT IS TO BE TAKEN BY THE LESSEE

RESULTS OF CALCULATION

LEASE RATE: 11.9786 % PER YEAR, 4 TIMES PER YEAR.

LESSOR'S AFTER-TAX RETURN: 20.2391 % PER YEAR, COMPOUNDED

DO YOU WANT A LISTING OF FLOWS? YES↵

LISTING OF CASH FLOWS

PERIOD	INT	PRINC.	DEPREC.	TAX PAYMENT	CASH FLOW
0	.00	.00	.00	.00	-7000.00
1	5.66	3294.35	833.33	1915.06	-515.06
2	4.94	3295.06	833.33	1915.42	-515.42
3	4.22	3295.78	833.33	1915.77	-515.77
4	3.50	3296.50	833.33	1916.13	-516.13
5	2.79	3297.21	833.33	1916.48	-516.48
6	2.07	3297.93	833.33	1916.84	-516.84
7	1.35	3298.65	833.33	1917.20	-517.20
8	.64	3299.36	833.33	-8911.78	14811.78

:QUIT↵

-

#MORTGAGE
MORTGAGE ANALYSIS

Description

There are four essential elements to every mortgage:

- Interest rate
- Life of the loan
- Amount borrowed
- Monthly payment

When given any three of these elements, #MORTGAGE computes the fourth element and prints an annual or monthly mortgage table.

Instructions

The program is called by typing #MORTGAGE ↵ in the EXECUTIVE. The program is conversational and has the standard set of commands. When the colon appears, the user types RUN ↵.

The program requests the necessary data. Each item of data should be entered as requested and followed by a Carriage Return. The interest is entered as a decimal rather than percent. For example, 6% is entered as .06.

The program computes the unknown element and prints the terms of the mortgage and a table of interest, principal repayment, and outstanding principal. It then produces a summary of interest and principal payment and outstanding principal.

Example

```
-#MORTGAGE↵

:RUN↵

WHAT DO YOU WANT TO FIND? (RATE,LIFE,AMOUNT,MONTHLY PYMT.) : RATE↵

ENTER THE LIFE OF THE MORTGAGE: YEARS, MONTHS  : 8,0↵

ENTER THE AMOUNT TO BE BORROWED  : 10000↵

ENTER THE AMOUNT OF ONE MONTHLY PAYMENT  : 157.34↵

ENTER THE DATE THE MORTGAGE LOAN IS TO BE MADE:
(EX. MAR,1970) ? SEPT,1971↵

FOR HOW MANY YEARS DO YOU WANT
THE MORTGAGE TABLE PRINTED? : 8↵

WHAT TABLE DO YOU WANT PRINTED? (ANNUAL, MONTHLY) : ANNUAL↵
```

*** MORTGAGE TERMS ***

NOMINAL ANNUAL RATE = 11.047200 %
LIFE OF MORTGAGE = 8 YEARS, 0 MONTHS
AMOUNT BORROWED = $10000.00
MONTHLY PAYMENT = $157.34

*** MORTGAGE TABLE ***

YEAR	INTEREST	PRINCIPAL REPAYMENT	ENDING PRINCIPAL OUTSTANDING
1971	274.37	197.65	9802.35
1972	1040.84	847.24	8955.11
1973	942.35	945.73	8009.38
1974	832.41	1055.67	6953.71
1975	709.71	1178.37	5775.34
1976	572.73	1315.35	4459.99
1977	419.86	1468.22	2991.77
1978	249.18	1638.90	1352.87
1979	63.03	1352.87	.00

AFTER 8 YEARS,
TOTAL INTEREST PAID $5104.48
TOTAL PRINCIPAL REPAID $10000.00
TOTAL PRINCIPAL OUTSTANDING $.00

:QUIT↵

-

#PRESVAL
INVESTMENT ANALYSIS

Description

#PRESVAL successively approximates the rate of return on investments, given a series of cash flows. When the cash flows are discounted to within .000001% of the initial investment, the program prints the correct rate of return.

The program then prints the net present value based on the calculated rate of return and the discounted flows for each period. If the user chooses, the program can calculate the net terminal value of an investment for the user-supplied discount rate. #PRESVAL can analyze as many as 100 cash flows.

Instructions

To call the program, the user types #PRESVAL ↵ in the EXECUTIVE. The program is conversational and has the standard set of commands. When the colon appears, the user types RUN ↵.

Data is entered from the terminal as requested. Cash outflows should be entered as negative values and cash inflows as positive values. Cash flows should be entered in thousands and may be carried to two decimal places. For example, 211,571 should be entered as 211.57. END ↵ is typed by the user when all flows have been entered.

Input Example: -1000,375.5,3000,550,-100,END ↵

Example

```
-#PRESVAL↵

:RUN↵


DO YOU WANT NET PRESENT VALUE BASED ON CALCULATED RATE
OF RETURN OR NET TERMINAL VALUE BASED ON DISCOUNT RATE
THAT YOU INPUT.  (C=CALCULATED RATE,M=YOUR RATE) : C↵

ENTER INITIAL CASH FLOW (OUTFLOW,NEGATIVE  INFLOW,POSITIVE): -1000↵

ENTER SUBSEQUENT CASH FLOWS (AFTER LAST ENTRY TYPE  END )
1000,1000,1000,3000,3000,3000,1000,-23000,12000,END↵

ARE FLOWS MONTHLY,QUARTERLY, SEMI-ANNUAL, OR ANNUAL (M,Q,S,A) : A↵

RETURN =    -4.262 %


DO YOU WISH TO SEE THE DISCOUNTED FLOWS BY PERIOD ? YES↵


PERIOD        CASH  FLOW          DISCOUNTED TO PRESENT VALUE
               (-)    (+)              (-)        (+)
------        -----------         ---------------------------
 0        -1000.00                -1000.00
 1            .00     1000.00         .00     1044.51
 2            .00     1000.00         .00     1090.99
 3            .00     1000.00         .00     1139.55
 4            .00     3000.00         .00     3570.80
 5            .00     3000.00         .00     3729.73
 6            .00     3000.00         .00     3895.72
 7            .00     1000.00         .00     1356.37
 8      -23000.00         .00    -32584.93        .00
 9            .00    12000.00         .00    17757.48
         --------    --------     --------    --------
        -24000.00    25000.00    -33584.93    33585.14


:QUIT↵

-
```

SECTION 2
SCHEDULING

#CPM1 AND #CPM2
CRITICAL PATH METHOD SCHEDULING PACKAGE

Description

The Critical Path Method (CPM) scheduling package is a management tool which enables a project manager to predict when his project activities can be expected to occur. The method flags those activities which can result in delays in overall project completion if the activities require more time than originally allotted. These activities form a sequence which is called the Critical Path. The program also calculates the amount of slippage tolerable for each event and the earliest and latest starting dates possible within the schedule. The CPM method requires that the project manager plan completely the many details required to complete a project. This is usually done by drawing a network showing the interrelationship among the various activities of the project.

