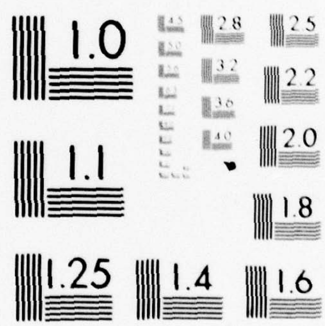
AD-A073 971 ASSESSMENT GROUP SANTA MONICA CA* F/G 5/1
DEMONSTRATION MODEL SYSTEM. VOLUME IV. SLIDE-RULE MODEL SYSTEM --ETC(U)
JUL 79 T M NECHES N00014-78-C-0465
UNCLASSIFIED AG-PR-A101-VOL-4 NL

| OF |

AD
A073971

END
DATE
FILMED
10-79
DDC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

theassessmentgroup
710 Wilshire Boulevard, Suite 301
Santa Monica, California 90401
213/394-6778

(12)

(14) AG-PR-A1Ø1-VOL-4

(6) DEMONSTRATION MODEL SYSTEM.

VOLUME IV:

SLIDE-RULE MODEL SYSTEM
PROGRAM MANUAL.

by

(10) Thomas M./Neches

A073970

(11) July 1979

(12) 57p.

(15) Contract No. N00014-78-C-0465

Submitted to:

Office of Naval Research
Department of the Navy
Arlington, Virginia

393 637

JOB

TABLE OF CONTENTS

1.0 INTRODUCTION

The Slide-Rule Life Cycle Cost Model System (SRS) has been designed as an aid to system, subsystem and assembly designers in making cost estimates and trade-offs early in the design process. At this stage it is still possible for cost analysis to influence design - system cost has not yet been "locked in" due to the lack of flexibility in system configuration which occurs in the later phases of design.

The SRS consists of four linked programs implemented on a Texas Instruments TI-59 programmable calculator coupled to a TI PC-100A printer. Each program is appropriate to a different design phase and aggregation level. The first estimates the life cycle costs of a system by making simplifying assumptions about its subelements; the second is used for the design of a single Lowest Removable Assembly (LRA); the third estimates system or subsystem costs by aggregating the costs of its subelements, computed in the second program; the fourth is a specialized program used to compute the achieved system confidence level against a stock-out of spare parts.

The running times of all programs are less than one minute. This, combined with the "no-cost" running feature of the program, makes the SRS an excellent design tool for experimenting with design/cost trade-offs early and often in the design process.

Sections 2 through 5 describe each program, including the cost equations, input variable definitions, and flow charts, where appropriate. Annotated program listings are provided in four Appendices.

2.0 THE TOP-DOWN MODEL

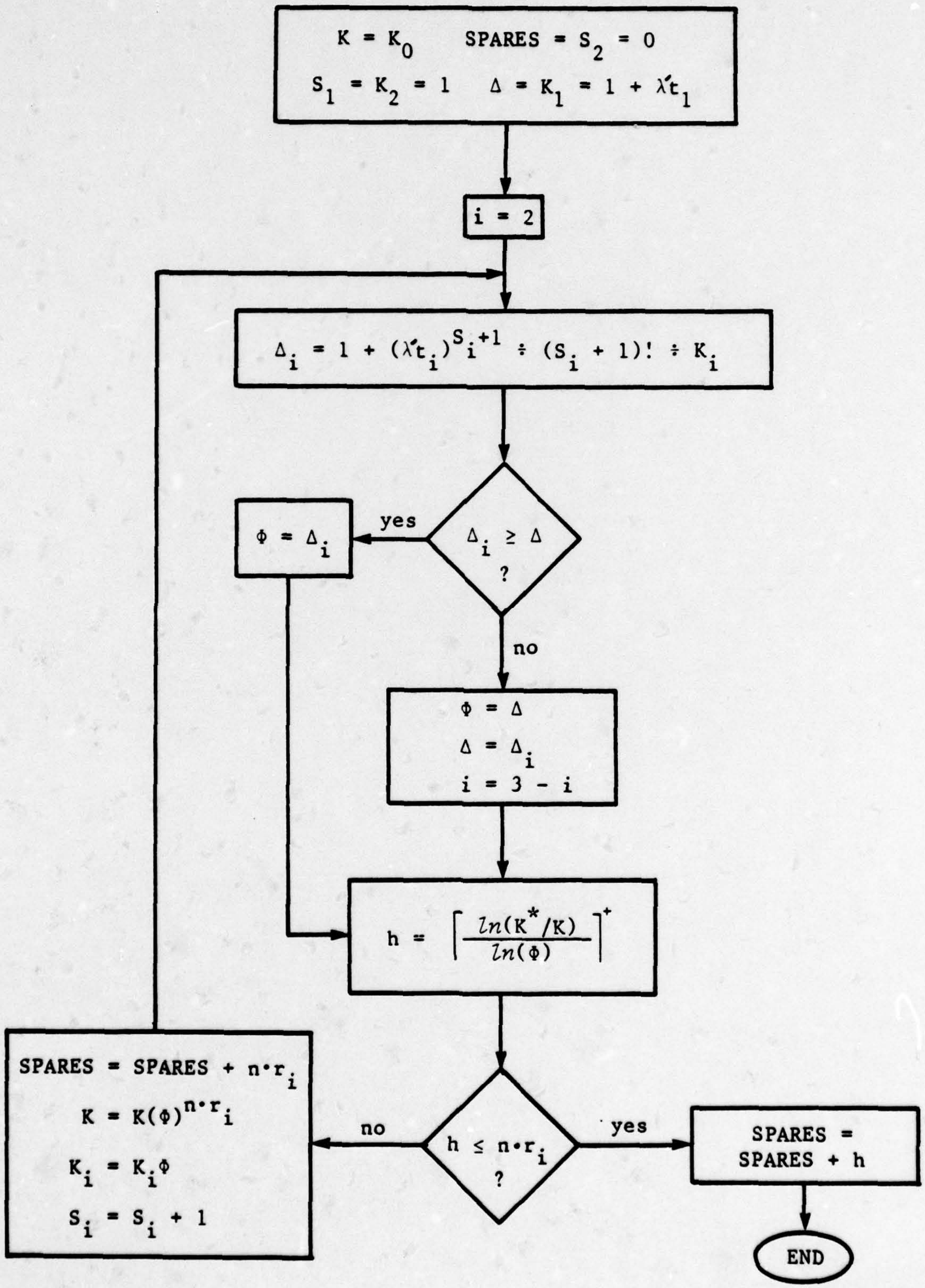
The Top-Down Model (TDM) computes total life-cycle cost as the sum of nine cost categories: maintenance personnel compensation, maintenance training, operator compensation, operator training, production and spares, support and test equipment, repair costs, item entry and management, and documentation. The first seven cost categories are computed for each ship and then multiplied by the number of ships on which the system is to be deployed.

The program requires as input 27 data elements which characterize the design of a system. An additional 21 constants, describing the operating environment of the system, are incorporated into the program code. Design/cost trade-offs are accomplished by altering the input data elements and observing the effect on life-cycle cost.

Some of the main features of the cost model are: a sophisticated routine for determining on-board spares for repairable items (a flow chart of this routine is presented in Figure 1, a learning curve routine which adjusts the estimated unit production cost of the system based on the total number of systems and spares procured, discounting to present value of recurring costs (this option can be suppressed, if desired), and a new approach to manpower costing based on the concept of opportunity cost.

Program operation is simple and quick. Data elements are input to the model by storing them in appropriate memory registers. Execution is initiated by pressing a single button. Output consists of each of the cost categories and total life-cycle cost. To further

Figure 1 TDM Spares Algorithm



simplify the output, the printing of any or all of the cost categories can be suppressed by setting an appropriate flag. Turn-around time for cost results is approximately one minute.

The next sections summarize the TDM cost equations and present detailed definitions of the input variables.

TDM Cost Equation Summary

System failure rate: $\lambda = Q \cdot AHR/MTBF$

LRA peak failure rate: $\lambda' = s\lambda/n$

On-board spares: $S = SPARES/n'$

Demand at depot: $\mu = \lambda' \cdot N/d \cdot DRT$

Depot spares: $B = \lceil \mu + Z_b \sqrt{\mu} \rceil \cdot d \cdot r_2 \cdot n/n'$

Replenishment spares: $S' = \lambda \cdot D \cdot h \left[1+(r_1+r_2)(1-COND)\right]/n'$

Adjusted unit production cost: $UC = UC_\ell \left[\frac{N(Q+S+S')+B}{\ell}\right]^{\log RRATE/\log 2}$

Life cycle discount factor:

$$L = \begin{cases} \frac{(1+\rho)^{LC}-1}{\rho(1+\rho)^{LC}} & \rho \neq 0 \\ LC & \rho = 0 \end{cases}$$

Peak operator demand: $M'_o = s \cdot Q \cdot \theta \cdot PHR/WH_o$

Peak maintenance manpower demand: $M'_m = (s\lambda MTTRS(1+\tilde{M}r_1)+Q \cdot SM)/(U \cdot WH_m)$

Maintenance "C" training course cost: $TC_m = TS_m(1+\tilde{T}nr_1)$

TDM Cost Equation Summary (cont'd)

