Appendix I

I. H. D. Research - A guide to Later Students.

The I. H. D. student should feel privileged amongst students. He is tackling a real problem, in industry, while still retaining the academic 'roots' of his traditional research - student contemporaries. The position, however, is not wholly 'clear-cut' and 'rosy' as it would appear.

The I. H. D. student is faced with many problems not experienced by the traditional post-graduate student. The I. H. D. student (if I am a typical 'sample') feels that he is in a state of limbo. He is neither a conventional employee of the firm nor is he 100 per cent a student. This results in a sense of lost direction. The interests of the firm are not always consistent with those of his academic supervisors, or indeed, the student's own conscience. The student must decide for himself what is his objective. Is it; to solve the firms problem, perhaps at the expense of his thesis or even his degree? Or is his target to produce an academic thesis which may or may not happen to coincide with what the firm see as being their 'real' problem? I submit that the optimum research path often lies between these two extremes, a practical solution to the problem, but approached and analysed in such a way as to satisfy the academic requirements of post-graduate research.

The I. H. D. student also faces a more practical stumbling block.

He is investigating an industrial problem and industry is staffed by people. There is thus a problem of communication. The researcher may ask a question and the employee will often give an 'idealized' answer, since he thinks this is the answer he ought to give. The employee often feels a sense of being threatened by the student and his research. His answers to the student's questions reflect this in that the employee gives too little or inaccurate (i.e. idealized) information. The student must choose his phraseology carefully, trying to reassure the employee that his job is not in danger and that the student's research is merely to provide another 'tool' for the employee to ease his own work load, in much the same way that, say, a slide-rule would.

The point has already been made that in an industrial environment projects are often evaluated using criteria different from those used by the research worker in his analytical studies. Management often adopts criteria which the research worker might consider irrelevant, the situation being influenced by company policy and an individual's intrinsic prejudices. The student must recognise this ambivalence and respect it, even though it may conflict with his own conclusions, reached by careful consideration of scientific fact.

Appendix II

FORTRAN programme for calculating design variables for axisymmetric forgings. [30]

The FORTRAN programme developed at the Battelle Institute for calculating various design variables; flash-weight, flash gap ratio, net weight etc., for axisymmetric forgings is considered below.

The programme in its original form required data for each corner/fillet of a half-section of the forging to be input on punched cards. The authors considered the forging shown in figure A1 and included a listing of the corresponding data cards required, table A1 (a).

A specimen output from this programme, for the forging shown, is reproduced in table A2.

Definition Of Input Parameters.

a - Number of corners/fillets above parting-line.

b - Number of corners/fillets below parting-line.

k - Equals; +1 if the volume of this element is to be added to the total, -1 otherwise.

Rj - Radial distance of corner/fillet from center-line

Rj = zero if this dimension is not available from the drawing.

Hcj - Axial distance between two consecutive corners.

rj - Fillet or corner radius.

α_j - Angle between j th. and (j+1) th. corner.

f - Equals; zero if corner radial distance is obtained from the engineering drawing ; +1 if this (unknown) dimension is to be calculated from the (j-1) th. dimension; -1 if to be calculated from the (j+1) th. dimension.

p - Density of material.

D_o - Diameter of initial round stock.

D_c - Maximum diameter of forging.

h_o - Minimum distance between flat surfaces upon which the stock was resting when the dies were completely closed.

H_a - Distance between internal and external parting lines.

H_s - Final height of the forging.

The logic of the programme is admirably described by the authors[30], the approach being essentially one of breaking the forging down into concentric cylinders to be added to or subtracted from the total volume.

By modifying the input routine, the present author was able to significantly reduce the amount of numerical data necessary for the analysis of any axisymmetric forging. The modifications were such that decision procedures were included in the logic, enabling the programme itself to choose between alternate courses of action instead of requiring 'flags' to indicate this, as was the case with the original programme.

The modified form of data input for the forging shown in

figure A1 is given in table A1(b). The increase in computational time due to this modification is negligible.

The values in column 'f', indicating from which corner dimension an unknown corner dimension is to be calculated, are no longer required. The sign of α_j, positive or negative, is also not necessary in the modified version. The advantage here is not in the small saving in punching/documenting required but rather in the fact that the user does not himself have to decide if the value should be positive or negative, a fairly involved decision.

The values; D_c, h_o, h_a, H_s, require similar involved decisions on the part of the user; the modified programme generates these values internally.

The overall reduction in data input required is of the order of 20%, from 102 to 82 items. A listing of the modified FORTRAN programme is given.

INPUT DATA FOR THE EXAMPLE FORGING
(EACH LINE CORRESPONDS TO ONE IBM CARD)

Computer Card Column Number					
10	20	30	40	50	60
7	9				
K	Ri	Hci	ri	αi	f
1	0.0000	0.0000	0.0000	-90.0000	0.0000
1	0.7870	0.0000	0.2360	-6.0000	0.0000
1	0.0000	0.3940	0.1160	-90.0000	1.0000
1	1.3790	0.0000	0.3940	-6.0000	0.0000
1	0.0000	0.5900	0.1180	-90.0000	1.0000
1	0.0000	0.0000	0.3940	-6.0000	-1.0000
1	2.7570	0.3940	0.0000	0.0000	0.0000
1	2.7570	0.0000	0.0000	6.0000	0.0000
1	0.0000	0.1970	0.1575	90.0000	1.0000
1	2.1650	0.0000	0.1970	5.7000	0.0000
1	0.0000	1.5750	0.1970	90.0000	1.0000
-1	0.0000	0.0000	0.1970	9.0000	-1.0000
-1	1.1820	0.9850	0.1575	90.0000	0.0000
-1	0.0000	0.0000	0.1970	9.0000	-1.0000
-1	0.5910	0.7870	0.1970	90.0000	0.0000
-1	0.0000	0.0000	0.0000	90.0000	0.0000
0.2830	5.5140	5.5140	0.3935	0.1967	3.1450
p	Do	D_C	h_O	hA	H_S

Table A1 (a).

INPUT DATA FOR THE EXAMPLE FORGING
(EACH LINE CORRESPONDS TO ONE IBM CARD)

Computer Card Column Number					
10	20	30	40	50	60
7	9				
K	Ri	Hci	ri	αi	
1	0.0000	0.0000	0.0000	90.0000	
1	0.7870	0.0000	0.2360	6.0000	
1	0.0000	0.3940	0.1180	90.0000	
1	1.3790	0.0000	0.3940	6.0000	
1	0.0000	0.5900	0.1180	90.0000	
1	0.0000	0.0000	0.3940	6.0000	
1	2.7570	0.3940	0.0000	0.0000	
1	2.7570	0.0000	0.0000	6.0000	
1	0.0000	0.1970	0.1575	90.0000	
1	2.1650	0.0000	0.1970	5.7000	
1	0.0000	1.5750	0.1970	90.0000	
-1	0.0000	0.0000	0.1970	9.0000	
-1	1.1820	0.9850	0.1575	90.0000	
-1	0.0000	0.0000	0.1970	9.0000	
-1	0.5910	0.7870	0.1970	90.0000	
-1	0.0000	0.0000	0.0000	90.0000	
0.2830	5.5140				
p	D_O				

Table A1 (b).

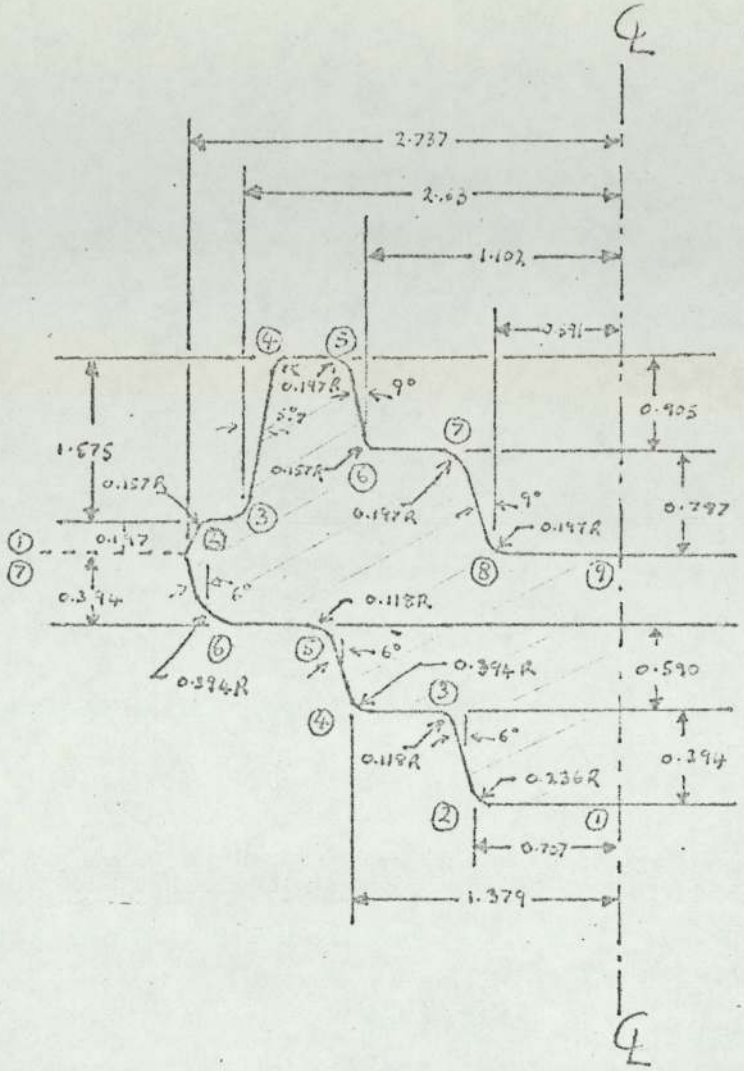


Figure A1. Axisymmetric Forging. (After Altan et al.[30])