The package consists of two programs, #CPM1 and #CPM2, each of which performs a separate phase of the scheduling operation. #CPM1 checks the CPM network for internal consistency and offers the user an opportunity to make corrections or to add new activities to the network. #CPM2 performs the actual computations and prints the results which can be sorted in a number of different ways for the user's convenience. In calculating the schedule, the program can take into account the number of days worked in a week and any holidays which may occur during the project. The schedule can be obtained with total elapsed days, actual calendar dates, or both.

Alternatively, scheduling can be done by using the network analysis program #PERT. #PERT and CPM are compared on page 44.

Instructions

There are five steps which should be followed to generate a CPM schedule.

1. **Network Diagram**

   A network diagram is drawn to show the individual items of work, services, or tasks (called activities in CPM) that are involved in completing the project. The network is usually drawn as a collection of arrows, with each arrow representing one activity. At the ends of each arrow are circles which represent events. An event is the starting point of an activity and occurs only when all activities preceding it have been completed.

   An activity must be identified uniquely by using the event number at its tail (called the I index) and the event number at its head (called the J index). Therefore, each activity must have unique I and J indices. This rule sometimes makes it necessary to use a type of activity that does not represent real work. This activity is referred to as a dummy and is drawn as a dotted line. The numbers used for the events may be random and need not

be in any particular sequence. A network diagram may have only one starting and one ending event, but this does not limit the number of starting or ending activities.

2. **Duration and Cost**

The next step of network development requires that the project manager estimate both the time necessary to complete each activity and the associated cost. The duration and cost estimates are made for each activity individually. Any dummy activities which are in a network will, of course, have both a zero cost and duration. For convenience, the project manager may write the duration and cost information on the network diagram. The usual convention is to put the duration (in days) above the arrow for each activity and the cost below.

3. **Network File**

Data for the critical path network must be written on a symbolic data file. This file can be written in EDITOR, created from a RETRIEVE data base with #STATLINK, or created by another program. The user may choose any name for this file. Data should be written in the format:

```
N
I(1)  J(1)  D(1)  C(1)
I(2)  J(2)  D(2)  C(2)
  .     .     .     .
  .     .     .     .
  .     .     .     .
I(N)  J(N)  D(N)  C(N)
```

where N is the number of activities in the network; and for each activity, I is the number of the starting event, J is the number of the terminal event, D is the duration, and C is the cost.

Note that I, J, and D may be any integer between 0 and 999. The maximum cost for any activity is $99,999.99.

4. **Network Check**

#CPM1 is used to check the file for internal consistency. After the user types #CPM1 ↵ in the EXECUTIVE, the program requests the name of the file on which the data is located. The file name is then entered followed by a Carriage Return.

If any errors are detected by #CPM1, an appropriate message is printed. For example,

```
MULTIPLE STARTS:
        1          1          2
        2          3          4
MULTIPLE ENDS:
        1          1          2
        5          4          6
```

Following each error condition is a list of those activities which are involved in the error condition. The first number is an index which denotes the number of the erroneous entry in the file. The second and third numbers on the line are the event numbers of the activity.

After the error indications are printed, the user can make any corrections or additions that are needed. He then enters the name of a file on which the output data is to be written.

Many legitimate networks have multiple ends. CPM1 considers this a bad network and throws it out. To satisfy CPM1, multiple ends can be reduced to a single end by using activities of 0 duration and 0 cost.

5. **CPM Schedule**

The CPM schedule is produced by typing #CPM2 ↵ in the EXECUTIVE. #CPM2 requests the name of the output file used in CPM1 and the starting date for the project. The program then requests information about holidays. The user may specify holidays from the terminal or from a previously created file. The holiday file should have the following format. The first line is an integer representing the number of holidays in the file. Each successive line contains the holiday date, one per line, which is specified as follows:

MM/DD/YY

All the numbers must contain two digits. Each line is terminated with a Carriage Return.

After the initial parameters are entered, the program computes the schedule and prepares to print the results. Before the output is printed, the program sorts the information so that one of the columns will appear in ascending order. The columns are:

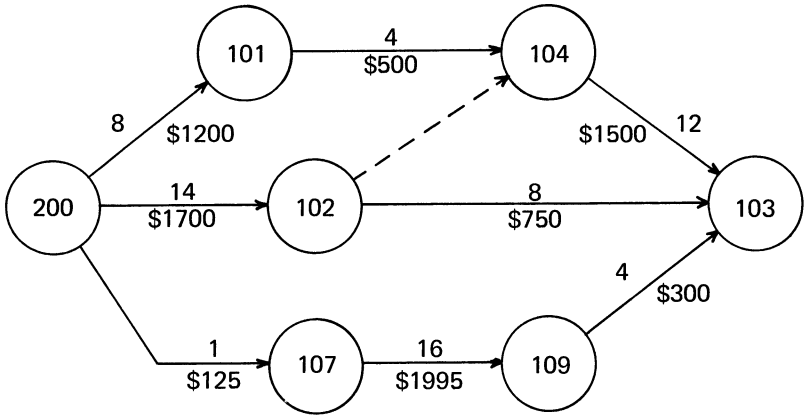
Label	Significance
I	Starting event
J	Ending event
DUR	Duration
ES	Earliest time project can be started if there are no delays
EF	Finish time corresponding to earliest start time
LS	Latest time project can be started without delaying the entry date
LF	Finish time corresponding to latest start time
TF	Total float (LF-EF)
LEV	Level (position in network or number of events from the beginning)
COST	Cost

The user specifies the column to be sorted by typing one of the labels above. The user is asked also to specify how many activities he wishes to have printed. He may type the total number of activities or a smaller number if he wants to see fewer lines. He may choose to have the output written on a file, printed on his terminal, or both at the same time. The output can be produced with actual calendar dates (D), in terms of numeric elapsed time from the project start (N), or both (B).

In the printout, all activities which lie on the Critical Path of the network are flagged by an asterisk at the left-hand margin of the page.

Example

Find the critical path through the network below where (I) denotes event I and D/A denotes an activity of duration D and cost A.



The data file is created in EDITOR, then #CPM1 and #CPM2 are run.

```
-EDITOR
*APPEND
9
200,101,8,1200
101,104,4,500
104,103,12,1500
200,102,14,1700
102,103,8,750
102,104,0,0
200,107,1,125
107,109,16,1995
109,103,4,300
*WRITE NET
 NEW FILE
133 CHARACTERS
*QUIT
```