Maintenance wage: $C_1 = \text{Wage}(M_m', AN_m, BN_m)$

Maintenance training: $C_2 = \text{Trn}(M_m', AN, TC_m)$

Operator wage: $C_3 = \text{Wage}(M_o', AN_o, BN_o)$

Operator training: $C_4 = \text{Trn}(M_o', AN_o, TC_o)$

Production and spares: $C_5 = [(Q+S+S'L)\cdot N + B]UC$

Support and test equipment: $C_6 = STE(1+SGM(r_1)\tilde{S})(1+mL)\cdot N$

Repair: $C_7 = \lambda\cdot D\cdot h(r_1 RP + r_2 COD)\cdot N$

Item entry and management: $C_8 = IECn + IMC\cdot L\cdot n\,[\,1 + PP(r_1 + r_2)]$

Technical data: $C_9 = DOC(1+\tilde{D}r_1 n)$

Life cycle cost: $LCC = \sum_{j=1}^{9} C_j$

TDM Cost Equation Summary (cont'd)

Personnel Costs:

$$A(M') = \lceil M' - \min(M', AN) \rceil$$

$$Wage(M', AN, BN) = \left[M \cdot BG + A(M')(BN - BG) \right] \cdot L \cdot N,$$

$$Trn(M', AN, TC) = \left[\lceil M' \rceil \cdot TC + A(M')TA \right] (1 + TOR \cdot L) \cdot N,$$

TDM Input Variable Summary

r_1	The fraction of LRA types in the system coded local repair.
r_2	The fraction of LRA types in the system coded depot repair. $1-r_1-r_2$ is the fraction of LRA types coded discard on failure.
LRT (weeks)	The average time for an item coded local repair to be returned to ready for issue status.
D (weeks)	The length of the deployment period.
n'	The total number of LRA's in the system.
n	The number of LRA types in the system. (Each LRA appears an average of n'/n times in the system.) Note: because of logic of the sparing algorithm, the smaller the value of n, the longer the program running time.
s	The ratio of peak operating hours to average operating hours. The input s can be seen as a policy variable which determines the ability of the supply system to withstand periods of increased activity.
N	The number of ships on which the system is to be deployed.
AN_m	The size of the pool of available, trained maintenance personnel on-board ship.
BN_o ($'000)	Annual billet cost for trained personnel used to maintain the system. Value taken from the Billet Cost Model (BCM). The undiscounted value should be used.
TC_m ($'000)	System repair training course cost for maintenance personnel. Does not include the course cost of training to repair individual LRA's coded local repair.
AN_o	Same as AN_m for operators.
BN_o	Same as Bn_m for operators.
TC_o ($'000)	Operator training course cost.

TDM Input Variable Summary (cont'd)

LC (years)	Length of system life cycle.
MTBF (hours)	Mean time between failure of the system. The value used should be adjusted for fixed field operations.
UC_ℓ (\$'000)	The estimated unit production cost of the system assuming that ℓ units are produced.
ℓ	The lot size used to define UC_ℓ.
Q	The number of systems deployed per ship.
AHR (hr./wk.)	The average weekly operating hours of the system per operating week.
MTTRS (manhour)	The number of manhours required to restore the system to operational status after the failure of an LRA.
SM (man-hr./wk.)	The weekly scheduled maintenance requirement. Includes all facility and preventative maintenance.
θ	The number of operators required to man the system when fully operational.
STE (\$'000)	Purchase cost of all support and test equipment necessary for the repair of the system. Does not include common or specific STE used for the repair of failed LRA's.
COD (\$'000)	The average cost of a repair at a contractor operated depot repair facility. COD includes the round-trip transportation cost of the item in addition to all other costs, direct and indirect, of a repair.
RP (\$'000)	The average repair parts material cost required for the local repair of an LRA.
DOC (\$'000)	The cost of documentation for system operation and repair. Does not include the documentation cost for the repair of individual LRA's.

TDM Variable Summary (Code Constants)*

BG ($´000)	The undiscounted annual billet cost for general labor personnel. Value taken from the BCM.
TA ($´000)	Average cost of "A" school training for operators and maintenance personnel.
TOR ($´000)	Average annual personnel attrition rate for military personnel.**
K*	Desired system level confidence level against stock-out of on-board spares.
DRT/d (weeks)	The average time for an item sent to a repair depot to be returned to the holding depot stock-pile divided by the number of holding depots.
Z_b	The number of standard deviations from the mean required to achieve the desired confidence level against stock-out at the depot.
d	The number of holding depots at which failed LRA's from ships are replaced by ready-to-issue LRA's.
h	The number of deployments in a year.
1-COND	1 minus the ratio of failed LRA's coded repair which cannot be repaired to the total number of LRA's coded repair. COND is an average value for local and depot repair facilities.**
log RRATE/log 2	The learning curve cost reduction coefficient. Equal to the log of the reduction rate divided by log 2.
ρ	Annual discount rate.**
$\tilde{M}$	The ratio of the mean time to repair a failed LRA coded local repair to the mean time to remove and replace the LRA (MTTRS, defined above).

*Code constants are input variables, describing the operating environment in which the system is placed, which have been incorporated into the code of the cost model programs.

**Default value taken from "Naval Air Systems Command Avionics Level of Repair Model, Mod III Default Data Guide, 1 July 1977," NWESA, Washington Naval Yard.

TDM Variable Summary (Code Constants)* (cont'd)

$WH_m \cdot U$ (hr./wk.)	Available weekly work hours at sea for maintenance personnel (non-watchstanders), times the utilization rate, which serves to decrease available work time by accounting for delays arising from fatigue, environmental effects, personal needs, unavoidable interruptions, in addition to maintenance put-away, administrative, and overhead time.***
$\tilde{T}$	The ratio of the average training course cost specific to the repair of an individual LRA coded local repair to the system repairs training course cost (TC_m, defined above).
WH_o	Available weekly work hours for operators, that is, total available hours weekly (168) minus sleep, messing, personal needs and free time.***
$\tilde{S}$	The ratio of purchase cost of the common support and test equipment which would be needed if any of the LRA's in the system are coded local repairs to system repair STE (STE, defined above).
m	Annual support of support equipment maintenance factor.**
IEC ($´000)	Cost of entering a new item into the Naval Stock System (NSN) inventory.**
IMC ($´000)	Annually recurring cost of retaining an item in the NSN.**
PP	The average number of unique new components in each LRA type; nPP gives the total number of new components in the system.
$\tilde{D}$	The ratio of the cost of specific documentation required for an individual LRA coded local repair to the cost of system repair documentation (DOC, defined above).

**Default value taken from "Naval Air Systems Command Avionics Level of Repair Model, Mod III Default Data Guide, 1 July 1977," NWESA, Washington Naval Yard.

***Default value taken from OPNAV 10P-23, "Guide to the Preparation of Ship Manning Documents, Volume I: Policy Statement," OPNAV 10P-23 Washington, D.C., 1971.

3.0 The Lowest Removable Assembly Model

The Lowest Removable Assembly Model (LRAM) computes total life-cycle cost as the sum of seven cost categories: maintenance personnel wage, maintenance personnel training, production and spares, support and test equipment, repair, item entry and management, and documentation. The first five categories are multiplied by the number of ships on which the LRA is to be deployed.

The program requires as input 16 data elements which characterize the design of the LRA. An additional set of 21 variables are input to the model on an operating environment card provided by the system designer. This card also includes the printing op. codes for the program output labels. Design/cost trade-offs are accomplished by altering the input data element and observing the effect on life cycle cost.

Some of the main features of the model are: level of repair analysis capabilities which include local repair, depot repair and discard options, a sophisticated spares routine which automatically calculates the (near) optional mix of on-board and depot spares required to meet the LRA confidence level, complete manpower cost formulations for maintenance personnel, and full output labeling capabilities. As in the TDM, printing of individual cost categories can be suppressed by setting appropriate flags. Turn-around time for cost results is approximately forty seconds.

The next sections summarize the LRAM cost equations and present detailed definitions of input variables. Flow charts of model subroutine logic are presented in Figure 2.

Figure 2 LRAM Subroutine Flowcharts

Figure 2-A

S(t,K): Number of spares needed to meet confidence level K given demand lead time t.