PERIMETER = 18.08550 SURFACE = 8.47459

VOLUME = 33.25583 RADIUS OF C OF G

= 1.24910

SHAPE DIFFICULTY FACTOR IS 2.02221

FORGING WEIGHT WITHOUT FLASH IS 9.41140

FLASH THICKNESS, FLASH WIDTH, FLASH RATIO ARE

0.11403 0.46062 4.03952

FORGING WEIGHT, FLASH WEIGHT, TOTAL WEIGHT ARE

9.41140 1.04123 10.45263

THE PROJECTED AREA INCLUDING FLASH IS 32.52509

Table A2. Output from computer programme.

```
      PROGRAM(SHAPE)
      INPUT 1=CR0
      OUTPUT 2=LP0
      TRACE 2
      END
      MASTER
C     UNIVERSITY OF ASTON FORGING TEAM
C                        DETERMINATION OF SHAPE FACTOR IN FORGING
C                    MODIFICATION  IF BEGINNING AND END CORNERS ARE
C                    NOT FLAT
      COMMON R(100),NR,CH(100),ALFA(100),PER(100),SURF(100),VOL(100),
     1RG(100),FR(100),MARK(100),DELH(100)
      PI=3.1415926536
C                               R(I)= RADIUS OF THE FORGING AT VARIOUS
C                               LOCATIONS
C                               NR= NUMBER OF CORNERS ON ONE HALF OF THE
C                               FORGING
C                               CH(I)= AXIAL DISTANCE BETWEEN I-TH AND
C                               (I-1)TH CORNER
C                               ALFA(I)= ANGLE OF THE TAPER AT THE I-TH
C                               CORNER WITH THE AXIS
C                    PER(I)= PERIMETER OF A SLICE
C                    SURF(I)=SURFACE AREA OF A SLICE
C                    VOL(I)=VOLUME OF A SLICE
C                               FR(I)= FILLET RADIUS BETWEEN THE I-TH AND
C                               (I-1)TH CORNERS ,FR(I)=0. FOR SHARP CORNER
C                             MARK(I)=1 FOR INCREASING VOLUME
C                            MARK(I)=-1 FOR DECREASING VOLUME
C               NR1=NUMBER OF CORNERS FROM BEGIN TO PARTING LINE
C               NR2=NUMBER OF CORNERS FROM PARTING LINE TO END
   74 READ(1,333)LAYOUT
      WRITE(2,333)LAYOUT
      READ(1,21)NR1,NR2
      IF(NR1.EQ.0)GO TO 73
      WRITE(2,22)NR1,NR2
```

```
      NR=(NR1-1)*2+(NR2-1)*2
      NRD=(NR1-1)*2
      NRDM=NRD-1
      WRITE(2,86)
      DO 79 I=1,NRDM,2
      READ(1,82) MARK(I),R(I),CH(I),FR(I),ALFA(I)
      WRITE(2,87)MARK(I),R(I),CH(I),FR(I),ALFA(I)
      IF(I.EQ.1)GO TO 81
      MARK(I-1)=MARK(I)
      R(I-1)=R(I)
      CH(I-1)=CH(I)
      FR(I-1)=FR(I)
   81 ALFA(I-1)=ALFA(I)
   79 CONTINUE
      READ(1,82) MARK(NRD),R(NRD),CH(NRD),FR(NRD),ALFA(NRD)
      WRITE(2,87)MARK(NRD),R(NRD),CH(NRD),FR(NRD),ALFA(NRD)
      NRDP=NRD+1
      NRM=NR-1
      DO 80 I=NRDP,NRM,2
      READ(1,82) MARK(I),R(I),CH(I),FR(I),ALFA(I)
      WRITE(2,87)MARK(I),R(I),CH(I),FR(I),ALFA(I)
      IF(I.EQ.NRDP) GO TO 83
      MARK(I-1)=MARK(I)
      R(I-1)=R(I)
      CH(I-1)=CH(I)
      FR(I-1)=FR(I)
   83 ALFA(I+1)=ALFA(I)
   80 CONTINUE
      READ(1,82)MARK(NR),R(NR),CH(NR),FR(NR),ALFA(NR)
      WRITE(2,87)MARK(NR),R(NR),CH(NR),FR(NR),ALFA(NR)
C              START CALCULATING ANY R(I)'S NOT GIVEN AND ALSO
C              DCYL,MAX DIAMETER OF FORGING.
      NRMIN=NR-1
      CONVERT=PI/180.
      RCYL=0.
```

```
      DO 88 I=1,NPMTN,2
      IF(I.EQ.1)GO TO 84
      IF(R(I).NE.0.)GO TO 84
      IF(I.EQ.NRDP) GO TO 85
      ADIFF=ALFA(I-2)-90.
      ADIFF=ABS(ADIFF)
      IF(ADIFF.LT.0.00001) GO TO 85
      ANGLE=ALFA(I-2)
      RI=R(I-2)
      CHI=CH(I)
      VALUE=-1
      GO TO 99
   85 ANGLE=ALFA(I)
      RI=R(I+1)
      CHI=CH(I+1)
      VALUE=1
   99 ANGLE=ANGLE*CONVERT
      R(I)=RI-(VALUE*TAN(ANGLE)*CHI)
      IF(I.GT.NRD) R(I)=RI+(VALUE*TAN(ANGLE)*CHI)
      R(I-1)=R(I)
   84 CONTINUE
      IF(R(I).GT.RCYL) RCYL=R(I)
      RCYL=RCYL
   88 CONTINUE
      DCYL=2*RCYL
C              START CONVERTING ALFA(I) TO RADIANS AND DETERMINE
C              SIGN, + OR -
      DO 32 I=1,NR
      ALFA(I)=ALFA(I)*CONVERT
      IF(I.LE.NRD) ALFA(I)=-ALFA(I)
   32 ALFA(I)=ALFA(I)
      WRITE(2,22)NR
      WRITE(2,24)
      WRITE(2,23)(R(I),I=1,NR)
      WRITE(2,25)
```

```
      WRITE(2,23)(CH(I),I=1,NR)
      WRITE(2,29)
      WRITE(2,23)(FR(I),I=1,NR)
      WRITE(2,27)
      WRITE(2,23)(ALFA(I),I=1,NR)
      WRITE(2,23)(PER(I),I=1,NR)
      WRITE(2,21)(MARK(I),I=1,NR)
      READ(1,23) DENS,DIAM
C              START CALCULATING HCYL , MAX HEIGHT OF FORGING
      ACOUNT=0.
      BCOUNT=0.
      MIN=0.
      MAX=0.
      LIMIT=NR-NRD-2
      DO 126 I=2,NRD,2
      BCOUNT=BCOUNT+CH(I)*MARK(I)
      IF(BCOUNT.LT.MIN) GO TO 124
      MIN=MIN
      GO TO 126
  124 MIN=0.
      BCOUNT=0.
  126 CONTINUE
      DO 122 J=0,LIMIT,2
  128 ACOUNT=ACOUNT+CH(NR-J)*(-MARK(NR-J))
      IF (ACOUNT.GT.MAX) GO TO 130
      MAX=MAX
      GO TO 136
  130 MAX=0.
      ACOUNT=0.
  136 IF(J.NE.LIMIT) GO TO 122
      J=J+1
      GO TO 128
  122 CONTINUE
      HCYL=BCOUNT-ACOUNT
C              START CALCULATING TAHZ, THE MIN DISTANCE BETWEEN FLATS
```

```
      ACOUNT=0.
      BCOUNT=0.
      MAX=0.
      MIN=0.
      RADUS=DIAM/2.
      DO 200 I=2,NRD,2
      ACOUNT=BCOUNT+CH(I)*MARK(I)
      BDIFF=R(I)-RADUS
      IF(BDIFF.LT.-0.00001) GO TO 220
      GO TO 200
  220 IF(BCOUNT.GT.MIN) GO TO 230
      GO TO 200
      MIN=0.
  230 BCOUNT=0.
  200 CONTINUE
      DO 210 J=0,LIMIT,2
  240 ACOUNT=ACOUNT+CH(NR-J)*(-MARK(NR-J))
      CDIFF=R(NR-J-1)-RADUS
      IF(CDIFF.LT.-0.00001) GO TO 250
      GO TO 260
  250 IF(ACOUNT.LT.MAX) GO TO 270
      GO TO 260
  270 ACOUNT=0.
      MAX=0.
  260 IF(J.NE.LIMIT) GO TO 210
      J=J+1
      GO TO 240
  210 CONTINUE
      TAHZ=BCOUNT-ACOUNT
      TAHZ=ABS(TAHZ)
C               START CALCULATING TAHA , THE DISTANCE BETWEEN INT. AND
C               EXT. PARTING LINES
      ABSAA=ABS(ACOUNT)
      ABSBB=ABS(BCOUNT)
      TAHA=ABSBB-TAHZ/2.
```

```
      IF(ABSAA.GT.ABSBB) TAHA=ABSAA-TAHZ/2.
      IF(CH(NRD+1).EQ.0.) GO TO 280
      TAHA=ABSBB-TAHZ/2.+(CH(NRD+1)/2.)
      IF(ABSAA.GT.ABSBB) TAHA=ABSAA-TAHZ/2.-(CH(NRD+1)/2.)
  280 WRITE(2,28) DENS,DIAM,DCYL,TAHZ,TAHA,HCYL
C                                   DENS=DENSITY
C                                   DIAM=STOCK DIAMETER
C                                   DCYL=LARGEST DIAMETER OF FORGING