The user types D^C to terminate the EDITOR APPEND mode.

```
-#CPM1↵

VERSION 1.0

PLEASE TYPE THE NAME OF THE DATA FILE:NET↵

CPM
BEGINNING PHASE 1
HOW MANY ADDITIONS DO YOU WISH TO ENTER: 0↵
DO YOU WANT TO ENTER CORRECTIONS TO NETWORK (YES OR NO) NO↵


NO ERRORS.

NUMBER OF ACTIVITIES:     9
STARTING EVENT    200
ENDING EVENT    103



ENTER NAME OF OUTPUT DATA FILE, PLEASE:JOB↵
 NEW FILE
END PHASE 1

-#CPM2↵

PLEASE TYPE THE NAME OF THE OUTPUT FILE USED IN PHASE 1:JOB↵

ENTER THE STARTING DATE AS FOLLOWS MM/DD/YY
07/01/71↵

ENTER THE NUMBER OF DAYS WORKED PER WEEK:5↵

ARE THE HOLIDAYS ON A FILE (YES OR NO) NO↵

ENTER THE NUMBER OF HOLIDAYS IN THE YEAR:1↵

ENTER THE HOLIDAYS LIKE THE STARTING DATE WAS ENTERED
07/04/71↵

CPM
BEGINNING PHASE 2
NUMBER OF ACTIVITIES:          9
STARTING EVENT:      200       ENDING EVENT:      103

WHAT COLUMN FOR SORT J↵

HOW MANY ACTIVITIES 9↵
```

```
DO YOU WISH THE OUTPUT TO BE ON THE TELETYPE(T),
 A FILE(F), OR BOTH(B)
T↵

DO YOU WANT THE OUTPUT DATE(D),NUMERIC(N), OR BOTH(B)
B↵

PROJECT DURATION:      26        1JUL71   TO    9AUG71
```

C	I	J	DUR	EARLIEST START	EARLIEST FINISH	LATEST START	LATEST FINISH	TF	LEV	COST
	200	101	8	1JUL71 0	14JUL71 8	6JUL71 2	16JUL71 10	2	1	1200.00
*	200	102	14	1JUL71 0	22JUL71 14	1JUL71 0	22JUL71 14	0	1	1700.00
*	104	103	12	22JUL71 14	9AUG71 26	22JUL71 14	9AUG71 26	0	3	1500.00
	102	103	8	22JUL71 14	3AUG71 22	28JUL71 18	9AUG71 26	4	2	750.00
	109	103	4	27JUL71 17	2AUG71 21	3AUG71 22	9AUG71 26	5	3	300.00
	101	104	4	14JUL71 8	20JUL71 12	16JUL71 10	22JUL71 14	2	2	500.00
*	102	104	0	22JUL71 14	22JUL71 14	22JUL71 14	22JUL71 14	0	2	.00
	200	107	1	1JUL71 0	2JUL71 1	9JUL71 5	12JUL71 6	5	1	125.00
	107	109	16	2JUL71 1	27JUL71 17	12JUL71 6	3AUG71 22	5	2	1995.00
									TOTAL: $	8070.00

```
DO YOU WISH TO CONTINUE (YES OR NO)NO↵

-
```

#FORECAST

Description

#FORECAST is a program to aid businessmen responsible for planning and decision making. #FORECAST analyzes the past behavior of an article, determines the trends, and predicts future values based on these trends. Some business applications of #FORECAST are:

- Sales forecasting
- Inventory requirements
- Future labor needs
- Cash requirements
- Income and balance sheet forecasting

#FORECAST models the behavior of the data to project future activity. The model used in #FORECAST follows a periodic series of 12; that is, each cycle consists of 12 values evenly spaced in time. The usual application is to monthly data which has a period of one year. The model has 16 adjustable variables contained in both trigonometric and polynomial terms which allow approximation of both the cyclical and the linear trends in the process that the data represents.[1]

Upper and lower limits corresponding to a user-specified probability are found for each forecast value and for yearly totals. These limits are based on the noise in the data (mean absolute deviation, MAD, from smoothed estimate), the lead time, and the smoothing constant (alpha), which is specified by the user.

Instructions

#FORECAST reads the data to be analyzed from a data file. The data file can be data created in EDITOR or can be the output of another program. A data file can be made from an item in a RETRIEVE data base by using the program #STATLINK.[2]

The first line of the data file is a title to be printed at the beginning of the output. Subsequent lines contain the data values in chronological order, 12 values per period. The values may be separated by commas, spaces, or Carriage Returns. The file must contain at least 24 data points.

The program is called from the EXECUTIVE by typing #FORECAST ↵. When the colon appears the user types RUN followed by a Carriage Return.

The programs ask for the name of the data file and eight format codes. The format codes specify the values of alpha and the probabilities at the calculated upper and lower limits, as well as the format of the output. The significance of each code is as follows.

Code	Value	Function
1	0	Program selects smoothing constant (alpha). Constant is a multiple of .1.
	.05 to .95	Value of alpha specified by user. Value must be a multiple of .05.

1 - The #FORECAST model is described in detail in *Smoothing, Forecasting and Prediction of Discrete Time Series*, R. Brown, Prentice-Hall.

2 - The relationship between #STATLINK and RETRIEVE is explained on page 54.

Code	Value	Function
2	0	Output does not include MAD versus alpha.
	1	Printout of MAD versus alpha.
3	0	Output includes both data and forecast.
	1	Printout of forecast only.
4	0	Limits calculated for 80% probability; codes 5 and 6 ignored.
	1	Probabilities specified by user in codes 5 and 6.
5	integer from 1 to 99	Percent probability at upper limit.
6	integer from 1 to 99	Percent probability at lower limit.
7	0	Forecast for two cycles calculated.
	integer from 1 to 5	Number of cycles to be forecasted.
8	0	Periods identified by number.
	1	Cycles and periods identified by month and year. Starting date requested.

If code 8 is 1, #FORECAST requests a starting month and year. These should be entered as 3/1970 for March 1970.

After #FORECAST has printed the requested information, a colon is displayed, and the program can be terminated by typing QUIT↵.

Example

```
-EDITOR↵
*APPEND↵
FORECAST OF PRICES↵
23,26,29,30,34,32,38,32,39,43,41,39↵
34,36,39,32,40,42,47,49,53,57,54,60↵
34,46,46↵
*WRITE PRICES↵          The user types D^C to terminate the EDITOR APPEND mode.
 NEW FILE↵
100 CHARACTERS
*QUIT↵
```

```
-#FORECAST
NOTE: NEW VERSION, THE FIRST LINE OF THE DATA FILE IS A TITLE.

:RUN

INPUT FILE? PRICES
TYPE FORMAT CODES
? .6,0,0,0,0,0,1,1

STARTING MONTH/YEAR = 3/1969


FORECAST OF PRICES



SMOOTHING CONSTANT=.60
        DATA      ESTM         ERR      MAD
<><><><><><><><><><><><><><><><><><><><><><><><><><><><>

YEAR =  1970

MAR        34        31          3        5
APR        36        34          2        5
MAY        39        37          2        5
JUN        32        39   -      7        5
JUL        40        43   -      3        5
AUG        42        39          3        5
SEP        47        45          2        5
OCT        49        40          9        5
NOV        53        47          6        5
DEC        57        52          5        5
JAN        54        51          3        5
FEB        60        49         11        5

ACTUAL FOR YEAR= 543
ESTIMATE FOR YEAR= 508


YEAR =  1971

MAR        34        41   -      7        5
APR        46        45          1        5
MAY        46        46                   5
```