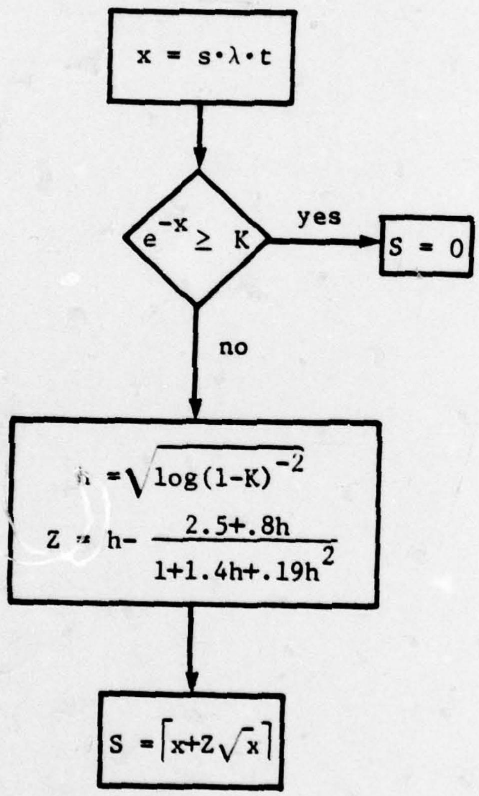
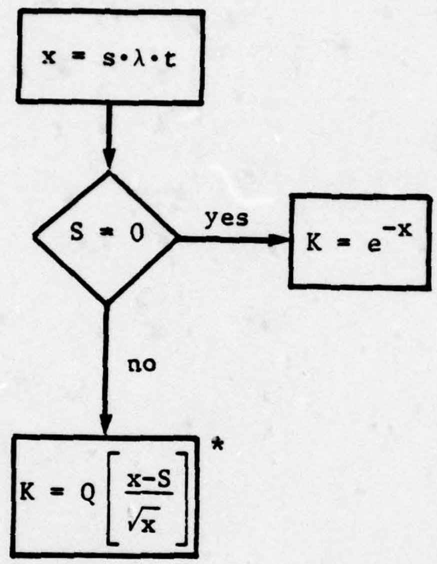


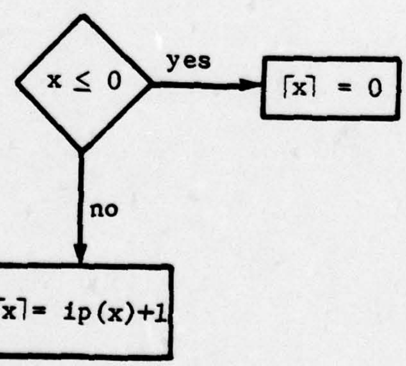
Figure 2 LRAM Subroutine Flowcharts (cont.)

Figure 2-B

K(t): Confidence level achieved with S spares and demand lead time λt.



⌈x⌉ : Round-up to next higher integer.



---

*Q(x) is computed in a program contained in the master library module which comes with the TI-59 calculator.

LRAM Cost Equation Summary

LRA peak failure rate: $\lambda = sqQ\delta AHR/MTBF$

Lead time: $t = r_1 LRT + r_2 D$

On-board spares: $S = S(t,K)$

Depot confidence level: $b = \dfrac{K-K(XD)}{K(D)-K(XD)}$

Depot spares: $B = r_2 S(N/d \cdot DRT, b) \cdot d$

Replenishment spares: $S' = \lambda \cdot D \cdot h\ 1-(r_1 + r_2)(1-COND)$

Adjusted unit production cost: $UC = UC_\ell \left[\dfrac{N(q \cdot Q+S+S') + B}{\ell}\right]^{\log RRATE/\log 2}$

Maintenance manpower peak demand: $M_m' = s\lambda(MTPR + r_1 MTTR)/(U \cdot WH_m)$

"A" school training requirement: $A(M_m') = \lceil M_m' - \min(M', AN_m) \rceil$

Maintenance wage: $C_2 = [M_m'/s \cdot BG + A(M_m')(BN_m - Bg)] L \cdot N$

Maintenance training: $C_3 = [M_m' (TFI + r_1 TR) + A(M_m') TA_m] \cdot (1+TOR \cdot L) \cdot N$

Production and spares: $C_4 = [(q \cdot Q+S+S'L)N + B]UC$

LRAM Cost Equation Summary (cont'd)

Support and test equipment:	$C_5 = (STE+r_1 STE_{rpr})(1+mL)\cdot N$
Repair:	$C_6 = \lambda D\cdot h\cdot L(r_1 UC/c+r_2\cdot COD)\cdot N$
Item entry and management:	$C_7 = (IEC+IMC\cdot L)(1+(r_1+r_2)\bar{c})$
Technical data:	$C_8 = P_f + r_1 P_r$
Life cycle cost:	$LCC = \sum_{j=2}^{8} C_j$

LRAM Input Variable Summary

q	Number of LRA appearances in the system as a whole.
δ	LRA duty cycle: the ratio of LRA to system operating hours.
MTBF (hrs.)	LRA mean time between failure.
r_1	Level of Repair switch which is set equal to 1 if the LRA is coded local repair, otherwise it is set equal to zero.
r_2	Level of Repair switch which is set equal to 1 if the LRA is coded depot repair, otherwise it is set equal to zero. If $r_1 = r_2 = 0$ the LRA is coded discard on failure.
UC_ℓ ($'000)	Estimated unit cost of the LRA assuming a production lot size of ℓ.
MTTRS (manhrs.)	The average manhours required to restore the system to operational status after a failure of the LRA. Equivalently, the mean time to fault isolate to, remove and replace the LRA upon failure.
MTTR (manhrs.)	The average maintenance manhours needed to repair the LRA if it is coded local repair.
TFI ($'000)	The addition to the system level maintenance training course cost needed to train personnel to fault isolate to, remove and replace this particular LRA. (In other words, $TFI > 0$ only if this LRA would require special mention in the system repair training course.)
TR ($'000)	The course cost required to train personnel to repair the LRA locally.
STE ($'000)	The purchase cost of any additional system repair support and test equipment needed to fault isolate to, remove and replace this particular LRA.

LRAM Input Variable Summary

STE_{rpr}	(\$'000)	The purchase cost of all support and test equipment necessary to the local repair or maintenance of this particular LRA.
c		The number of components in the LRA. (Repair of the LRA consists of removing and replacing components.)
$\bar{c}$		The number of new components, unique to the LRA, which must be entered in the Naval inventory management system.
DOC	(\$'000)	The addition to the system repair documentation cost needed to document the fault isolation, removal and replacement of this LRA.
DOC_{rpr}	(\$'000)	The cost of documentation of the procedure to repair the LRA locally.

All other variables are as defined in the TDM, with three exceptions:

K	is the desired confidence level against stock-out for the LRA, K_i's for each LRA are assigned so that $\prod K_i = K^*$.
AN_m	is the available pool of trained maintenance personnel available to the LRA. $AN_{m,i}$'s are assigned so that $\sum AN_{m,i} = M$, the desired system level maintenance manpower requirement.
L	is the discounted life cycle computed in the TDM.

4.0 THE SYSTEM AGGREGATION MODEL

The System Aggregation Model (SAM) computes life cycle cost as the sum of the same nine cost categories as the TDM. The input of the program is the output of the LRAM for each LRA type used in the system, plus system level input data. Design trade-offs are accomplished by altering the number and type of LRA's used to build up the system.

One of the most powerful features of the SAM is that SAM program output can be input to the model. Thus, for example, the SAM can be used to aggregate LRA's into sybsystems, and then used again to aggregate these subsystems and systems. Mixed aggregation levels are possible; input to the SAM can consist partially of LRA's (LRAM output) and partially of subsystems (LRAM output preaggregated using the SAM). And, of course the multiple aggregation-level option is available.

Other features of the SAM include: automatic calculation of achieved system MTBF and MTTR, based on the MTBF's and MTTR's of the LRA's used in a given system configuration; manpower cost calculations based on aggregated personnel demand and training course requirements, and complete output labeling capabilities. As in the other models, intermediate cost outputs can be suppressed by setting appropriate flags. Program running time is approximately ten seconds per LRA input, plus an additional 20 seconds to compute and print system level costs.

The next sections present a summary of SAM cost equations and input variables.

SAM Cost Equation Summary

Aggregation Factor:	$R_i = \frac{q \cdot QIPA_i}{q_i}$
Peak maintenance demand:	$M'_m = Q \cdot SM / U \cdot WH_m + \sum^n R_i M'_{m,i}$
Maintenance training cost:	$TC_m = TS_m + \sum^n R_i TC_{m,i}$
Operator training cost:	$TC_o = TS_o + \sum^n R_i TC_{o,i}$
Peak operator demand:	$M'_o = s \cdot Q \cdot q \cdot \theta \cdot AHR / WH_o$
Maintenance wage:	$C_1 = Wage(M'_m, AN_m, BN_m)$
Maintenance training:	$C_2 = Trn(M'_m, AN_m, TC_m, TA_m)$
Operator Wage:	$C_3 = Wage(M'_o, AN_o, BN_o)$
Operator training:	$C_4 = Trn(M'_o, AN_o, TC_o, TA_o)$
Hardware:	$C_5 = PT_\ell \left[\frac{N \cdot Q \cdot q}{\ell}\right]^{\log RRATE/\log 2} + \sum^n R_i HRDW_i$
Support and test equipment:	$C_6 = STE_{sys}(1+mL)N + \sum^n R_i STE_i$
Repair:	$C_7 = \sum^n R_i RPR_i$
Item entry and management:	$C_8 = \sum^n R_i IEMC_i$

SAM Cost Equation Summary (cont'd)

Technical Data:

$$C_9 = DOC_{sys} + \sum^{n} R_i DOC_i$$

Life cycle cost:

$$LCC = \sum_{j=1}^{9} C_j$$

System mean time to repair:

$$MTTR = \frac{\sum QIPA_i MTTR_i \cdot \lambda_i}{\sum QIPA_i \cdot \lambda_i}$$

System MTBF:

$$MTBF = \left[\sum^{n} \frac{QIPA_i}{MTBF_i} \right]^{-1}$$

SAM Input Variable Summary

q		The number of units in the system. If the SAM is being used at the system level, $q = 1$.
SM		Same as above. If the SAM is used at the subsystem level, refers to subsystem scheduled maintenance requirement.
AN_m		Same as above. If the SAM is used at the subsystem level, AN_m is assigned to the subsystem in the same manner as AN_m in the LRAM.
TA_m	(\$´000)	"A" school training course cost for maintenance personnel.
AN_o		Same as AN_m, above, but refers to operators.
TA_o	(\$´000)	"A" school training course cost for operators.
PT_ℓ	(\$´000)	The estimated assembly, or put-together, cost of the system. Equivalently, the total production cost of the system less the production costs of all sub-elements.