C                                   TAHZ=MINIMUM DISTANCE BETWEEN FLATS
C                                   TAHA=DISTANCE BETWEEN INTERNAL AND
C                                   EXTERNAL PARTING LINES
C                                   HCYL=TOTAL HEIGHT OF FORGING
      TVOL=0.
      TPER=0.
      TRGS=0.
      TH=0.
C                              START CALCULATIONS
      NLIM=NR-1
      DO1311I=3,NLIM,2
      IF(ALFA(I-1).EQ.0.)GO TO 41
C                                 THIS CHECKS THE PARTING LINE WHICH IS
C                                 CONSIDERED AS CYLINDER
      DIFF=ALFA(I-2)-ALFA(I)
      BETA=ABS(DIFF)
      PER(I)=BETA*FR(I)
      B2=BETA/2.
      T2=TAN(B2)
      S2=SIN(B2)
      C2=COS(B2)
      FR1=FR(I)*T2
      DELRM=FR1*SIN(ALFA(I-2))
      DELR =FR1*SIN(ALFA(I))
C                                 CORNER RADIUS WAS ORIGINALLY STORED AT
C                                 BOTH R(I) AND R(I-1)
      DELH(I)=FR1*COS(ALFA(I))
```

```
      DELH(I)=ABS(DELH(I))
      DELH(I-1)=FR1*COS(ALFA(I-2))
      DELH(I-1)=ABS(DELH(I-1))
      TAH=DELH(I)+DELH(I-1)
      SBETA=SIN(BETA)
      TAS=FR(I)**2*(BETA-SBETA)/2.
      TAXS=2.*FR(I)**3*S2**3/(3.*TAS)
      ALF2=ABS(ALFA(I-2))
      GAMMA=B2+0.5*PI-ALF2
C                              MODIFY HERE
      IF(ALFA(I-2).LT.0.)GAMMA=0.5*PI-B2-ALF2
      GAMMA=ABS(GAMMA)
      IF(ALFA(I).GT.ALFA(I-2))GO TO 51
      DELX=(FR(I)/B2-TAXS)*SIN(GAMMA)
      RS=R(I)+DELX
   52 CONTINUE
      R(I)=R(I)-DELR
      R(I-1)=R(I-1)+DELRM
      S1=R(I)+R(I-1)
      SURF(I)=S1*TAH/2.-TAS
      RIRMI=R(I)**2+R(I-1)**2+R(I)*R(I-1)
      VOL(I)=PI*(RIRMI*TAH/3.-TAS*2.*RS)
      RGCON=RIRMI/(3.*S1)
      RG(I)=(RGCON*S1*0.5+TAH-TAS*RS)/SURF(I)
   42 CONTINUE
C                                NOW CALCULATE FOR THE TRUNCATED CONE
C                                AS DEFINED BY POINTS I-1 AND I-2
      TAH1=CH(I)-DELH(I-1)-DELH(I-2)
C                                CH(I)=HEIGHT BETWEEN CORNERS STORED AT
C                                BOTH CH(I) AND CH(I-1)
   43 CONTINUE
      R11=R(I-1)+R(I-2)
      SURF(I-1)=R11 *0.5*TAH1
      RIRM1=R(I-1)**2+R(I-2)**2+R(I-1)*R(I-2)
      VOL(I-1)=PI*TAH1*RIRM1/3.
```

```
      DEN=SIN(ALFA(I-2))
      IF(ALFA(I-2).EQ.0.)DEN=1.
      PER(I-1)=ABS((R(I-1)-R(I-2))/DEN)
      RG(I-1)=RTRM1/(3.*RT1)
      TABS=ABS(DEN)-1.
      TABS=ABS(TABS)
      IF(TABS.GT.0.00001)GO TO 31
      SURF(I-1)=0.
      RG(I-1)=0.
      VOL(I-1)=0.
      GO TO 31
   41 CONTINUE
C                              CALCULATIONS FOR CYLINDER AT PARTING LINE
C                        AND FOR THE ADJACENT TRUNCATED CONE
      RG(I)=R(I)/2.
      SURF(I)=R(I)*CH(I)
      VOL(I)=PI*R(I)**2*CH(I)
      PER(I)=CH(I)
      TAHT=CH(I-1)-DELH(I-2)
      DELH(I)=0.
      GO TO 43
   51 CONTINUE
      PALF2=ABS(ALFA(I-2))
      GAMMA=0.5*PI-B2-PALF2
      IF(ALFA(I-2).LT.0.)GAMMA=0.5*PI-PALF2+B2
      GAMMA=ABS(GAMMA)
      DELX=(FR(I)/C2-TAXS)*SIN(GAMMA)
      RS=R(I)-DELX
      TAS=-TAS
      GO TO 52
   31 CONTINUE
      IK=I+1
      DO132K=1,3
      KK=IK-K
  132 WRITE(2,133)KK,PER(KK),SURF(KK),VOL(KK),RG(KK)
```

```
      GAM=GAMMA*180./PI
      WRITE(2,133)I,BETA,DELH(I),DELH(I-1),TAH,R(I),R(I-1),TAH1,GAM
      BETB=BETA*180./PI
      WRITE(2,133)I,BETB,R2,T2,S2,C2,FR1,DELR,DELRM
  131 CONTINUE
      SURF(NR)=0.
      VOL(NR)=0.
      RG(NR)=0.
      PER(NR)=ABS((R(NR)-R(NR-1)))
      TPER=0.
      TSURF=0.
      TVOL=0.
      RGS=0.
C               CALCULATE PERIMETER,VOLUME,SURFACE AND RAD. OF CENT. OF
C                GRAVITY
      DO61I=2,NR
      FMARK=1.
      TPER=TPER+PER(I)
      IF(MARK(I).LT.0)FMARK=-1.
      TSURF=TSURF+FMARK*SURF(I)
      TVOL=TVOL+FMARK*VOL(I)
      RGS=RGS+FMARK*RG(I)*SURF(I)
   61 CONTINUE
      RGT=RGS/TSURF
      TSURF=2.*TSURF
      TPER=2.*TPER
      WRITE(2,911)
      WRITE(2,67)TPER,TSURF,TVOL,RGT
C                                START CALCULATION OF SHAPE FACTOR
      EXF=TPER**2/TSURF
      EXC=4.*(HCYL+DCYL)**2/(HCYL*DCYL)
      ALFS=EXF/EXC
      RCYL=DCYL/2.
      BETS =2.*RGT/RCYL
      SHAPE=ALFS*BETS
```

```
      WRITE(2,62)SHAPE,ALFS,BETS
      WEIGH=TVOL*DENS
      WRITE(2,63)WEIGH,TVOL
C               FOR FLASH THICKNESS WE USE WOLF-S FORMULA
      T1=WEIGH/2.2046
      FTAT=0.89*T1**0.5-0.017*T1
      FTHICK=(1.13+FTAT)/25.4
C               FOR FLASH RATIO WE USE WOLF-S FORMULA IF WEIGHT IS
C               SMALLER THAN 1  LBS,OTHERWISE TETERIN-S FORMULA
      IF(WEIGH.LT.1.)GO TO 71
      RATIO=-0.02+0.0038*SHAPE*DCYL/FTHICK+4.93/(T1**0.2)
      GO TO 72
   71 PART=-1.09*T1
      RATIO=1.25*EXP(PART)+3.
   72 CONTINUE
      FWIDTH=RATIO*FTHICK
      WRITE(2,64)FTHICK,FWIDTH,RATIO
      GAM=HCYL/(TANZ*TANA)
      ETA=SHAPE*DIAM**2*GAM**2/DCYL**2
      AK2=0.7026*(1.+ETA*0.01969)*RATIO
      AK1=0.54+15.44*(T1**(-0.2))*(1.+0.00757*ETA)
      FLASHW=(AK1-AK2)*WEIGH/100.
      WTOT=WEIGH+FLASHW
      WRITE(2,65)WEIGH,FLASHW,WTOT
      DTOT=DCYL+2.*FWIDTH
      AREA=PI*DTOT**2/4.
      WRITE(2,66)AREA
  911 FORMAT(40H PERIMETER,SURFACE,VOLUME,R OF C.GRAVITY)
   62 FORMAT(27H SHAPE DIFFICULTY FACTOR IS,3F15.5)
   63 FORMAT(32H FORGING WEIGHT WITHOUT FLASH IS,3F15.5)
   64 FORMAT(44H FLASH THICKNESS,FLASH WIDTH,FLASH RATIO ARE,3F15.5)
   65 FORMAT(41H FORGING WEIGHT,FLASH WEIGHT,TOTAL WEIGHT,3F15.5)
   66 FORMAT (38H THE PROJECTED AREA INCLUDING FLASH IS,F15.5)
   67 FORMAT(11H PERIMETER=,F15.5,9H SURFACE=,F15.5//,8H VOLUME=,F15.5,2
     15H RADIUS OF C. OF GRAVITY=,F15.5)
```

```
 28 FORMAT(19H MATERIAL DENSITY =,F10.4,17H STOCK DIAMETER =,F10.4,/19
   1H LARGEST DIAMETER =,F10.4,/10H HEIGHTS =,3F10.4)
 27 FORMAT(27H ANGLES BETWEEN CORNERS ARE)
 29 FORMAT(18H FILLET RADII ARE )
 25 FORMAT(36H AXIAL DISTANCES BETWEEN CORNERS ARE)
 24 FORMAT(25H RADII OF THE CORNERS ARE)
 22 FORMAT(18H NUMBER OF CORNERS,2I15)
 86 FORMAT(11H INPUT DATA/70H PLUS/MINUS  CORNER RADIUS  AXIAL DISTANC
   1E  FILLET RADIUS  DRAFT ANGLE  CORNER TYPE)
 87 FORMAT(I6,F15.4,2F16.4,2F13.4)
 23 FORMAT(8F10.4)
133 FORMAT( I5,8F12.3)
 21 FORMAT(8I10)
 82 FORMAT(I10,6F10.4)
333 FORMAT(1H1,10A8)
    GO TO 74
 73 STOP
    END
    FINISH
```