The printout begins with the second year described in the data.

```
  **FORECAST**                       80% PROB       80% PROB
                                     LESS THAN      AT LEAST
<><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>

YEAR=  1971
JUN             46                        52             39
JUL             43                        49             36
AUG             46                        53             40
SEP             52                        58             45
OCT             49                        56             43
NOV             60                        66             53
DEC             63                        70             57
JAN             63                        69             56
FEB             57                        64             51
ACTUAL TO DATE 126
FORECAST FOR YEAR 605
WITH 80% PROB THAT ACTUAL WILL BE LESS THAN 624
AND 80% PROB THAT ACTUAL WILL BE AT LEAST 586


YEAR=  1972
MAR             53                        59             46
APR             48                        54             41
MAY             52                        59             46
JUN             53                        60             46
JUL             50                        57             43
AUG             54                        60             47
SEP             60                        66             53
OCT             58                        65             51
NOV             69                        76             62
DEC             73                        79             66
JAN             72                        79             66
FEB             67                        73             60
FORECAST FOR YEAR   707
WITH 80% PROB THAT ACTUAL WILL BE  LESS THAN  730
AND 80% PROB THAT ACTUAL WILL BE AT LEAST 684


:QUIT

-
```

LINPROG
LINEAR PROGRAMMING

Description

LINPROG[1] uses the SIMPLEX method to maximize a linear objective function (called the COST function):

$$Z = C_1X_1 + C_2X_2 + \cdots + C_qX_q$$

subject to a given set of linear equality/inequality constraints of the form:

$$a_{1,1}X_1 + a_{1,2}X_2 + \cdots + a_{1,q}X_q \underline{op}\ b_1$$
$$a_{2,1}X_1 + a_{2,2}X_2 + \cdots + a_{2,q}X_q \underline{op}\ b_2$$
$$\vdots$$
$$a_{p,1}X_1 + a_{p,2}X_2 + \cdots + a_{p,q}X_q \underline{op}\ b_p$$

where op may be either $\leqslant$, =, or $\geqslant$.

LINPROG can accommodate up to 30 constraints. The number of cost coefficients allowed is related to the number of constraints by the formula:

$$Q + P + G \leqslant 100$$

where

Q = number of coefficients per constraint.

P = number of constraints.

G = number of "greater than" constraints.

Instructions

LINPROG is called from the EXECUTIVE by typing LINPROG ↵. The system responds with a colon to indicate that LINPROG commands may be entered.

In addition to the standard commands listed on page 2, LINPROG accepts the following commands.

Command	Description
ADD	Allows user to add constraints to his problem.
CHANGE	Allows user to modify a constraint or a vector.
DETAILED SOLUTION	Prints solution containing basis vectors.
ENTER	Allows user to enter constraints.
LIST	Prints current program constraints.
READ	Reads constraints from a data file.
SOLUTION	Prints solution.
WRITE	Writes current problem specifications on a data file.

1 - For further information, see the *Tymshare LINPROG Manual*.

Each of these commands may be shortened to as few letters as are required to identify it uniquely.

In a given line, data items may be separated by commas, spaces, or the relational operators <, =, or >. Numerical data may be expressed as integers, decimal numbers, or in scientific notation. For example,

```
#1 3, 1<6.5 ↵
#2 4 2 = 8 ↵
#3 8.3E-1, 2.5>4 ↵
```

LINPROG accepts constraints as they would normally be written; that is, it expects the coefficients of each variable followed by the right hand side coefficient (with operator). The constraints may be typed in any order regardless of the operator (<,=,>). LINPROG allows negative values on the right hand sides. For example,

```
:ENTER ↵
#1 1.0, 2 1E3 <9.735E2 ↵
#2 3-7 0 = -4.1 ↵
#3 0, 1.32, 9.5 >15 ↵
```

Following the matrix input, the objective function may be typed, preceded by the word COST or an abbreviation of COST, C, CO, or COS.

```
#7 0 2 1 7.23 >95.72 ↵
#8 COST 3.17, 9.59, 8.73, 1E-1 ↵
```

After the constraints are entered, the user may select the output desired. If he types SOLUTION, the program prints the value of all vectors and slacks in the optimal basis as well as the optimal value (maximum) of the objective function.

The command DETAILED SOLUTION prints a line containing the value of the objective function at each iteration followed by the vectors in the basis at that particular point. It then performs the same operations as the SOLUTION command.

Surplus and slack variables which appear in the output are indicated with the letter S. If the variable is one of the vectors in the input matrix, the letter X is printed. Either of these letters is followed by the column number associated with it.

If the objective function is to be minimized rather than maximized, the data file may be modified before execution by reversing the signs of the coefficients in the objective function.

Example

Problem[1]: To maximize the objective function

$30\ X_1 + 45\ X_2$

1 - Taken from *Advanced Methods and Models*, R. I. Beggs, R. E. Herlehy, and C. H. Springer, Vol. 2, Ch. 8, Dow-Jones-Irwin, Inc., Homewood, Illinois, 1965.

while satisfying the following seven restrictions:

$$
\begin{array}{rrrr}
 & X_1 & & \leqslant 6000 \\
 & & X_2 & \leqslant 4000 \\
2.5 & X_1 + 2.0 & X_2 & \leqslant 24000 \\
 & X_1 & & \geqslant 1000 \\
 & & X_2 & \geqslant 1000 \\
3.0 & X_1 - & X_2 & \geqslant 0 \\
2.5 & X_1 + 2.0 & X_2 & \geqslant 10000
\end{array}
$$

```
-LINPROG↵

LINEAR PROGRAMMING PROGRAM

:ENTER↵
# 1 1 0 < 6000↵
# 2 1 0 > 1000↵
# 3 0 1 < 4000↵
# 4 0 1 > 1000↵
# 5 2.5 2 < 24000↵
# 6 3 -1 > 0↵
# 7 2.5 2 > 10000↵
# 8 COST 30 40↵

YOUR MATRIX WILL BE DESTROYED IN COURSE OF SOLUTION

WRITE TO: MATSAVE↵
 NEW FILE↵

:DETAILED SOLUTION↵

      VALUE          VECTORS IN THE PRESENT BASIS

   -1.20000E 17    S1   S2   S3   S4   S5   S6   S7
   -1.20000E 17    S1   S2   S3   S4   S5   X1   S7
   -7.83333E 16    S1   S2   S3   X2   S5   X1   S7
   -1.50000E 16    S1   X4   S3   X2   S5   X1   S7
    1.76471E  5    S1   X4   S3   X2   S5   X1   X3
    2.00000E  5    S1   X4   X6   X2   S5   X1   X3
    3.40000E  5    X5   X4   X6   X2   S5   X1   X3

      BASE          VALUE
       X5         14000.000
       X4          3000.000
       X6         13000.000
       X2          4000.000
       S5          1000.000
       X1          6000.000
       X3          5000.000

 OPTIMUM VALUE =     3.400000E  5

:QUIT↵
```

#MANPOWER
MANPOWER REQUIREMENTS

Description

#MANPOWER allows manufacturing management to determine future manpower requirements by contract, department, and function. With the aid of the Tymshare system, management can reevaluate manpower needs in minutes. This speed and flexibility offer the user two important advantages:

- Manpower loads can be adjusted smoothly, resulting in time and cost savings.
- New schedules reflecting changes in customer and program requirements can be immediately developed and initiated.