All other variables are as defined above.

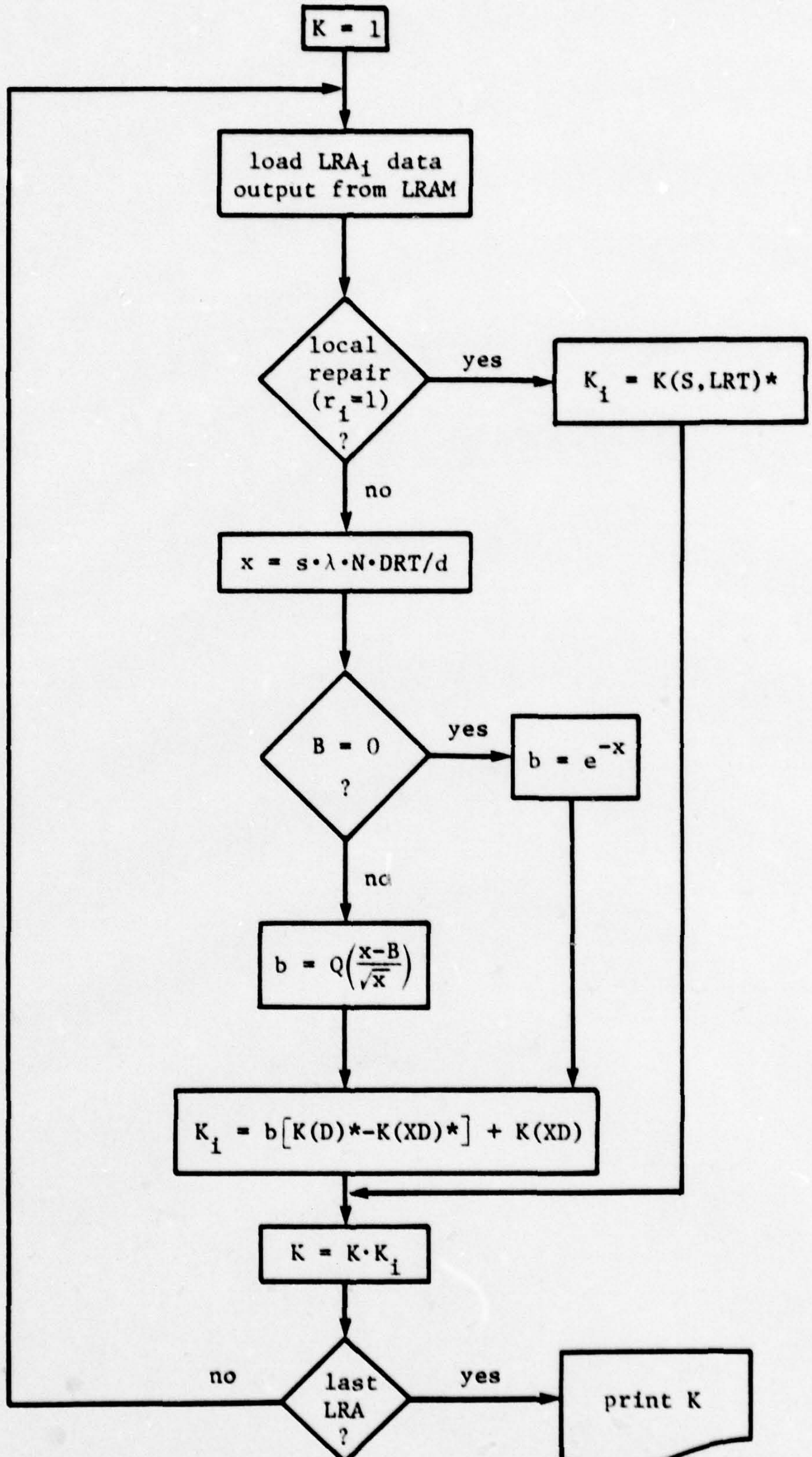
5.0 The System Confidence Model

The System Confidence Model (SCM) is a specialized program which determines the achieved confidence level against stockout for each LRA type and for the system as a whole.

SCM input consists of LRAM output for each LRA type plus a system operating environment card. The program multiplies LRA confidence levels into an accumulation register containing the current system confidence level. Once all LRA's have been read into the SCM, this register, containing the achieved system confidence level, is printed.

A flow chart of SCM logic is presented in Figure 3. All input variables to the SCM are as defined above.

Figure 3 SCM Program Logic



*These values were calculated and stored as part of the sparing algorithm in the LRAM.

APPENDICES:

ANNOTATED PROGRAM LISTINGS FOR

A: TOP-DOWN MODEL

B: LOWEST REMOVABLE ASSEMBLY MODEL

C: SYSTEM AGGREGATION MODEL

D: SYSTEM CONFIDENCE MODEL

APPENDIX A:

TOP-DOWN MODEL

PROGRAM LISTING

INDIRECT RECALL

```
000  76 LBL_
001  43 RCL    Recalls values
002  73 RC*    used in SPARES
003  00  00    and PERSONNEL
004  69 OP     routines
005  20  20
006  92 RTN
```

⌈x⌉

```
007  76 LBL_
008  59 INT
009  95  =
010  32 X:T
011  00  0
012  77  GE  ⌈x⌉ = { 0           x ≤ 0
013  00  00        { ip(x) + 1   x > 0
014  20  20
015  32 X:T
016  59 INT
017  85  +
018  01  1
019  95  =
020  92 RTN
```

PRINT and SUM to LCC

```
021  65  ×
022  43 RCL
023  13  13
024  95  =    LCC = LCC + N·C_j
025  44 SUM   if j ≤ 7 (enter at 021)
026  37  37
027  69 OP    LCC = LCC + C_j
028  21  21   if j ≥ 8 (enter at 024)
029  87 IFF
030  40 IND
031  01  01
032  00  00
033  35  35
034  99 PRT
035  92 RTN
```

SUBROUTINE PERSONNEL

```
036  43 RCL
037  38  38
038  32 X:T
039  43 RCL
040  38  38
041  75  -
042  71 SBR
043  43 RCL
044  22 INV
045  77  GE
046  00  00
047  49  49
048  32 X:T
049  71 SBR
050  59 INT
051  42 STO   A(m', AN) = ⌈m' - min(m', AN)⌉
052  04  04
```

Wage = (m'/6·BG + A(BN - BG))·L·N

```
053  94 +/-
054  85  +
055  43 RCL
056  38  38
057  55  ÷
058  43 RCL
059  12  12
060  95  =
061  65  ×
062  01  1
063  00  0   ← BG
064  93  .
065  05  5
066  85  +
067  71 SBR
068  43 RCL
069  65  ×
070  43 RCL
071  04  04
072  95  =
073  65  ×
074  43 RCL
075  03  03
076  71 SBR   print Wage
077  00  00
078  21  21
```

Trn = ([M⁻¹]TC + A·TA)(1+TOR·L)·N

```
079   43  RCL
080   38   38
081   71  SBR
082   59  INT
083   65   ×
084   71  SBR
085   43  RCL
086   85   +
087   43  RCL
088   04   04
089   65   ×
090   01   1   ← TA
091   00   0
092   93   .
093   00   0
094   95   =
095   65   ×
096   53   (
097   01   1
098   85   +
099   93   .   ← TOR
100   04   4
101   05   5
102   65   ×
103   43  RCL
104   03   03
105   54   )
106   71  SBR
107   00   00  Print Trn
108   21   21
109   92  RTN
```

end of subroutine PERSONNEL

BEGIN EXECUTION

```
110   76  LBL
111   11   A
112   43  RCL
113   24   24
114   65   ×
115   43  RCL
116   25   25
117   55   ÷
118   43  RCL
119   21   21
120   95   =
121   42  STO   λ= AHR·Q/MTBF
122   37   37
123   65   ×
124   43  RCL
125   12   12
126   55   ÷
127   43  RCL
128   11   11
129   95   =
130   42  STO   λ' = sλ/n
131   36   36
132   65   ×
133   43  RCL
134   11   11
135   65   ×
136   53   (
137   43  RCL
138   02   02
139   65   ×
140   43  RCL
141   05   05
142   85   +
143   43  RCL
144   06   06
145   65   ×
146   43  RCL
147   09   09
148   54   )
149   95   =
150   94  +/-
151   22  INV
152   23  LNX
153   42  STO
154   35   35   Kb = e^-[sλ(r1 LAT + r2 D)]
```

```
155  00   0
156  42  STO
157  34   34
158  42  STO   SPARES: s_2 = 0
159  07   07
160  01   1
161  42  STO
162  08   08
163  42  STO   s_1 = K_2 = 1
164  03   03
165  85   +
166  43  RCL
167  36   36
168  65   ×
169  43  RCL
170  02   02
171  95   =
172  42  STO
173  04   04
174  42  STO   Δ = k_1 = 1 + λ't_1
175  38   38
176  06   6     0 is ind reg
177  42  STO   i = 2   for i. r_2 is
178  00   00    stored in reg. 6
```