Appendix III

Shape difficulty factor for axisymmetric forgings.

Tetrin et al.[29] devised a shape difficulty factor for axisymmetric forgings which these authors include in a rather complex expression for predicting flash weight for such forgings. This expression has been discussed in section 2.3.2.. An explanation of the Teterin shape difficulty factor is given here.

A shape factor is defined as:-

$$x_f = \frac{P_f^2}{F_f}$$

(Where; P_f = Perimeter of axial C.S.A.

F_f = Surface area ---"-----)

This factor is a dimensionless value, enabling any convenient unit to be used for evaluation. A high value of numerator and/or a low value of denominator implies a high degree of complexity in the axial C.S.A. . In deriving their expression, Teterin et al considered the circumscribing cylinder. This cylinder is assumed to be the easiest forgeable shape, any modification to this fundamental shape representing a deviation from simplicity. Thus, we have a shape factor for a simple cylinder such that:-

$$x_c = \frac{P_c^2}{F_c}$$

In order that the forging shape may be compared to the (reference) cylindrical shape, the authors suggested what they

term a "longitudinal shape factor", α.

$$\alpha = \frac{x_f}{x_c}$$

A forging with a simple shape, very near to that of a cylinder would have a value near to 1.0 , while increasingly complex forgings would have values progressively greater than 1.0 . The authors recognised that projections or rims were increasingly more difficult to form as they were located at progressively greater distances from the center-line of the forging, figures A2(a), A2(b). To accommodate these variations, Teterin et al. proposed another, lateral, shape difficulty factor:-

$$\beta = \frac{2.Rg}{Rc}$$

(Where; Rg = Radial distance from the axis of the center of gravity of half the C.S.A. .

Rc = Maximum radius of forging. (i.e. Radius of circumscribing cylinder.)

As with the longitudinal factor, a shape near to a cylinder has a value approaching 1.0 , while more complex shapes (with projections towards the circumference of the forging) have values of β greater than 1.0 .

These two sub-factors are combined to give a shape difficulty factor, S_f :-

$$S_f = \alpha . \beta$$

This may be expanded by substitution to :-

$$S_f = \frac{\left(\frac{P_f}{F_f}^2\right)}{\left(\frac{P_c}{F_c}^2\right)} \cdot \frac{2.Rg}{Rc}$$

If the part is forged in more than one operation, the authors suggest that a factor for each step should be calculated.

The factors, they suggest, should be defined in terms of the starting shape for the operation of interest (i.e. the shape resulting from the previous forging operation).

Thus:-

$$S = \frac{S_1}{S_0}$$

(Where; S_1 = Shape difficulty factor for the forging step under consideration.

S_o = Shape difficulty factor for the work piece prior to this forging step.) .

If the starting shape is a cylinder, then this expression simplifies to:-

$S = S_1$ (S_0 being 1.0)

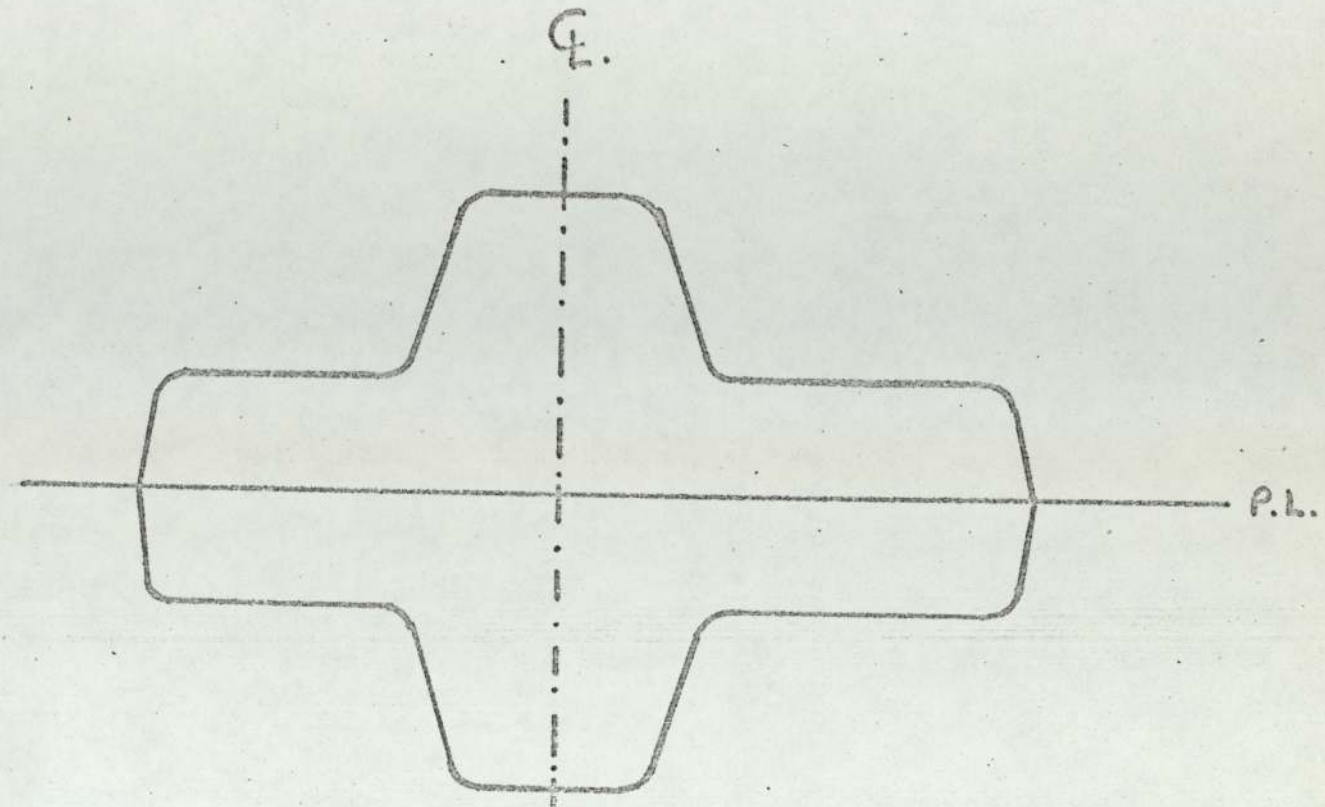


Figure A2(a).