#MANPOWER handles up to 30 different forecasted units simultaneously, each of which may have a time span of as many as 105 weeks. The projected manpower requirements for each unit can be approximated by a parabolic or linear curve.

A parabolic curve describes work that increases each week to a peak and then gradually trails off. The user can specify how peaked or how level the distribution should be. Furthermore, #MANPOWER automatically adjusts the distribution if the program detects an ahead-of- or behind-schedule condition.

A linear curve describes work which is evenly distributed with the same number of hours each week.

#MANPOWER prints out two tables: the weekly current and accumulative hours for each unit and the monthly requirements, both by unit and total, with the equivalent personnel.

To calculate a meaningful equivalent personnel, #MANPOWER needs to know the company's accounting calendar. The user supplies this information on a file as described below. This feature allows the program to consider union holidays, government holidays, and the variations in the company's accounting calendar.

Instructions

Before he can use #MANPOWER, the user must create a calendar file describing the company's accounting calendar. The file named /CALEN/ must describe 165 weeks. The first week in the file must be the current calendar week. Each week is described by a single line; the week number is right justified in the first three places. Column 5 contains a 1 if the week ends the month, otherwise a 0. Columns 7, 8, and 9 contain the number of accounting hours in the month if column 5 contains a 1. For example,

```
Column: 1 2 3 4 5 6 7 8 9
          9 8   0
          9 9   0
        1 0 0   0
        1 0 1   1   1 3 6
        1 0 2   0
        1 0 3   0
        1 0 4   0
        1 0 5   0
        1 0 6   1   1 5 1
```

This file can be created in EDITOR with the APPEND command. For example,

```
-EDITOR↵
*APPEND↵
 50 0↵
 51 0↵
 52 0↵
 53 1 136↵
 54,0↵
•
•
•
 98 0↵
 99 0↵
100 0↵
101 1 160↵
•
•
•
*WRITE /CALEN/↵
 NEW FILE↵
1040 CHARACTERS
*QUIT↵

-
```

The user types D^C to terminate the EDITOR APPEND mode.

If the user does not need the whole 165 weeks of accounting data, he can enter zeros for the weeks he does not want #MANPOWER to consider.

Information about the project work distribution can be entered into #MANPOWER directly from the terminal or from a file. When information is entered directly, the program prompts the user by requesting the data required next. All numeric data, whether from the terminal or a file, must be terminated by a comma.

To enter data directly from the terminal, the user simply calls #MANPOWER from the EXECUTIVE by typing #MANPOWER ↵. When the colon appears, he types RUN ↵. When #MANPOWER asks if input is from a file, the user answers N ↵. Next the program asks the number of the current calendar week. The week number must be terminated by a comma and must correspond to the first week in the calendar file. Then the program requests a description of each unit as follows:

UNIT:	A name for the unit. The name can have up to five characters. The letter E is entered to terminate input.
SPAN:	The number of weeks the unit is projected to be in process.
OGIVE:	The distribution function. Zero indicates linear distribution. The larger the OGIVE (from 1 to 100), the more markedly peaked the distribution. An OGIVE of 60 for 1000 hours distributed over 10 weeks would indicate weekly expenditures of:

28.32	80.99	127.83	169.07	193.78
155.94	113.33	75.74	42.13	13.12

CAL. WEEK FROM:	The week number the unit will begin or the current week number if unit is already in process. An in-work date before the current week is designated by specifying the correct span. A span of 10 with the unit from week number 49 to week number 55 implies in-work in week number 46.
TO:	The week number the unit should be complete (SPAN + true in-work date).
EAC:	Estimated total hours required for unit.
ACTUALS:	Number of hours completed on unit to date.
IS THERE STORAGE?	Answered with Y or N. N indicates that the project remains in-work continuously. Y indicates that the work is interrupted and stored for periods of time. If there is storage, the program requests the number of the week in which the work is interrupted and the number of weeks the unit is shelved. Storage extends the completion date.
IS THIS OK	Answered with Y or N. N indicates that #MANPOWER should ignore the unit just entered. This allows the user to correct mistakes by reentering the unit description.

When all the units are described, the user enters the letter E for the unit name. #MANPOWER then prints the current status of each unit (the number of hours ahead of or behind schedule). The program asks if the user wants the weekly breakdown of each unit. If the user answers N, only the monthly data is printed.

If the input is from a file, the unit data must be in the same order as data entered from the terminal. The current calendar week is entered from the terminal. The file contains the information for each unit with the name alone on the first line, the next six numbers each followed by a comma on the next line, and Y or N on the third line. If the third line contains Y, the unit has a fourth line containing the two numbers describing storage. The file does not contain an answer to the IS THIS OK question. Refer to the second example below.

When the problem is finished, #MANPOWER asks DO YOU WANT TO RUN THIS PROGRAM AGAIN? If the user answers NO, he is returned to FORTRAN and a plus sign (+) is displayed. He types ;Q ↵ to return to the EXECUTIVE.

Example

The following example uses data entered from the terminal.

```
-#MANPOWER↵

:RUN↵
```

```
                    **** M A N P O W E R ****
                    MANPOWER FORECASTING PROGRAM


FILE INPUT(Y OR N): N↵
ENTER CUR. CAL. WK: 50,↵
-----------
UNIT :  AC143↵
SPAN        :  10,↵
OGIVE       :  60,↵
CAL. WEEK
      FROM:  50,↵
         TO:  56,↵
EAC         :  1000,↵
ACTUALS     :  250,↵
IS THERE STORAGE? N↵
IS THIS OK Y↵
-----------
UNIT :  BT059↵
SPAN        :  10,↵
OGIVE       :  60,↵
CAL. WEEK
      FROM:  52,↵
         TO:  61,↵
EAC         :  1000,↵
ACTUALS     :  0,↵
IS THERE STORAGE? N↵
IS THIS OK Y↵
-----------
UNIT :  CY103↵
SPAN        :  10,↵
OGIVE       :  60,↵
CAL. WEEK
      FROM:  50,↵
         TO:  58,↵
EAC         :  1000,↵
ACTUALS     :  20,↵
IS THERE STORAGE? Y↵
BEG. CAL WK:  50,↵
HOW MANY WKS:  6,↵
IS THIS OK Y↵
-----------
UNIT :  E↵

----------
UNIT  AC143 AHEAD OF SCHEDULE    12.86 HOURS

----------
UNIT  BT059 ON SCHEDULE

----------
UNIT  CY103 BEHIND SCHEDULE     8.32 HOURS
```

```
WANT CUR. & CUM PRINTOUT? YES↵

UNIT AC143  OGIVE: 62.500

WK       CUR.HRS.    CUM.HRS.

 50       181.16       431.16
 51       193.84       625.00
 52       149.62       774.62
 53       106.57       881.19
 54        69.43       950.62
 55        37.86       988.48
 56        11.52      1000.00

UNIT BT059  OGIVE: 60.000

WK       CUR.HRS.    CUM.HRS.

 52        28.32        28.32
 53        80.99       109.31
 54       127.83       237.14
 55       169.07       406.22
 56       193.78       600.00
 57       155.94       755.94
 58       113.33       869.28
 59        75.47       944.74
 60        42.13       986.88
 61        13.12      1000.00

UNIT CY103  OGIVE: 48.750

WK       CUR.HRS.    CUM.HRS.

 50       S T O R A G E
 51       S T O R A G E
 52       S T O R A G E
 53       S T O R A G E
 54       S T O R A G E
 55       S T O R A G E
 56        56.81        76.81
 57        96.90       173.72
 58       136.70       310.41
 59       177.09       487.50
 60       182.77       670.27
 61       143.34       813.60
 62       103.14       916.75
 63        62.34       979.08
 64        20.92      1000.00
```