$$\Delta_i = 1 + \frac{(\lambda' t_i)^{s_i+1}}{(s_i+1)!\ K_i}$$

```
179  01   1
180  85   +
181  53   (
182  43  RCL
183  36   36
184  65   ×
185  71  SBR
186  43  RCL
187  54   )
188  45  Y^X
189  53   (
190  71  SBR
191  43  RCL
192  85   +
193  01   1
194  54   )
195  42  STO
196  01   01
```

```
197  55   ÷
198  53   (     ┐
199  01   1     │
200  65   ×     │
201  43  RCL    │
202  01   01    │ computes s!
203  97  DSZ    │
204  01   01    │
205  02   02    │
206  00   00    │
207  54   )     ┘
208  55   ÷
209  71  SBR
210  43  RCL
211  95   =
212  32  X:T
```

```
213  43  RCL
214  38   38
215  22  INV   Δ < Δ_i ?
216  77   GE
217  02   02
218  30   30   no
219  32  X:T
220  42  STO   Δ = Δ_i
221  38   38
222  01   1
223  04   4
224  75   -
225  43  RCL
226  00   00
227  95   =
228  42  STO   i = 3-i
229  00   00
230  93   .     ← K*
231  09   9
232  05   5
233  55   ÷
234  43  RCL
235  35   35
236  95   =
237  23  LNX
238  55   ÷
239  32  X:T
240  42  STO   hold φ in R_1
241  01   01
242  23  LNX
243  71  SBR   h = ⌈ ln(K*/K) / ln φ ⌉^+
244  59  INT
245  32  X:T   hold h in t
```

```
246   73  RC*
247   00   00
248   65   ×
249   43  RCL    if nr_i > h
250   11   11    then end
251   95   =
252   77   GE    otherwise
253   02   02
254   79   79
255   44  SUM
256   34   34    SPARES = SPARES + nr_i
257   48  EXC
258   01   01
259   69  OP
260   30   30
261   64  PD*    K_i = φK_i
262   00   00
263   45  Y×
264   43  RCL
265   01   01
266   95   =
267   49  PRD    K = Kφ^(nr_i)
268   35   35
269   69  OP
270   30   30
271   01   1
272   74  SM*    S_i = S_i + 1
273   00   00
274   69  OP
275   30   30
276   61  GTO
277   01   01
278   79   79
279   32  X:T    final spares buy
280   44  SUM    SPARES = SPARES + h
281   34   34
```

$\bar{S} = (B/N + S)/n' + Q$

```
282   43  RCL
283   36   36
284   65   ×
285   43  RCL
286   13   13
287   65   ×
288   01   1     ← DRT/d
289   03   3
290   95   =
291   42  STO
292   01   01    μ = λ' N·DRT/d
293   34  √X
294   65   ×
295   01   1     ← Z_b
296   93   .
297   06   6
298   05   5
299   85   +
300   43  RCL
301   01   01
302   71  SBR
303   59  INT
304   65   ×
305   02   2     ← d
306   65   ×
307   43  RCL
308   09   09
309   65   ×
310   43  RCL
311   11   11
312   55   ÷
313   43  RCL    B = ⌈μ + Z_b√μ⌉·d·r_2·n/N
314   13   13
315   85   +
316   43  RCL
317   34   34
318   95   =
319   55   ÷
320   43  RCL
321   10   10
322   85   +
323   43  RCL
324   24   24
325   95   =
326   42  STO
327   34   34
```

$S' = \bar{\lambda}\left[1-(r_1+r_2)(1-\text{COND})\right]/h$

328	43	RCL	
329	37	37	
330	65	×	
331	43	RCL	
332	06	06	
333	65	×	
334	02	2	← h
335	95	=	
336	42	STO	$\bar{\lambda} = \lambda \cdot D \cdot h$
337	36	36	
338	55	÷	
339	43	RCL	
340	10	10	
341	65	×	
342	53	(	
343	01	1	
344	75	-	
345	53	(	
346	43	RCL	
347	05	05	
348	85	+	
349	43	RCL	
350	09	09	
351	54	)	
352	42	STO	
353	39	39	
354	65	×	
355	93	.	← 1-COND
356	09	9	
357	08	8	
358	54	)	
359	95	=	
360	42	STO	
361	07	07	S'

$UC = UC_e\left[\frac{N(S'+Z)}{e}\right]^{\log \text{RRATE}/\log 2}$

362	85	+	
363	43	RCL	
364	34	34	
365	95	=	
366	65	×	
367	43	RCL	
368	13	13	
369	55	÷	
370	43	RCL	
371	23	23	
372	95	=	
373	45	Y×	
374	93	.	← log RRATE / log 2
375	01	1	
376	05	5	
377	94	+/-	
378	65	×	
379	43	RCL	
380	22	22	
381	95	=	
382	42	STO	
383	08	08	

$L = \frac{(1+\rho)^{LC}-1}{\rho(1+\rho)^{LC}}$

384	93	.	← ρ
385	01	1	
386	00	0	
387	42	STO	
388	03	03	
389	85	+	
390	01	1	
391	95	=	
392	45	Y×	
393	43	RCL	
394	20	20	
395	87	IFF	if flag 0 is set then L=LC
396	00	00	
397	04	04	
398	16	16	
399	75	-	
400	01	1	
401	95	=	
402	55	÷	
403	43	RCL	
404	03	03	
405	55	÷	
406	53	(	
407	01	1	
408	85	+	
409	43	RCL	
410	03	03	
411	54	)	
412	45	Y×	
413	43	RCL	
414	20	20	
415	95	=	
416	42	STO	L
417	03	03	

$M'_m = [s\lambda \, MTTRS(1+\tilde{M} r_1) + Q \cdot SM] / (U \cdot WH_m)$

418	43	RCL	
419	12	12	
420	65	×	
421	43	RCL	
422	37	37	
423	65	×	
424	43	RCL	
425	26	26	
426	65	×	
427	53	(	
428	01	1	
429	85	+	
430	01	1	← $\tilde{M}$
431	93	.	
432	00	0	
433	65	×	
434	43	RCL	
435	05	05	
436	54	)	
437	85	+	
438	43	RCL	
439	24	24	
440	65	×	
441	43	RCL	
442	27	27	
443	95	=	
444	55	÷	
445	05	5	← $U \cdot WH_m$
446	03	3	
447	95	=	
448	42	STO	
449	38	38	

$TC_m = TS(1+\tilde{T} r_1 n)$

450	43	RCL	
451	28	28	
452	65	×	
453	53	(	
454	01	1	
455	85	+	
456	01	1	← $\tilde{T}$
457	93	.	
458	00	0	
459	65	×	
460	43	RCL	
461	05	05	
462	65	×	
463	43	RCL	
464	11	11	
465	54	)	
466	95	=	
467	42	STO	
468	16	16	

initialize

469	01	1	
470	04	4	
471	42	STO	indirect register for
472	00	00	o and m subscripted
473	00	0	variables
474	42	STO	
475	37	37	LCC → 0
476	42	STO	indirect register for
477	01	01	print / no print flag
478	58	FIX	
479	02	02	

manpower costs

480	71	SBR	maintenance wage
481	00	00	and training
482	36	36	
483	43	RCL	
484	24	24	
485	65	×	
486	43	RCL	
487	29	29	
488	65	×	
489	43	RCL	
490	25	25	
491	55	÷	
492	07	7	← WH_o
493	04	4	
494	65	×	
495	43	RCL	
496	12	12	
497	95	=	
498	42	STO	
499	38	38	$M'_o = s \cdot \theta \cdot MHR \cdot Q / WH_o$
500	71	SBR	
501	00	00	
502	36	36	

$C_5 = (\bar{S} + S'L) \cdot UC$

503	43	RCL	
504	34	34	
505	85	+	
506	43	RCL	
507	07	07	
508	65	×	
509	43	RCL	
510	03	03	
511	95	=	
512	65	×	
513	43	RCL	
514	08	08	
515	71	SBR	
516	00	00	
517	21	21	Hardware

$C_6 = STE(1+SOm(r_1)\tilde{S})(1+mL)\cdot N$

518	43	RCL	
519	30	30	
520	65	×	
521	53	(	
522	01	1	
523	85	+	
524	43	RCL	
525	05	05	
526	69	OP	
527	10	10	
528	65	×	
529	01	1	← $\tilde{S}$
530	93	.	
531	00	0	
532	54	)	
533	65	×	
534	53	(	
535	01	1	
536	85	+	
537	93	.	← m
538	01	1	
539	02	2	
540	65	×	
541	43	RCL	
542	03	03	
543	54	)	
544	71	SBR	support and
545	00	00	test equipment
546	21	21	