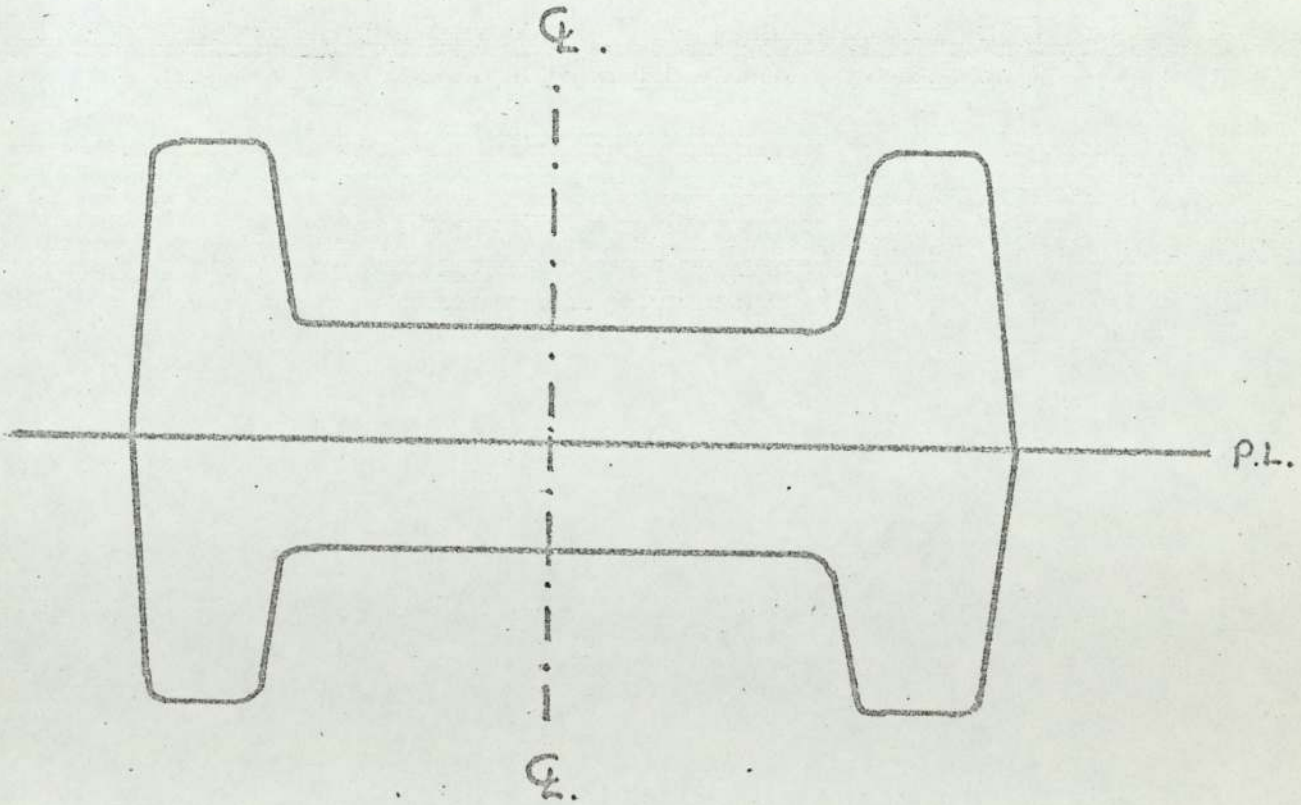


Figure A2(b).

Appendix IV

FORTRAN programme for plotting results of regression analysis.

The use of a FORTRAN programme for plotting the results of regression analyses was mentioned in section 5.1. . , a listing of this programme is given here. The programme was written in such a way that it may be used by almost anyone who requires their experimental or industrial data in a graphical form.

The method and logic are described in detail in a University Of Aston internal report[75] . The data is input on cards in a specific, formated manner. The data cards required fall into four distinct categories:-

1. TYPE 'A' DATA - Initial parameters specifying the amount and form of the observations to be expected.
2. TYPE 'B' DATA - Transformation data, cards required to initiate and carry out the transformation of variables.
3. TYPE 'C' DATA - Observation data proper, i.e. users experimental or industrial results.
4. TYPE 'D' DATA - Graph titles, axis titles etc.

The output obtained from this programme includes; numerical results (observed value, calculated value, error) on the line-printer and various graphical displays on a CALCOMP x-y plotter

The graphs produced are:-

i. A plot of 'y' versus 'x_1' ; $x_2, x_3, ---- x_n$ being held constant at some predetermined, typical level.

ii. A plot of of 'y' versus 'x_2' ; $x_1, x_3 ----- x_n$ being held constant.

etc.

etc.

etc.

n. A plot of 'y' versus 'x_n' ; $x_1, x_2, x_3, ---- x_{(n-1)}$ being held constant.

n+1. A plot of observed versus calculated value of 'y', using the full regression equation.