*****M O N T H L Y L O A D R E C A P*****

	MONTH ENDING CALENDAR WEEK								
	53	58	62	66	70	75	79	83	88
UNIT									
AC143	632	119	0	0	0	0	0	0	0
BT059	109	760	130	0	0	0	0	0	0
CY103	0	291	606	83	0	0	0	0	0
TOTAL	741	1170	736	83	0	0	0	0	0
EP	5	7	5	0	0	0	0	0	0

DO YOU WANT TO RUN THIS PROGRAM AGAIN? NO↵

PROGRAM FINISHED
STOP

+;Q↵

-

Example

The file below is created in EDITOR and entered into #MANPOWER in the following example.

```
-EDITOR↵
*APPEND↵
AC143↵
10,60,50,56,1000,250,↵
N↵
BT059↵
10,60,52,61,1000,0,↵
N↵
CY103↵
10,60,50,58,1000,20,↵
Y↵
50,6,↵
E↵
*WRITE MANSCHED↵        The user types D^C to terminate the EDITOR APPEND mode.
 NEW FILE↵
95 CHARACTERS
*QUIT↵

-#MANPOWER↵


:RUN↵
```

**** M A N P O W E R ****
MANPOWER FORECASTING PROGRAM

FILE INPUT(Y OR N): Y
ENTER FILE NAME : MANSCHED
ENTER CUR. CAL. WK: 50.

UNIT AC143 AHEAD OF SCHEDULE 12.86 HOURS

UNIT BT059 ON SCHEDULE

UNIT CY103 BEHIND SCHEDULE 8.32 HOURS

WANT CUR. & CUM PRINTOUT? NO

*****M O N T H L Y L O A D R E C A P*****

	MONTH ENDING CALENDAR WEEK								
	53	58	62	66	70	75	79	83	88
UNIT									
AC143	632	119	0	0	0	0	0	0	0
BT059	109	760	130	0	0	0	0	0	0
CY103	0	291	606	83	0	0	0	0	0
TOTAL	741	1170	736	83	0	0	0	0	0
EP	5	7	5	0	0	0	0	0	0

DO YOU WANT TO RUN THIS PROGRAM AGAIN? NO

PROGRAM FINISHED
STOP

+;Q

-

#PERT
PERT NETWORK ANALYSIS

Description

#PERT analyzes a Program Evaluation and Review Techniques (PERT) network. A PERT network describes the procedures required to complete a project. Each activity in the project has a beginning event and an ending event. The user describes the project to #PERT in terms of these activities and events. The user supplies the structure of the network and, for each activity, the most optimistic, most pessimistic, and most likely times for completion. From these values, the program computes the expected completion time for each activity and the associated variance. These quantities are computed as:

T(A) = (A+B+4*C)/6 = expected completion time
V(A) = ((A-B)/6)↑2 = variance

where A, B, and C are the most optimistic, most pessimistic, and most likely times for an activity.

If there is more than one path to an event, the path taking the longest total expected time is called the critical path. This path is critical because any delay in any activity along the path delays the final event.

#PERT provides tabular output of:

- The variance for each event. This is the sum of the variances for each activity along the critical path to that event. The variance for each activity is described above. This variance is an absolute value rather than a fractional variance and applies to both the earliest and latest start times.
- The earliest time the event can occur if there are no delays in the schedule. This calculation assumes that each activity takes the expected completion time as defined above. The earliest completion time is the sum of the expected completion times for activities along the critical path to that event.
- The latest time an event can occur without delaying the final event. For events on the critical path, the latest time is the same as the earliest time.
- The total slack between the earliest time and the latest time.

Events along the critical path to the final event are indicated by zero slack time.

There are no fixed limits to the size of the network that can be analyzed. However, there is a dynamic trade-off between the total number of events and the number of different paths to the final event. Should this limit be exceeded, an appropriate message is printed on the terminal.

A unique feature of this program is the ability of the user to change the activity times and recompute. See the description of the MODIFY command on page 46.

Comparison of CPM and #PERT

Both the CPM[1] package and #PERT analyze a PERT network and print the earliest and latest times an event can occur within a schedule, the critical path to the final event, and the slack allowable for each event. In addition, the CPM package calculates the total cost for the

1 - For a detailed discussion of CPM, see page 23.

project and the position of the event in the network. In the CPM package, the expected duration for each event is entered explicitly. In #PERT, the user enters the most optimistic, most pessimistic, and most likely durations. This allows #PERT to calculate not only the expected duration but also a variance for each duration. Therefore, #PERT calculates a variance for the occurrence of each event which is a composite of the variances of the activities leading to that event.

The CPM package prints actual dates and takes holidays into account. This makes the CPM results especially easy to understand. The CPM package has other features for the convenience of the user such as sorting the output table, detailed network checking before a calculation is attempted, and several output options.

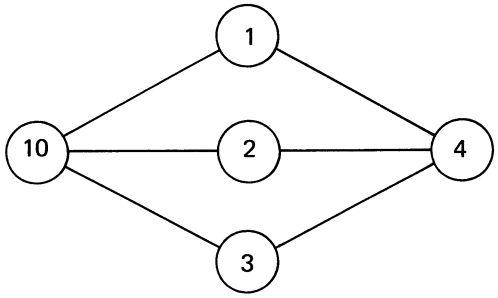
Instructions

To perform a PERT analysis, the user draws a PERT chart, creates a data file, and then types #PERT ↵ in the EXECUTIVE.

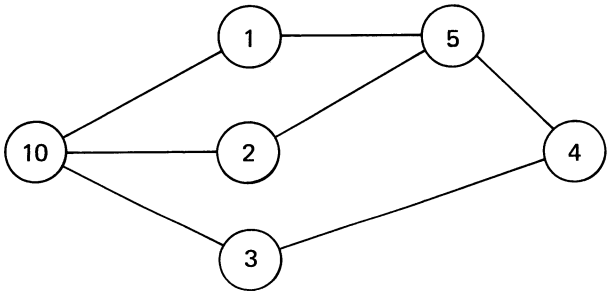
1. **Creating a PERT Chart**

   A PERT network shows the individual activities involved in completing a project. The network is usually drawn as a collection of arrows with each arrow representing an activity. At the ends of each arrow are circles representing events. An event occurs only when all of the activities preceding it are completed.