$C_7 = \tilde{\lambda} L(r_2 COD + r_1 RP)\cdot N$

547	43	RCL	
548	36	36	
549	65	×	
550	43	RCL	
551	03	03	
552	65	×	
553	53	(	
554	43	RCL	
555	09	09	
556	65	×	
557	43	RCL	
558	31	31	
559	85	+	
560	43	RCL	
561	05	05	
562	65	×	
563	43	RCL	
564	32	32	
565	54	)	
566	71	SBR	
567	00	00	repair
568	21	21	

$C_8 = IECn + IMCnL(1+PP(r_1+r_2))$

569	93	.	← IEC
570	04	4	
571	05	5	
572	85	+	
573	93	.	
574	02	2	← IMC
575	03	3	
576	65	×	
577	43	RCL	
578	03	03	
579	65	×	
580	53	(	
581	01	1	
582	85	+	
583	01	1	← PP
584	93	.	
585	00	0	
586	65	×	
587	43	RCL	
588	39	39	
589	54	)	
590	95	=	
591	65	×	
592	43	RCL	
593	11	11	
594	71	SBR	item entry and
595	00	00	management
596	24	24	

$C_9 = BOC(1+r_1 n\tilde{\delta})$

597	69	OP	
598	31	31	
599	43	RCL	
600	33	33	
601	65	×	
602	53	(	
603	01	1	
604	85	+	
605	43	RCL	
606	05	05	
607	65	×	
608	43	RCL	
609	11	11	
610	65	×	
611	01	1	← $\tilde{\delta}$
612	93	.	
613	00	0	
614	54	)	
615	71	SBR	
616	00	00	Documentation
617	24	24	

Print System Life Cycle Cost

618	98	ADV
619	43	RCL
620	37	37
621	99	PRT
622	98	ADV
623	22	INV
624	58	FIX
625	92	RTN
626	00	0
627	00	0
628	00	0
629	00	0
630	00	0
631	00	0
632	00	0
633	00	0
634	00	0
635	00	0
636	00	0
637	00	0
638	00	0
639	00	0

APPENDIX B:

LOWEST REMOVABLE ASSEMBLY

MODEL PROGRAM LISTING

PRINT and SUM to LCC

```
000  65  ×
001  43  RCL
002  37   37
003  95   =     LCC = LCC + N·c_j
004  44  SUM    if j ≤ 6 (enter at 000)
005  02   02
006  72  ST*    LCC = LCC + c_j
007  01   01    if j ≥ 7 (enter at 003)
008  87  IFF
009  40  IND    store c_j in R_j j≥2
010  01   01
011  00   00    if flag j is set
012  15   15    do not print c_j
013  69  OP
014  06   06
015  69  OP
016  20   20
017  69  OP
018  21   21
019  73  RC*
020  00   00    load op. code
021  69  OP     for next lable
022  04   04
023  92  RTN
```

x = y·s·λ

```
024  65   ×
025  43  RCL    used
026  32   32    three
027  65   ×     times
028  43  RCL
029  29   29
030  95   =
031  42  STO
032  09   09
033  92  RTN
```

K(t)

```
034  29  CP
035  71  SBR
036  00   00
037  24   24    x = s·λ·t
038  75   -
039  43  RCL
040  04   04
041  67   EQ    if S=0
042  00   00    then K = e^-x
043  57   57    otherwise ↓
044  95   =
045  55   ÷
046  43  RCL
047  09   09
048  34  √X
049  95   =
050  36  PGM
051  14   14
052  11   A
053  36  PGM
054  14   14    K = Q((x - S)/√x)
055  12   B
056  92  RTN
```

if S=0

```
057  95   =
058  43  RCL
059  09   09
060  94  +/-
061  22  INV    K = e^-x
062  23  LNX
063  92  RTN
```

S(t,K)

```
064  71  SBR
065  00   00
066  24   24    x = s·λ·t
067  94  +/-
068  22  INV
069  23  LNX
070  32  X:T
071  43  RCL
072  07   07
073  22  INV    if e^-x ≥ K then
074  77   GE    S=0, otherwise ↓
075  01   01
076  37   37
```

077	75	-	
078	01	1	
079	95	=	
080	33	X²	
081	35	1/X	
082	23	LNX	
083	34	ΓX	
084	42	STO	
085	02	02	$h = \sqrt{\log\left(\frac{1}{1-x}\right)^2}$
086	75	-	
087	53	(	
088	02	2	
089	93	.	
090	05	5	
091	85	+	
092	93	.	
093	08	8	
094	65	×	
095	43	RCL	
096	02	02	
097	54	)	
098	55	÷	
099	53	(	
100	01	1	
101	85	+	
102	01	1	
103	93	.	
104	04	4	
105	65	×	
106	43	RCL	
107	02	02	
108	85	+	
109	93	.	
110	01	1	
111	09	9	
112	65	×	
113	43	RCL	
114	02	02	
115	33	X²	
116	54	)	
117	95	=	$Z = h - \frac{2.5+.8h}{1+1.4h+.19h^2}$
118	65	×	
119	43	RCL	
120	09	09	
121	34	ΓX	
122	85	+	
123	43	RCL	
124	09	09	
125	95	=	$h = x + Z\sqrt{x}$

$\lceil x \rceil$

126	32	X:T	
127	00	0	
128	77	GE	
129	01	01	
130	36	36	
131	32	X:T	$\lceil x \rceil = \begin{cases} 0 & x \le 0 \\ \text{INT}(x)+1 & x > 0 \end{cases}$
132	85	+	
133	01	1	
134	95	=	
135	59	INT	
136	92	RTN	
137	00	0	
138	92	RTN	

end of S(t,K)

139	76	LBL	BEGIN EXECUTION
140	11	A	
141	43	RCL	
142	30	30	
143	65	×	
144	43	RCL	
145	12	12	
146	65	×	
147	43	RCL	
148	13	13	
149	65	×	
150	43	RCL	
151	31	31	
152	55	÷	
153	43	RCL	
154	14	14	
155	95	=	
156	42	STO	$\lambda = g \cdot Q \cdot \delta \cdot AHR / MTBF$
157	29	29	
158	43	RCL	
159	35	35	
160	42	STO	$K = K_i$
161	07	07	
162	43	RCL	
163	15	15	
164	65	×	
165	43	RCL	
166	33	33	
167	85	+	
168	43	RCL	
169	16	16	
170	65	×	
171	43	RCL	
172	34	34	
173	95	=	
174	42	STO	$t = r_1 LRT + r_2 D$
175	28	28	

```
176   71  SBR
177   00   00   S= S(t,Ki)
178   64   64
179   42  STO
180   04   04
181   43  RCL
182   28   28
183   71  SBR
184   00   00
185   34   34
186   42  STO
187   28   28   Save K(t)

188   43  RCL   if r2=0
189   16   16   then B=0
190   67   EQ   otherwise ↓
191   02   02
192   32   32

193   43  RCL
194   36   36
195   71  SBR
196   00   00   K(XD)
197   34   34
198   42  STO
199   10   10
200   75   -
201   43  RCL
202   35   35
203   95   =
204   55   ÷
205   53   (
206   43  RCL
207   10   10
208   75   -
209   43  RCL
210   28   28
211   54   )
212   95   =
213   42  STO
214   07   07   b = (Ki - K(XD)) / (K(D) - K(XD))

215   43  RCL
216   37   37
217   65   ×
218   43  RCL
219   38   38
220   55   ÷
221   02   2    ← d
222   71  SBR
223   00   00
224   64   64   B= S(N·DRT/d, b)
```

```
225   44  SUM   Save B as B.K(t)
226   28   28
227   65   ×
228   02   2    ← d
229   55   ÷
230   43  RCL
231   37   37   B̄ = B·d/N

232   85   +
233   43  RCL
234   30   30
235   65   ×
236   43  RCL
237   12   12
238   95   =
239   44  SUM
240   04   04   S = S + B̄ + q·Q
```

Replenishment Spares

```
241   43  RCL
242   29   29
243   65   ×
244   43  RCL
245   34   34
246   65   ×
247   02   2    ← h
248   95   =
249   42  STO
250   06   06   λ̄ = λ·D·h
251   65   ×
252   53   (
253   01   1
254   75   -
255   53   (
256   43  RCL
257   15   15
258   85   +
259   43  RCL
260   16   16
261   54   )
262   42  STO   r̄ = r1 + r2
263   07   07
264   65   ×
265   43  RCL   ← 1-COND
266   40   40
267   54   )
268   95   =
269   42  STO   s' = λ̄(1 - r̄(1-COND))
270   05   05
```

$UC = UC_e \left[\frac{N(S+S')}{e} \right]^{\log RRATE/\log 2}$

```
271  85  +
272  43  RCL
273  04   04
274  95  =
275  65  ×
276  43  RCL
277  37   37
278  55  ÷
279  43  RCL
280  39   39
281  95  =
282  45  YX
283  43  RCL   ← log RRATE/log 2
284  41   41
285  65  ×
286  43  RCL
287  17   17
288  95  =
289  42  STO
290  59   59
```

Initialize for printing

```
291  58  FIX
292  02   02
293  02  2
294  42  STO   indirect register
295  01   01   for Cj
296  05  5
297  01  1
298  42  STO   indirect register
299  00   00   for lables
300  71  SBR
301  00   00   place op code for
302  19   19   C2 in print register
```