Examples of graphs obtained by the use of this programme may be seen in section 5.1. . (Anyone requiring detailed information regarding the use of this programme is referred to Aston and Homer[75].

```
      PROGRAM (FYXX)
      INPUT  1 = CR0
      INPUT  3 = TR0
      INPUT  5 = CR1
      OUTPUT 2 = LP0
      OUTPUT 6 = LP1
      COMPACT
      NO TRACE
      END
      MASTER(MULP)
      REAL L1,L2
      DIMENSION VAR(20),AVRG(20),COEFF(20),VAL(20),A(100),NOVAR(20),PT(1
     *0),XT(10),YT(10),PT1(10),XT1(10),YT1(10),B(600),C(600),D(600)
     *,E(300),XFACT(10),AMAXX(10),AMINX(10),SMINY(10),SMINX(10)
     *,SMAXX(10),SMAXY(10),NFUNCT(20),NY(20),NX(20),CONYY(20),NZ(20)
      CALL OPENPLOT
      READ(1,200) N0,N1,N2,N3,N4,N5
      READ(1,300) NI,ND,(NOVAR(N),N=1,NI)
      READ(1,105)AMINY,AMAXY
      READ(1,101)(AMINX(N),N=1,NI)
      READ(1,101)(AMAXX(N),N=1,NI)
      READ(1,101)(AVRG(N),N=1,NI)
      READ(1,102)(COEFF(N),N=1,NI),CONST
      WRITE(2,200)(NOVAR(N),N=1,NI)
      AMINOBS=AMINY
      AMAXOBS=AMAXY
      IF(AMINY.GE.AMAXY) GO TO 333
      DO 335 N=1,NI
  335 IF(AMINX(N).GE.AMAXX(N)) GO TO 333
      STRD=CONST
      DO 120 N=1,NI
      STRD=STRD+COEFF(N)*AVRG(N)
  120 CONTINUE
      I,J,K=1
      M=0
```

```
  222 M=M+1
      READ(1,799)NFUNCT(M)
      IF(NFUNCT(M).EQ.0) GO TO 223
      IF(NFUNCT(M).LE.6) GO TO 324
      IF(NFUNCT(M).LE.11) GO TO 325
      READ(1,798)NY(M),NX(M),NZ(M)
      GO TO 222
  324 READ(1,797)NY(M),NX(M)
      GO TO 222
  325 READ(1,796)NY(M),NX(M),CONYY(M)
      GO TO 222
  223 CONTINUE
    5 READ(1,110)NAME,(A(N),N=1,N0)
      IF(A(1).LT.-9.85.AND.A(1).GT.-9.995) GO TO 6
      IF(N1.EQ.0) GO TO 122
      READ(1,110)NAME,(A(N),N=N0+1,N1+N0)
      IF(N2.EQ.0) GO TO 122
      READ(1,110)NAME,(A(N),N=N0+N1+1,N0+N1+N2)
      IF(N3.EQ.0) GO TO 122
      READ(1,110)NAME,(A(N),N=N0+N1+N2+1,N0+N1+N2+N3)
      IF(N4.EQ.0) GO TO 122
      READ(1,110)NAME,(A(N),N=N0+N1+N2+N3+1,N0+N1+N2+N3+N4)
      IF(N5.EQ.0) GO TO 122
      READ(1,110)NAME,(A(N),N=N0+N1+N2+N3+N4+1,N0+N1+N2+N3+N4+N5)
  122 CONTINUE
C     TRANSFORMATIONS  BEGIN
      N=0
  224 M=M+1
      IF(NFUNCT(M).EQ.0) GO TO 225
      MARK=NFUNCT(M)
      GO TO(11,22,33,44,55,66,77,88,99,100,111,112,113,114,115),
     1MARK
   11 A(NY(M))=ALOG10(A(NX(M)))
      GO TO 224
   22 A(NY(M))=ALOG(A(NX(M)))
```

```
      GO TO 224
   33 A(NY(M))=SIN(A(NX(M)))
      GO TO 224
   44 A(NY(M))=COS(A(NX(M)))
      GO TO 224
   55 A(NY(M))=TAN(A(NX(M)))
      GO TO 224
   66 A(NY(M))=SQRT(A(NX(M)))
      GO TO 224
   77 A(NY(M))=(A(NX(M)))**CONYY(M)
      GO TO 224
   88 A(NY(M))=A(NX(M))+ CONYY(M)
      GO TO 224
   99 A(NY(M))=A(NX(M))-CONYY(M)
      GO TO 224
  100 A(NY(M))=A(NX(M))*CONYY(M)
      GO TO 224
  111 A(NY(M))=A(NX(M))/CONYY(M)
      GO TO 224
  112 A(NY(M))=A(NY(M))+A(NZ(M))
      GO TO 224
  113 A(NY(M))=A(NX(M))-A(NZ(M))
      GO TO 224
  114 A(NY(M))=A(NX(M))*A(NZ(M))
      GO TO 224
  115 A(NY(M))=A(NX(M))/A(NZ(M))
      GO TO 224
  225 CONTINUE
C     TRANSFORMATIONS  END
      DO 2 M=1,NT
      M=NOVAR(N)
      VAR(N)=A(M)
      B(I)=VAR(N)
    2 I=I+1
      CRECTD=A(ND)
```

```
      DO 3 M=1,NI
    3 CRECTD=CRECTD-COEFF(M)*(VAR(M)-AVRG(M))
      CALC=CONST
      DO 4 M=1,NI
      VAL(M)=CRECTD+COEFF(M)*(VAR(M)-AVRG(M))
      C(K)=VAL(M)
      CALC=CALC+COEFF(M)*VAR(M)
    4 K=K+1
      D(J)=A(ND)
      E(J)=CALC
      J=J+1
      ERROR=(CALC-A(ND))*100.0/A(ND)
      WRITE(2,201) NAME,A(ND),(VAR(M),M=1,NI)
      WRITE(2,203) CALC,(VAL(M),M=1,NI)
      WRITE(2,202) ERROR
      GO TO 5
C     THIS OBSERVATION PROCESSED , LOOPS BACK TO 5 TO COMMENCE
C     NEXT OBSERVATION . WHEN LAST OBS PROCESSED ,BEGINS TO
C     PLOT GRAPHS.......
    6 CONTINUE
      CALL HGPLOT(-5.0,24.0,0,4)
      READ(1,106) S1
      XREC=S1+1.50
      YREC=S1+5.25
      MS=(I-1)/NI
      YFACT=(AMAXY-AMINY)/S1
      DO 998 N=1,I-1
  998 C(N)=(C(N)-AMINY)/YFACT
      DO 999 N=1,NI
  999 XFACT(N)=(AMAXX(N)-AMINX(N))/S1
      DIFFA=ABS(COEFF(1))+0.00001
      DIFFB=CONST+0.00001
      IF(DIFFA.LT.0.0001.AND.DIFFB.LT.0.0001) GO TO 997
C     FOLLOWING PROCEDURE , TO 997, SCALES RESULTS , SETS
C     AXIS SCALES E.T.C....
```

```
      DO 997 N=1,NT
      TEMINX=AMINX(N)-XFACT(N)
  995 TEMINX=TEMINX+XFACT(N)
      SMINY(N)=STRD-COEFF(N)*AVRG(N)+COEFF(N)*TEMINX
      IF(SMINY(N).LT.AMINY.OR.SMINY(N).GT.AMAXY) GO TO 995
      SMINY(N)=(SMINY(N)-AMINY)/YFACT
      SMINX(N)=(TEMINX-AMINX(N))/XFACT(N)
      TEMAX)=AMAXX(N)+XFACT(N)
  994 TEMAXX=TEMAXX-XFACT(N)
      SMAXY(N)=STRD-COEFF(N)*AVRG(N)+COEFF(N)*TEMAXX
      IF(SMAXY(N).LT.AMINY.OR.SMAXY(N).GT.AMAXY) GO TO 994
      SMAXY(N)=(SMAXY(N)-AMINY)/YFACT
      SMAXX(N)=(TEMAXX-AMINX(N))/XFACT(N)
  997 CONTINUE
C     FOLLOWING PROCEDURE , TO 7 , CONSIDERS ONE VARIABLE ONLY , AND
C     PLOTS GRAPH FOR THAT VARIABLE.
      READ(1,104) YT
      DO 7 L=1,NT
      READ(1,104) PT
      READ(1,104) XT
      CALL HGPSYMBL(1.0,-2.5,0.25,PT,0.0,80)
      CALL HGPSYMBL(1.0,-0.75,0.2,XT,0.0,40)
      CALL HGPSYMBL(-0.75,1.0,0.2,YT,90.0,40)
      XORIGN=AMINX(L)
      XINCMT=XFACT(L)
      YORIGN=AMINY
      YINCMT=YFACT
      CALL HGPAXIS(0.0,0.0,14HSCALE READINGS,-14,S1,0.0,XORIGN,XINCMT)
      CALL HGPAXIS(0.0,0.0,14HSCALE READINGS,14,S1,90.0,YORIGN,YINCMT)
      DO 8 M=1,I-1,NT
      B(M)=(B(M)-AMINX(L))/XFACT(L)
      CALL HGPLOT(B(M),C(M),3,0)
      CALL HGPLOT(B(M)+0.05,C(M),2,0)
      CALL HGPLOT(B(M)-0.05,C(M),1,0)
      CALL HGPLOT(B(M),C(M),1,0)
```

```
      CALL HGPLOT(B(M),C(M)+0.05,1,0)
      CALL HGPLOT(B(M),C(M)-0.05,1,0)
      CALL HGPLOT(B(M),C(M),1,0)
    8 CONTINUE
      IF(DIFFA.GT.0.0001.AND.DIFFB.GT.0.0001)
     1CALL HGPDASHN(SMINX(L),SMINY(L),SMAXX(L),SMAXY(L),0.5)
      CALL HGPRECT(-1.1,-3.5,YREC,XREC,0.0,3)
C     FOLLOWING PROCEDURE ,TO 10 ,DECIDES WHERE TO MOVE PEN
C     BETWEEN GRAPHS IN ORDER TO OBTAIN CORRECT POSITIONAL LAYOUT.
      IF (L.EQ.NI) GOTO 7
      CALL HGPLOT(0.0,0.0,3,0)
      IF(L/2*2.EQ.L) GOTO 9
      CALL  HGPLOT(0.0,-13.0,0,4)
      GOTO 7
    9 CALL HGPLOT(-10.0,13.0,0,4)
    7 CONTINUE
      CALL HGPLOT(0.0,0.0,3,0)
      IF(NI/2*2.EQ.NI) GOTO 10
      CALL HGPLOT(-10.0,0.0,0,4)
      GOTO 13
   10 CALL HGPLOT(-10.0,13.0,0,4)
   13 D(J),E(I)=0.0
      IF(DIFFA.LT.0.0001.AND.DIFFB.LT.0.0001) GO TO 334
C     FOLLOWING PROCEDURE , TO 12 , CONSIDERS OBS. V CALC.
C     GRAPH
   16 CALL HGPSYMBI(1.0,-2.5,0.25,22HOBSERVED V CALCULATED.,0.0,22)
      CALL HGPSYMBI(1.0,-0.75,0.2,19H    OBSERVED VALUE.,0.0,19)
      CALL HGPSYMBI(-0.75,1.0,0.2,19H  CALCULATED VALUE.,90.0,19)
      OBFACT=(AMAXOBS-AMINOBS)/S1
      CALL HGPAXIS(0.0,0.0,0.14HSCALE READINGS,-14,S1,0.0,AMINOBS,OBFACT)
      CALL HGPAXIS(0.0,0.0,0.14HSCALE READINGS,14,S1,90.0,AMINOBS,OBFACT)
      DO 12  M=1,J-1
      D(M)=(D(M)-AMINOBS)/OBFACT
      E(M)=(E(M)-AMINOBS)/OBFACT
      CALL HGPLOT(D(M),E(M),3,0)
```

```
      CALL HGPLOT(D(M)+0.05,E(M),2,0)
      CALL HGPLOT(D(M)-0.05,E(M),1,0)
      CALL HGPLOT(D(M),E(M),1,0)
      CALL HGPLOT(D(M),E(M)+0.05,1,0)
      CALL HGPLOT(D(M),E(M)-0.05,1,0)
      CALL HGPLOT(D(M),E(M),1,0)
   12 CONTINUE
      CALL HGPDASHIN(0.9,0.0,31,31,0.5)
      CALL HGPRECT(-1.1,-3.5,YREC,XREC,0.0,3)
      CALL HGPLOT(0.0,0.0,3,0)
      CALL HGPLOT(10.0,24.0,3,0)
      GO TO 334
  333 WRITE(2,795)
  334 CALL CLOSEPLOT
  300 FORMAT(2310)
  101 FORMAT(20F0.0)
  102 FORMAT(20F0.0)
  110 FORMAT(A8,20F0.0)
  104 FORMAT(10A8)
  105 FORMAT(2F0.0)
  200 FORMAT(/24H  ROW NAME OBS/CALC/PERR,20(I7,5X)//)
  201 FORMAT(A8,4X,20F12.4)
  202 FORMAT(12X,E12.4/)
  203 FORMAT(12X,20F12.4)
  106 FORMAT(1F0.0)
  299 FORMAT(6I0)
  298 FORMAT(1I0)
  797 FORMAT(3I0)
  796 FORMAT(2I0)
  795 FORMAT(/65H  ERROR IN INPUT    MAX. AND MIN. VALUES OF X OR Y ARE
     1INTERPOSED/)
      STOP
      END
      FINISH
```

Appendix V
Further discussion of the problem of hammer selection.