   An activity is identified by its starting and ending events; therefore, only one activity can connect any two events. Furthermore, no more than two activities can end at each event. If three or more activities are required for an event, dummy activities and events can be used to simulate the project. The dummy activities have duration times of zero. For example,



becomes



where the activity from 5 to 4 takes no time.

Each event is assigned a number. The number can be between 1 and 999; the initial event is number 0. *NOTE: There can be only one final event.*

Each activity is assigned a most optimistic time, a most pessimistic time, and a most likely time.

2. **Creating a Data File**

The data file can be created in EDITOR or by a program. In the example, the file is created in EDITOR. *NOTE: If the data is to be entered directly into #PERT from the terminal, there is no need to create a data file.*

The first entry in the file is the total number of events in the network, with a line of data for each event. Each line contains nine entries each of which must be separated by a comma or a space.

Entry Number	Contents
1	The number of the event.
2	The number of the first event prior to this event. If none, zero is entered.
3-5	The activity times governing these events: optimistic, pessimistic, most likely.
6	The number of the second event leading to this event. If none, zero is entered.
7-9	The activity times for the second activity leading to the event. If none, zeros are entered.

3. **Using #PERT**

#PERT is called from the EXECUTIVE by typing #PERT ↵. When the colon appears, the user types RUN ↵.

The program is conversational; the user simply answers the questions. The questions are:

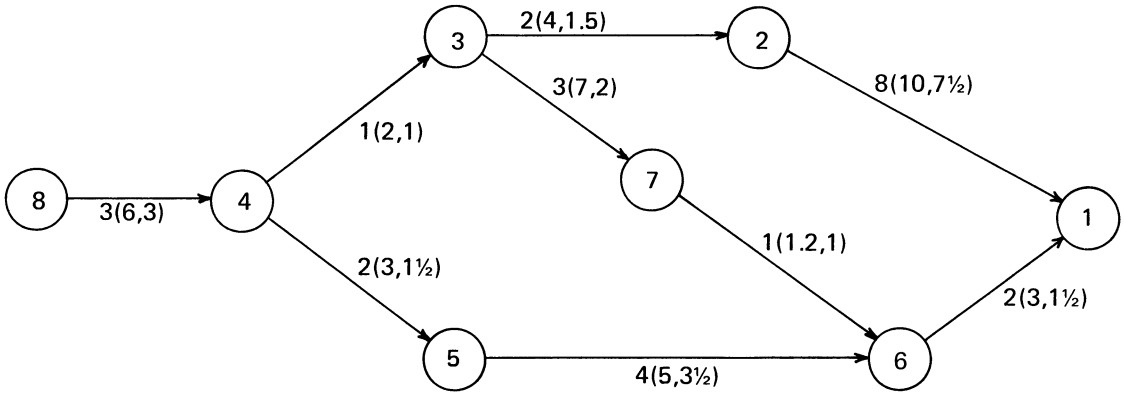
INPUT FILE?	The user enters the data file name or T for the terminal. If input is from the terminal, the user enters the data in the order described under Creating a Data File, above.
THE FINAL EVENT IS LABELED?	The number of the final event is entered. There can be only one final event.
OUTPUT FILE?	The user enters the name of an output file or T if output is to the terminal.

#PERT then prints a table of the times, slack, and variance for each event.

The MODIFY command is used to change any activity times. The data file is not affected; therefore, the change is made only for the subsequent run.

After the network has been analyzed, #PERT displays the colon (:). To alter the network, the user types MODIFY ↵, and gives the number of data lines to be changed and the new data. When the colon reappears, the user types RUN ↵ to reexecute.

Example



PERT Chart

The numbers associated with each activity are C(B,A), where C is the most likely time for completion of the activity, B is the most pessimistic time for completion, and A is the most optimistic time for completion.

```
-EDITOR↵
*APPEND↵
8↵
8,0,0,0,0,0,0,0,0↵
4,8,3,6,3,0,0,0,0↵
3,4,1,2,1,0,0,0,0↵
5,4,1.5,3,2,0,0,0,0↵
7,3,2,7,3,0,0,0,0↵
6,7,1,1.2,1,5,3.5,5,4↵
2,3,1.5,4,2,0,0,0,0↵
1,2,7.5,10,8,6,1.5,3,2↵
*WRITE PERTDATA↵
 NEW FILE↵
159 CHARACTERS
*QUIT↵
```

The user types D^C to terminate the EDITOR APPEND mode.

```
-#PERT↵

:RUN↵

PROGRAM EXECUTION.....

INPUT FILE? PERTDATA↵

THE FINAL EVENT IS LABELED ? 1↵

OUTPUT FILE ? T↵
```

EVENT NO.	TOTAL VARIANCE	EARLIEST COMPLETION	LATEST COMPLETION	TOTAL SLACK
1	1.04	15.17	15.17	.00
2	.45	6.92	6.92	.00
3	.28	4.67	4.67	.00
4	.25	3.50	3.50	.00
8	.00	.00	.00	.00
6	.38	9.67	13.08	3.42
5	.31	5.58	9.00	3.42
7	.97	8.17	12.05	3.88

```
:QUIT↵

-
```

SECTION 3
INFORMATION RETRIEVAL

RETRIEVE
INFORMATION RETRIEVAL SYSTEM

Description

RETRIEVE is a general information retrieval system used to process a uniformly formatted data base. The program is described in greater detail in a separate manual which can be obtained from a Tymshare representative.

Instructions

To call the program, the user types RETRIEVE↵ in the EXECUTIVE. The program responds with

PLEASE NAME YOUR DATA BASE:

The user enters the name he wants to assign to the data base. The name can be used to refer to the same data base in the future. If the data base is new, RETRIEVE requests a structure. The user enters the name and length of each item and the type (N=numeric; C=alphanumeric).

RETRIEVE displays a period when it is ready to accept a command. The RETRIEVE commands are:

Command	Action
APPEND [FROM file]*	Adds records to the data base from the terminal or a file.
CAPABILITIES	Describes RETRIEVE capabilities.
CHANGE items [FOR conditions]*	Requests new values for the specified items in the records satisfying the conditions.
CHARGES	Lists the premium charges.
COUNT [FOR conditions]*	Counts the number of records satisfying the conditions.
CREDITS	Lists the author of the program.
DELETE [FOR conditions]*	Deletes the records satisfying the conditions.
DO command file	Takes the subsequent RETRIEVE commands from a command file.
FAST [items] [FOR conditions]	Lists, without headings, the items in the records satisfying the conditions.
HELP (or ?)	Lists all RETRIEVE commands with a brief description of each.