$M'_m = s\lambda(MTRR + r_1\ MTTR\ )/u \cdot WH_m$

```
303  43  RCL
304  18   18
305  85  +
306  43  RCL
307  15   15
308  65  ×
309  43  RCL
310  19   19
311  95  =
312  55  ÷
313  05  5
314  03  3     ← u·WHm
315  71  SBR
316  00   00
317  24   24
```

$A(M') = \lceil M' - \min(M', AN) \rceil$

```
318  32  X:T
319  43  RCL
320  09   09
321  75  -
322  43  RCL
323  44   44
324  22  INV
325  77   GE
326  03   03
327  29   29
328  32  X:T
329  71  SBR
330  01   01
331  25   25
332  42  STO
333  03   03
```

$C_2 = (M'/s \cdot BG + A(M')(BN - BG))L \cdot N$

```
334  65  ×
335  53  (
336  43  RCL
337  45   45
338  75  -
339  43  RCL
340  46   46
341  54  )
342  85  +
343  43  RCL
344  09   09
345  55  ÷
346  43  RCL
347  32   32
348  65  ×
349  43  RCL
350  46   46
351  95  =
352  65  ×
353  43  RCL
354  47   47
355  71  SBR
356  00   00   print wage
357  00   00
```

$C_3 = (\lceil m' \rceil (TFI + r_1 TR) + A(m')TA)(1+TOR \cdot L) \cdot N$

```
358  43  RCL
359  09   09
360  71  SBR
361  01   01 ← ⌈m'⌉
362  26   26
363  65   ×
364  53   (
365  43  RCL
366  20   20
367  85   +
368  43  RCL
369  15   15
370  65   ×
371  43  RCL
372  21   21
373  54   )
374  85   +
375  43  RCL
376  03   03
377  65   ×
378  43  RCL
379  48   48
380  95   =
381  65   ×
382  43  RCL ← 1+TOR·L
383  49   49
384  71  SBR
385  00   00 Print Training
386  00   00
```

$C_4 = (S + S' \cdot L) \cdot UC \cdot N$

```
387  43  RCL
388  04   04
389  85   +
390  43  RCL
391  05   05
392  65   ×
393  43  RCL
394  47   47
395  95   =
396  65   ×
397  43  RCL
398  59   59
399  71  SBR
400  00   00 Print Hardware
401  00   00
```

$C_5 = (STE_{sys} + r_1 STE_{rpr})(1 + mL) \cdot N$

```
402  43  RCL
403  22   22
404  85   +
405  43  RCL
406  15   15
407  65   ×
408  43  RCL
409  23   23
410  95   =
411  65   ×
412  43  RCL
413  50   50 ← 1+mL
414  71  SBR
415  00   00 Print Support and
416  00   00 Test Equipment
```

$C_6 = \bar{\lambda} \cdot L (r_1 UC/c + r_2 COD)$

```
417  43  RCL
418  15   15
419  65   ×
420  43  RCL
421  59   59
422  55   ÷
423  43  RCL
424  24   24
425  85   +
426  43  RCL
427  16   16
428  65   ×
429  43  RCL
430  43   43
431  95   =
432  65   ×
433  43  RCL
434  06   06
435  65   ×
436  43  RCL
437  47   47
438  71  SBR
439  00   00 Print Repair
440  00   00
```

$C_7 = (1 + F Z)(IEC + IMC \cdot L)$

```
441   01   1
442   85   +
443   43  RCL
444   07   07
445   65   ×
446   43  RCL
447   25   25
448   95   =
449   65   ×
450   43  RCL
451   42   42   ← IEC + IMC·L
452   71  SBR
453   00   00   Print Item Entry
454   03   03   and Management
```

$C_8 = P_f + r_1 P_r$

```
455   43  RCL
456   26   26
457   85   +
458   43  RCL
459   15   15
460   65   ×
461   43  RCL
462   27   27
463   95   =
464   71  SBR
465   00   00  Print Documentation
466   03   03
```

Print Life-Cycle Cost

```
467   98  ADV
468   43  RCL
469   02   02
470   69  OP
471   06   06
```

```
472   00   0
473   42  STO   set TC_0 = 0
474   03   03
475   22  INV
476   58  FIX
477   92  RTN
```

APPENDIX C:

SYSTEM AGGREGATION MODEL

PROGRAM LISTING

PRINT, STORE, SUM to LCC

```
000  76  LBL
001  99  PRT
002  85   +
003  73  RC*
004  00   00     Σ_i C_ij
005  65   ×
006  43  RCL
007  44   44
008  95   =
009  72  ST*     output
010  02   02     register
011  69  OP
012  20   20
013  69  OP
014  22   22
015  87  IFF     if flag j is
016  40  IND     set do not
017  03   03     print C_j
018  00   00
019  22   22
020  69  OP
021  06   06
022  44  SUM
023  40   40     sum to LCC
024  73  RC*
025  01   01
026  69  OP      Store op code
027  04   04     for next lable
028  69  OP      in print reg.
029  21   21
030  69  OP
031  23   23
032  92  RTN
```

PERSONNEL

```
033  73  RC*
034  00   00
035  69  OP
036  20   20
037  42  STO
038  45   45     M'
039  32  X:T
040  43  RCL
041  45   45
042  75   -
043  73  RC*
044  00   00
045  69  OP
046  20   20
047  22  INV
048  77   GE
049  00   00
050  52   52
051  32  X:T
052  95   =
053  32  X:T
054  00   0
055  77   GE
056  00   00
057  63   63
058  32  X:T
059  59  INT
060  85   +
061  01   1
062  95   =
063  42  STO
064  46   46     A(m') = ⌈m' - min(m', AN)⌉
```

Wage = (m'/s·BG + A(m')(BN-BG))·L·N

```
065  65   ×
066  53   (
067  73  RC*
068  00   00
069  69  OP
070  20   20
071  75   -
072  43  RCL
073  07   07
074  54   )
075  85   +
076  43  RCL
077  45   45
078  55   ÷
079  43  RCL
080  05   05
081  65   ×
082  43  RCL
083  07   07
084  95   =
085  65   ×
086  43  RCL
087  09   09
088  65   ×
089  43  RCL
090  08   08
091  95   =
092  71  SBR
093  00   00   print Wage
094  15   15
```

Trn = (A(m')TA + [m']TC)(1+TDR·L)·N

```
095  43  RCL
096  46   46
097  65   ×
098  73  RC*
099  00   00
100  69  OP
101  20   20
102  85   +
103  53   (
104  43  RCL
105  45   45
106  32  X:T
107  43  RCL
108  45   45
109  59  INT
110  67   EQ
111  01   01
112  15   15
113  85   +
114  01   1
115  54   )
116  65   ×
117  73  RC*
118  00   00
119  69  OP
120  20   20
121  95   =
122  65   ×
123  43  RCL
124  29   29
125  65   ×
126  43  RCL
127  08   08
128  95   =
129  71  SBR
130  00   00   print Training
131  15   15
132  92  RTN
```

end of subroutine PERSONNEL

AGGREGATION

```
133  76 LBL
134  12  B
135  94 +/-   <- -QIPA_i
136  76 LBL
137  11  A
138  42 STO
139  45  45   <- QIPA_i
140  44 SUM
141  41  41
142  55  ÷
143  43 RCL
144  14  14
145  95  =
146  44 SUM   Σ QIPA_i / MTBF_i
147  42  42
148  43 RCL
149  18  18
150  85  +
151  43 RCL
152  15  15
153  65  ×
154  43 RCL
155  19  19
156  95  =
157  65  ×
158  43 RCL
159  45  45
160  95  =
161  44 SUM   Σ MTRR_i + r_{s,i} MTTR_i
162  43  43
163  43 RCL
164  20  20
165  85  +
166  43 RCL
167  15  15
168  65  ×
169  43 RCL
170  21  21
171  95  =
172  42 STO   TC_m = TS_i + r_{s,i} TC_{m,i}
173  02  02
174  43 RCL
175  45  45
176  55  ÷
177  43 RCL
178  12  12
179  95  =
180  42 STO   R_i = QIPA_i / q_i
181  46  46
```

initialize indirect registers

```
182  09  9
183  42 STO   C_{j,i} ind reg
184  00  00
185  03  3
186  09  9
187  42 STO   Σ_i C_{i,j} ind reg
188  01  01
```

sum in accumulation registers

```
189  73 RC*
190  00  00
191  65  ×    for j = 9-1
192  43 RCL
193  46  46   C_j = C_j + R_i C_{i,j}
194  95  =
195  74 SM*
196  01  01
197  69 OP
198  31  31
199  97 DSZ
200  00  00
201  01  01
202  89  89
```

```
203  43 RCL
204  11  11   print LRA
205  69 OP    identifier
206  04  04
207  43 RCL
208  45  45
209  69 OP
210  06  06
211  92 RTN
```