In section 4.2.2. a method of automating the choice of hammer unit for the manufacture of any forging was discussed. The approach was essentially one using the technique of discriminant analysis to develop a function capable of making such a prediction with a high degree of confidence. It was pointed out that, due to the lack of suitably documented records of erroneous predictions, the best that such an analysis could achieve would be a function capable of simulating the man's human judgment.

The whole question of hammer selection is discussed below. A scheme of research is proposed which, it is hoped, will result in a more rational method for the choice of optimum production unit for any given forging.

The estimator at the study firm chooses the production unit on the basis of the smallest (least energy available) unit capable of producing a satisfactory, completely filled, forging. This is an estimate based chiefly on experience.

By choosing the smallest suitable unit to be the optimum unit, the approach implies that the savings arising from the increased production rate possible with a larger unit (i.e. fewer blows required for a given forging) are completely out weighed by the increased costs carried by the larger production units. In particular, at the study forge, the larger units are burdened with substantially higher

contribution rates, making them less financially attractive than the smaller units. The final choice of unit is influenced to some degree by the ancillary equipment necessary for production; wide-bed press, dummying facilities etc. Not all production units have the same combination of ancillary facilities.

The first stage would be devising an objective method of predicting the smallest suitable production unit. One such method is to use multiple regression analysis to produce a function capable of predicting the number of blows required on any particular hammer, from the various forge-component characteristics. Exceeding a certain number of blows indicates that a given unit is unsuitable, too small, because the work piece would cool below forging temperature during the prolonged forging cycle. This 'cut-off' point should not be too difficult to assess with the aid of forge foreman, supervisors etc., and by observation. (It has been suggested that there may also be some upper limit to the size of production unit suitable for a given forging. If this is the case, then an upper 'cut-off' point will similarly have to be assessed).

If we take this discussion a stage further, there may be some advantage in disregarding the philosophy adopted at the study forge, that is, that the costs and cost rates carried by the larger production units far outweigh any savings due to increased production rates that may be possible using a larger hammer (requiring few blows to make satisfactory forging).

The generalized situation can be represented by the following diagram. Curve 'a' represents occupation time per forging (inversely proportional to production rate). The curve falls as it progresses from

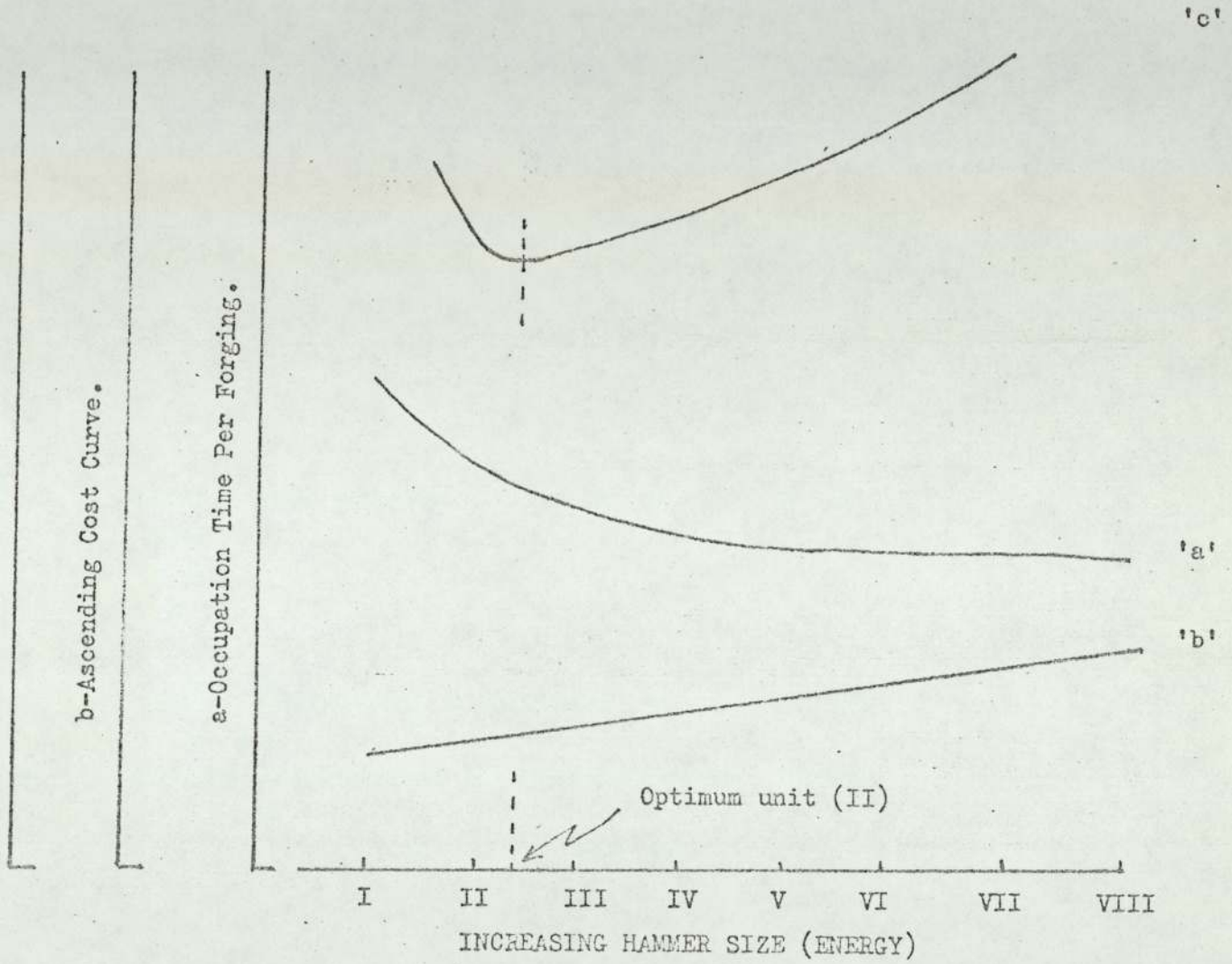


Figure A3. Generalized Case.

left to right, from larger to smaller production units. Curve 'b' represents cost rates which increase as unit size increases. For example, fuel-oil, electricity consumption, contribution or overhead rates etc. The resultant cost curve, 'c', is the product of 'a' and 'b'.

The construction of curve 'a' requires that the production rate for a given forging on any unit can be estimated. The main body of this thesis has discussed how such predictive models were constructed for the study forge using regression analysis. The approach could be extended to other forges.

The accountancy department at the Drop Forge would have to be approached in an attempt to provide a means of deducing the rising cost curve, 'b'. It is likely that the forge management could also offer various other reasons for production costs rising with increasing hammer size, for example, increased maintenance cost, increased risk of premature die failure etc.. The researcher must be aware that this cost line will inevitably be the result of rather subjective reasoning: in setting differential contribution rates, quantifying risk of premature die failure etc.

The minimum cost point on the (generalized) diagram represents the optimum choice of production unit for a particular forging, provided that the optimum unit is at least as large as the smallest suitable unit predicted by the 'Number of blows' - model.

The ordered state of affairs of the general case may be complicated, however, when considering a forge containing a mixture of both automatic and manual hammers. The two types of unit have different striking rates. A larger, manual hammer, although requiring fewer blows to make the forging, may have a lower striking rate than a smaller unit, and hence,

a lower overall production rate.

Similarly, the cost line 'b' in practice will rarely be linear. For example, there may be different numbers of crew required on certain hammers, units in the middle of the range may have disproportionally high cost rates due to the excessive fuel consumption of inefficient furnaces, contribution rates may be set low in the case of old, 'written off', equipment etc..

The net result is that both curves, 'a' and 'b', may assume any shape, making the manual estimation of the cheapest or optimum unit a tiresome process of repetitive calculation. This problem could easily be surmounted by using a computerized hammer selection programme. The smallest suitable unit could be evaluated using a model as indicated earlier. The production rate on each unit could be similarly predicted. (The cost curve, 'b', information would be stored within the computer programme). The selection of optimum production unit for the forging would then be a simple matter of computing the different production costs on each possible unit, the optimum unit for production being the least costly of these.

Clearly, the costing policy operating at the firm (relative contribution rates etc.) has to be defined, since it will influence this optimum choice by controlling the shape and slope of the cost curve, 'b'. The prediction of smallest suitable unit, however, is obviously independant of costing considerations, the prediction depending only on certain component and material characteristics. (It should be noted that the choice of optimum production unit should be made on a marginal costing basis, contribution or overhead not entering into the considerations. Many forges, however,

adhere to the belief that overheads should be considered when making this choice. This does not really effect the underlying theory of the approach, only the slope of the cost curve, 'b', and ultimately, curve 'c').

The above very brief resumè should serve to introduce a more thorough approach to the problem of optimum hammer selection. It is anticipated that such a project could be completed in three to four months (possibly as an M.Sc. student project) at a forge having good record and file keeping systems from which to obtain production data.

Appendix VI

Derivation of 'Sums of squares' expression for assessing relative contributions of each independent variable in a regression model.