Command	Action
HUSH	Suppresses extraneous printing.
INSTRUCTIONS	Prints the instructions for using RETRIEVE.
LIST [items] [FOR conditions]	Lists the specified items with headings and record numbers for all the records satisfying the conditions.
LOAD file*	Loads a new data base.
PRINT [items] [FOR conditions]*	Lists, with headings, the specified items for records satisfying the conditions.
QUIT*	Returns the user to the EXECUTIVE.
RECOVER	Recovers an old data base.
REPLACE items WITH expression [FOR conditions]*	Replaces the specified items with the value of the expression for all records satisfying the conditions.
REPORT [FOR conditions]	Generates a report of all records satisfying the conditions.
SAMPLE	Prints a sample run.
SAVE file [FOR conditions]*	Saves on the file all records satisfying the conditions.
SIZE	Prints the total number of records.
SORT BY items*	Sorts the data base on the items specified.
STRUCTURE	Lists the data base structure.
SUM expression [FOR conditions]*	Adds the values of the expression for all records satisfying the conditions.
TALK	Negates HUSH.
TYPE 'message'	Prints a user-specified message on the terminal.
VERSION	Prints the RETRIEVE version number.

The commands marked with an asterisk are illustrated in the example below.

Example

```
-RETRIEVE↵

PLEASE NAME YOUR DATA BASE: ACCOUNTS↵

I NEED TO KNOW THE STRUCTURE OF YOUR DATA BASE.
PLEASE DESCRIBE EACH ITEM


ITEM  NAME SIZE TYPE

 1      CUSTOMER,10,C↵
 2      ACCT,5,C↵
 3      ITEM,5,C↵
 4      QUANTITY,5,N↵
 5      PRICE,5,N↵
 6      SHIPPING,5,N↵
 7      YR,2,N↵
 8      MO,2,N↵
 9      DA,2,N↵
10      ↵

ACCOUNTS CURRENTLY CONTAINS 0 42-CHARACTER RECORDS.
ACCOUNTS'STR.D' NOW CONTAINS BASE STRUCTURE.

.APPEND↵

CUSTOMER    ACCT   ITEM   QUANTITY
                                PRICE SHIPPING
                                          YR MO DA
ACE HDWARE,284A,35575,340,.18,3.05,69,10,15↵
ZIMMER CO,111B,45967,95,.12,0.00,69,9,23↵
SIMON INC,201A,87567,100,.05,0.00,69,11,28↵
LD SYSTEM,177A,45907,250,.10,6.00,70,1,15↵
WATSON MFG,108,23454,80,.15,0.00,69,12,29↵
COX&SONS,310B,45967,200,.24,4.19,68,12,27↵
↵

 6 RECORDS PROCESSED

.SORT BY CUSTOMER↵

ACCOUNTS'OLD' CONTAINS YOUR UNSORTED DATA BASE.
ACCOUNTS IS NOW SORTED.
SHALL WE RETAIN ACCOUNTS'OLD'? YES↵
```

```
•PRINT↵

CUSTOMER    ACCT   ITEM   QUANTITY
                              PRICE SHIPPING
                                          YR MO DA
ACE HDWARE  284A   35575    340    •18   3•05 69 10 15
COX&SONS    310B   45967    200    •24   4•19 68 12 27
LD SYSTEM   177A   45907    250    •10   6•00 70  1 15
SIMON INC   201A   87567    100    •05   0•00 69 11 28
WATSON MFG  108    23454     80    •15   0•00 69 12 29
ZIMMER CO   111B   45967     95    •12   0•00 69  9 23

 6 RECORDS PROCESSED

•REPLACE SHIPPING WITH SHIPPING+10•00 FOR YR<69↵


 1 RECORDS PROCESSED

•PRINT FOR YR=68↵

CUSTOMER    ACCT   ITEM   QUANTITY
                              PRICE SHIPPING
                                          YR MO DA
COX&SONS    310B   45967    200    •24  14•19 68 12 27

 1 RECORDS PROCESSED

•CHANGE PRICE FOR QUANTITY>200↵

PRICE

  •18
•16↵
  •10
↵              Typing only a Carriage Return leaves the item unchanged.

 2 RECORDS PROCESSED

•REPLACE SHIPPING WITH SHIPPING + 1•75 FOR SHIPPING<1•00↵


 3 RECORDS PROCESSED

•DELETE FOR YR<69↵


ACCOUNTS'OLD' CONTAINS YOUR UNCHANGED DATA BASE•
 1 RECORDS DELETED•
ACCOUNTS NOW CONTAINS 5 RECORDS
SHALL WE RETAIN ACCOUNTS'OLD'? NO
```

```
•COUNT FOR 'A' IN ACCT↵

 3 RECORDS PROCESSED

•SAVE PRIME FOR 'A' IN ACCT↵

 NEW FILE↵

 3 RECORDS PROCESSED

•SUM QUANTITY*PRICE+SHIPPING↵

SUM IS 122.1

 5 RECORDS PROCESSED

•LOAD PRIME↵

PRIME CURRENTLY CONTAINS 3 42-CHARACTER RECORDS.

•FAST↵

ACE HDWARE 284A  35575   340   .16  3.05 69 10 15
LD SYSTEM  177A  45907   250   .10  6.00 70  1 15
SIMON INC  201A  87567   100   .05  1.75 69 11 28

 3 RECORDS PROCESSED

•QUIT↵

-
```

#STATLINK

Description

The program #STATLINK allows the user to perform statistical analyses on data from a RETRIEVE data base. #STATLINK extracts the appropriate information from a RETRIEVE data base and creates an input file for STATPAK. STATPAK is Tymshare's multipurpose statistical analysis package.[1] STATPAK fits curves, calculates correlations, and analyzes distributions.

#STATLINK can also create an input file for #FORECAST. #FORECAST uses a sophisticated mathematical model to predict future values based on historical data. Refer to the description of #FORECAST on page 29.

By changing the first line of the file, the user can use a #STATLINK file for the Critical Path Method program (page 23).

Instructions

#STATLINK is called from the EXECUTIVE by typing #STATLINK↵. The program first asks WHAT IS YOUR DATA BASE? The user enters the name of the RETRIEVE data base. In response to subsequent questions, the user enters the name of the output file, the number of items to be included on the file, and the item names. All of the items must be numeric.

#STATLINK creates a file with the number of items and the number of records on the first line and with the specified items from each record in the same order as they appear in the RETRIEVE data base. Any selection of records should be done in RETRIEVE with the SAVE[2] command.

Example

```
-RETRIEVE↵

PLEASE NAME YOUR DATA BASE: SALES↵

SALES CURRENTLY CONTAINS 11 58-CHARACTER RECORDS.

.STRUCTURE↵

ITEM TYPE WIDTH NAME

  1    C      5  ITEM
  2    C     17  CUSTOMER
  3    C      2  AREA
  4    N      4  ORDER
  5    N      6  DATE
  6    N      3  QTY
  7    N      6  PRICE
  8    C      8  SALESMAN
  9    N      6  TOTAL

.QUIT↵
```

1 - Refer to the *Tymshare STATPAK Reference Manual.*

2 - For a detailed explanation of the SAVE command, see page 50.

```
-#STATLINK↵
WHAT IS YOUR DATA BASE? SALES↵

WHAT IS THE FILE TO BE USED IN STATPAK? SALEANAL↵

HOW MANY ITEMS ARE TO BE USED? 3↵

WHAT ARE THE ITEMS? ORDER,DATE,TOTAL↵

JOB COMPLETED -- YOU MAY NOW CALL STATPAK

-STATPAK↵                    The user continues the execution of STATPAK.
•
•
•
```