end of AGGREGATION

SYSTEM COSTS

212	76	LBL	
213	13	C	
214	58	FIX	
215	02	02	
216	43	RCL	
217	12	12	
218	65	×	
219	43	RCL	
220	06	06	
221	95	=	
222	42	STO	
223	44	44	$n = g \cdot Q$
224	65	×	
225	53	(	
226	43	RCL	
227	13	13	
228	55	÷	
229	05	5	
230	03	3	← $U \cdot WH_m$
231	85	+	
232	43	RCL	
233	39	39	
234	54	)	
235	95	=	
236	42	STO	
237	13	13	$m'_m = n \; SM / U \cdot WH_m$
238	43	RCL	
239	32	32	$+ \sum_i m'_{m(i)}$
240	65	×	
241	43	RCL	
242	44	44	
243	95	=	
244	44	SUM	
245	17	17	$TC_m = TS + n \sum_i TC_{m(i)}$
246	43	RCL	
247	05	05	
248	65	×	
249	43	RCL	
250	10	10	
251	55	÷	
252	07	7	
253	04	4	← WH_o
254	65	×	
255	43	RCL	
256	44	44	
257	95	=	
258	49	PRD	
259	18	18	$m'_o = s \cdot AHR \cdot n \cdot \theta / WH_o$
260	43	RCL	
261	33	33	
262	65	×	
263	43	RCL	
264	44	44	
265	95	=	
266	44	SUM	
267	22	22	$TC_o = TS_o + n \sum TC_o$

initialize indirect registers

268	00	0	
269	42	STO	
270	40	40	LCC
271	42	STO	ind. reg. for print/no print flags
272	03	03	
273	01	1	
274	03	3	
275	42	STO	ind. reg. for o and
276	00	00	m subscripted variables
277	04	4	
278	07	7	
279	42	STO	ind. reg. for lables
280	01	01	
281	71	SBR	store op. code for
282	00	00	first lable in print register
283	24	24	

C_1 = Maintenance Wage
C_2 = Maintenance Training

284	71	SBR	
285	00	00	call subroutine PERSONNEL
286	33	33	

C_3 = Operator Wage
C_4 = Operator Training

287	71	SBR	
288	00	00	call subroutine PERSONNEL
289	33	33	

initial initalize registers

290	04	4	
291	42	STO	indirect reg. for
292	02	02	output costs
293	03	3	
294	04	4	
295	42	STO	ind. reg. for
296	00	00	$\sum_i C_{i,j}$

$C_5 = N \cdot n \cdot PT_\ell \left(\frac{N \cdot n}{\ell} \right)^{\log RRATE/\log 2} + n \sum_i HRDW_i$

297	43	RCL	
298	08	08	
299	65	×	
300	43	RCL	
301	44	44	
302	55	÷	
303	43	RCL	
304	24	24	
305	95	=	
306	45	Y^X	
307	43	RCL	
308	25	25	← log RRATE/log 2
309	65	×	
310	43	RCL	
311	23	23	
312	65	×	
313	43	RCL	
314	08	08	
315	65	×	
316	43	RCL	
317	44	44	
318	71	SBR	
319	99	PRT	Print Hardware

$C_6 = N \cdot STE \cdot (1 + mL) + n \sum_i STE_i$

320	43	RCL	
321	08	08	
322	65	×	
323	43	RCL	
324	27	27	
325	65	×	
326	43	RCL	
327	28	28	
328	71	SBR	Print Support &
329	99	PRT	Test Equipment

$C_7 = n \sum_i RPR_i$

330	71	SBR	
331	00	00	
332	03	03	Print Repair

$C_8 = n \sum_i IEMC_i$

333	71	SBR	
334	00	00	Print Item Entry
335	03	03	and Management

$C_9 = DOC_{sys} + n \sum_i DOC_i$

336	69	OP	
337	33	33	Flag 8 will suppress
338	43	RCL	printing C_8 and C_9
339	26	26	
340	71	SBR	
341	99	PRT	Print Documentation

Make output compatible with Aggregation input

342	43	RCL	
343	13	13	
344	42	STO	
345	09	09	STO Ni_m
346	43	RCL	
347	17	17	
348	42	STO	
349	20	20	STO TC_m
350	43	RCL	
351	22	22	
352	42	STO	
353	03	03	STO TC_0
354	00	0	
355	42	STO	
356	15	15	set $r_i = 0$
357	43	RCL	
358	43	43	
359	55	÷	
360	43	RCL	
361	41	41	
362	95	=	
363	42	STO	$MTTRS = \frac{\sum MTTR_i}{\sum_i QIPA_i}$
364	18	18	
365	43	RCL	
366	42	42	
367	35	1/X	
368	42	STO	$MTBF = \left(\sum_i MTBF_i \right)^{-1}$
369	14	14	

Print Life Cycle Cost

```
370  98 ADV
371  43 RCL
372  40  40
373  71 SBR
374  00  00
375  20  20
376  22 INV
377  58 FIX
378  92 RTN
```

Print MTBF and MTTR

```
379  76 LBL
380  14  D
381  58 FIX
382  02  02
383  98 ADV
384  43 RCL
385  14  14
386  71 SBR  ← MTBF
387  00  00
388  20  20
389  43 RCL
390  18  18
391  71 SBR  ← MTTR
392  00  00
393  20  20
394  22 INV
395  58 FIX
396  92 RTN
```

APPENDIX D:

SYSTEM CONFIDENCE LEVEL

PROGRAM LISTING

```
069   76  LBL
070   13   C
071   00   0      <- divide out
072   75   -
073   76  LBL
074   12   B
075   01   1      <- prod K
076   95   =
077   42  STO
078   42   42

079   29  CP
080   43  RCL
081   15   15     if r_1 = 1
082   22  INV     (local repair)
083   67   EQ     K already
084   01   01     computed B=0
085   29   29     K(t) = K(LRT)
```

compute confidence level at depot

```
086   43  RCL
087   41   41
088   65   ×
089   43  RCL
090   29   29
091   95   =
092   42  STO
093   00   00     x = λy
094   75   -
095   43  RCL
096   28   28
097   59  INT
098   67   EQ     if B=0
099   01   01     b = e^-x
100   59   59
101   95   =
102   55   ÷
103   43  RCL
104   00   00
105   34  ΓX
106   95   =
107   36  PGM
108   14   14
109   11   A
110   36  PGM
111   14   14
112   12   B      b = Q((x-B)/√x)
```

compute achieved confidence level

```
113   65   ×
114   53   (
115   43  RCL
116   28   28
117   22  INV
118   59  INT
119   75   -
120   43  RCL
121   10   10
122   54   )
123   85   +
124   43  RCL
125   10   10
126   95   =
127   42  STO
128   28   28     K = b(K(D) - K(XD)) + K(XD)
```

initialize printing register

```
129   43  RCL
130   11   11
131   69  OP
132   04   04
133   58  FIX
134   02   02

135   43  RCL
136   28   28
137   45  Y^X
138   43  RCL
139   42   42
140   95   =
141   49  PRD
142   40   40     if R_42 = -1 then
143   45  Y^X     divide out of K
144   43  RCL     and print -K×100
145   42   42
146   65   ×
147   43  RCL     else prod K
148   42   42     and print K×100
149   65   ×
150   01   1
151   00   0
152   00   0
153   95   =
154   69  OP
155   06   06
156   22  INV
157   58  FIX
158   92  RTN
```

```
000  76 LBL    ; BEGIN EXECUTION
001  11  A
002  43 RCL
003  32  32
004  65  ×
005  43 RCL
006  37  37
007  65  ×
008  43 RCL
009  38  38
010  55  ÷
011  02  2     ; ← d
012  95  =
013  42 STO    ; y= s.N.DRT/d
014  41  41
015  01  1
016  42 STO    ; K=1
017  40  40
```

Print Initialization Heading

```
018  01  1
019  05  5
020  03  3
021  02  2
022  03  3
023  01  1
024  02  2
025  01  1
026  02  2
027  04  4
028  69 OP     ; "CONFI"
029  01  01
030  01  1
031  06  6
032  01  1
033  07  7
034  03  3
035  01  1
036  01  1
037  05  5
038  01  1
039  07  7
040  69 OP     ; "DENCE"
041  02  02
042  02  2
043  07  7
044  01  1
045  07  7
046  04  4
047  02  2
048  01  1
049  07  7
050  69 OP     ; ", LEVE"
051  03  03
052  02  2
053  07  7
054  00  0
055  00  0
056  05  5
057  05  5
058  06  6
059  01  1
060  05  5
061  06  6
062  69 OP     ; "L_v (%)"
063  04  04
064  98 ADV
065  69 OP
066  05  05
067  98 ADV
068  92 RTN
```

if B=0

```
159  95  =
160  43  RCL
161  00   00
162  94  +/-
163  22  INV
164  23  LNX   b=e^-x
165  61  GTD
166  01   01
167  13   13
```

Print System Confidence Level

```
168  76  LBL
169  14   D
170  69  DP
171  00   00
172  03   3
173  06   6
174  04   4
175  05   5
176  03   3
177  06   6
178  69  DP
179  02   02   "vvSYS"
180  03   3
181  07   7
182  01   1
183  07   7
184  03   3
185  00   0
186  00   0
187  00   0
188  00   0
189  00   0
190  69  DP    "TEMvv"
191  03   03
192  98  ADV
193  69  DP
194  05   05
195  58  FIX
196  02   02
197  43  RCL
198  40   40
199  65   ×
200  01   1
201  00   0
202  00   0
203  95   =
204  99  PRT   Print K×100
205  98  ADV
206  22  INV
207  58  FIX
208  92  RTN
```