In multiple regression analysis,

$$\text{SS due to regression} = b.x_1.\Sigma(x_1-\bar{x}_1).(y-\bar{y}) + b.x_2.\Sigma(x_2-\bar{x}_2).(y-\bar{y}) + \ldots\ldots\ldots b.x_n.\Sigma(x_n-\bar{x}_n).(y-\bar{y})$$

i.e.,

$$SS_{x_n} = b.x_n.\Sigma(x_n-\bar{x}_n).(y-\bar{y}) \qquad [1]$$

Now, since,

$$r^2 = \frac{b^2\Sigma(x-\bar{x})^2}{\Sigma(y-\bar{y})^2} \qquad [2]$$

and,

$$b = \frac{\Sigma(x-\bar{x}).(y-\bar{y})}{\Sigma(x-\bar{x})} \qquad [3]$$

substituting [3] into [2] we have,

$$r^2 = \frac{\{\Sigma(x-\bar{x}).(y-\bar{y})\}^2.\Sigma(x-\bar{x})^2}{\{\Sigma(x-\bar{x})^2\}^2.\Sigma(y-\bar{y})^2}$$

$$\therefore \quad r^2 = \frac{\{\Sigma(x-\bar{x}).(y-\bar{y})\}^2}{\Sigma(x-\bar{x})^2.\Sigma(y-\bar{y})^2}$$

$$\therefore \quad r = \frac{\Sigma(x-\bar{x}).(y-\bar{y})}{\sqrt{\Sigma(x-\bar{x})^2.\Sigma(y-\bar{y})^2}}$$

Rearranging,

$$\Sigma(x-\bar{x}).(y-\bar{y}) = r.\sqrt{\Sigma(x-\bar{x})^2.\Sigma(y-\bar{y})^2} \qquad [4]$$

Substituting [4] into [1] we get,

$$SS_{x_n} = b.r.\sqrt{\Sigma(x-\bar{x})^2.\Sigma(y-\bar{y})^2} \qquad \text{[5]}$$

Now, by definition,

$$\sigma x^2 = \frac{\Sigma(x-\bar{x})^2}{n-1} \qquad \text{[6]}$$

and similarly,

$$\sigma y^2 = \frac{\Sigma(y-\bar{y})^2}{n-1} \qquad \text{[7]}$$

Substituting [6] and [7] into expression [5], we get,

$$SS\,x_n = b.r.\sqrt{\sigma x^2.(n-1).\sigma y^2.(n-1)}$$

$$\therefore \quad SS\,x_n = b.r.\sigma x.\sigma y.(n-1)$$

Appendix VII

Problems associated with calendar dates in computer systems.

Scheduling systems, manual or computerized, inevitably involve the storage and manipulation of data in the form of calendar dates. Thus, one of the first decisions that has to be taken during the development of a computerized scheduling system, is the form of this 'date-data'.

It was realized that it was highly desirable, from a users point of view, that all dates should be input to, and output from (printed out) the computer system in conventional Day/Month/Year format. This presents problems. To store a date as D/M/Y requires a three dimensional array. This is extravagant with computer core store. Also, the procedure of step-wise up-dating a value held as D/M/Y necessitates a quite complex programming routine; checking number of days in each month etc.. The complex nature of a routine to perform the simple task of comparing two dates to decide which is the later (for example), and the difficulty of allowing for Saturday and Sunday as none-working days for scheduling purposes; provide further strong arguments for using some alternative form of internal 'date' storage.

The solution adopted was to store all dates internally as single, integer values, in the range;- 1 to 390. Each working day (Monday to Friday), beginning with the first working Monday in the year, is thus assigned a consecutive value; 1 to 5, 6 to 10, 11 to 15 etc.. Manipulation of dates is greatly simplified, each date having a single, arithmetic value.

To maintain continuity from the end of one year (i.e. November, December) through to the beginning of the following year (January, February etc), working days in the following year are assigned consecutive values in excess of 265 (The usual number of such days in any one year). In order that such a numbering system should not continue indefinitely, when day 390 is reached (Corresponding to approximately July of the following year), all stored values are reduced by 265, one year. Thus, day 390 becomes day 125, a stored integer value, 'Flagyear' is meanwhile increased by 1.0.

There is no need for the user to think in these terms, or even be aware of such things, the programme incorporates routines to transform D/M/Y representation to a single, integer value, and vica-versa.

Appendix VIII

Definition of scheduling terms used in this study.

Job - Refers to the work that is performed, and also, to the physical entity that is the object of the work. A job may comprise one or more operations.

Operation - Each operation is a task that must be performed on the job. The series of such tasks constituting a job usually follows some pre-defined sequence.

Facility - In a drop-forging context, a facility is a work centre where specific operations are performed. For example; a hammer, a shaper, a die-sinker etc..

Objective Function - An objective function is the criterion whereby the effectiveness of a schedule, or scheduling system, may be judged.

Heuristic - An heuristic may be defined as, any systematic device that contributes to the reduction in the search for a solution to a problem - it is an aid to the discovery of a solution.

For example, an heuristic may involve swapping the positions of neighbouring pairs of jobs and examining the effect produced on the value of the objective function.

Priority Rule - This term is ably defined in section 2.5.. It refers to the method whereby a scalar value may be ascribed to each waiting job to determine the job to be loaded next onto the relevant production facility, when it becomes available. For example, job slack. This is; Total time until due date, minus, production duration time.

Appendix IX

Data used to compare the precision of the model-based and experience-based forecasting systems.

DATA USED FOR COMPARING ESTIMATING SYSTEMS.(FLASH WEIGHT)

DIE	OBS.	MODEL	DIFF.	ERROR	EST,TOR	DIFF.	ERROR
3160	1.000	1.106	0.106	10.6	0.875	-0.125	-12.5
3085	1.750	1.249	-0.501	-28.6	1.325	-0.425	-24.3
2997	1.750	1.989	0.239	13.6	2.000	0.250	14.3
2646	0.620	0.762	0.142	22.9	1.125	0.505	81.5
3004	0.190	0.157	-0.033	-17.2	0.187	-0.003	-1.6
3132	0.750	0.663	-0.087	-11.6	0.481	-0.269	-35.9
3083	0.810	0.911	0.101	12.5	1.750	0.940	116.0
3082	0.870	0.902	0.032	3.6	1.813	0.943	108.4
2995	0.250	0.258	0.008	3.3	0.313	0.063	25.2
3112	1.560	0.902	-0.658	-42.2	1.562	0.002	0.1
3070	0.620	0.681	0.061	9.9	0.750	0.130	21.0
3025	0.310	0.515	0.205	66.0	0.250	-0.060	-19.4
2909	0.250	0.402	0.152	60.8	0.250	0.000	0.0
3080	1.620	0.727	-0.893	-55.1	0.625	-0.995	-61.4
3028	0.620	0.668	0.048	7.7	1.312	0.692	111.6
2999	0.750	1.116	0.366	48.9	0.750	0.000	0.0
2805	0.690	0.694	0.004	0.6	0.750	0.060	8.7
3141	0.690	0.745	0.055	8.0	0.525	-0.165	-23.9
2773	0.750	0.472	-0.278	-37.1	0.937	0.187	24.9
3102	1.560	1.907	0.347	22.2	4.000	2.440	156.4
3057	1.370	1.524	0.154	11.2	1.375	0.005	0.4
2921	1.370	1.067	-0.303	-22.1	1.562	0.192	14.0
2648	0.750	0.750	0.000	0.0	1.250	0.500	66.7
3107	0.120	0.263	0.143	119.3	0.125	0.005	4.2
3074	0.440	0.436	-0.004	-1.0	0.437	-0.003	-0.7
3067	0.250	0.253	0.003	1.2	0.375	0.125	50.0
2990	0.310	0.246	-0.064	-20.7	0.312	0.002	0.6
3037	0.250	0.266	0.016	6.5	0.312	0.062	24.8
2849	0.220	0.157	-0.063	-28.9	0.312	0.092	41.8
3087	0.370	0.166	-0.204	-55.2	0.315	-0.055	-14.9

2984	0.370	0.377	0.007	2.0	0.437	0.067	18.1
2951	0.250	0.306	0.056	22.4	0.281	0.031	12.4
2974	0.310	0.423	0.113	36.5	0.437	0.127	41.0
3142	0.560	0.260	-0.300	-53.6	0.430	-0.130	-23.2
3038	0.370	0.292	-0.078	-21.1	0.375	0.005	1.4
3000	0.060	0.136	0.076	126.5	0.125	0.065	108.3
2920	0.190	0.311	0.121	63.8	0.188	-0.002	-1.1
2853	0.560	0.468	-0.092	-16.4	0.562	0.002	0.4
3095	1.120	1.360	0.240	21.4	1.375	0.255	22.8

DATA USED FOR COMPARING ESTIMATING SYSTEMS.(PRODUCTION RATE)

DIE	OBS.	MODEL	DIFF.	ERROR	EST,TOR	DIFF.	ERROR
1750	71.0	93.8	22.8	32.2	98.0	27.0	38.0
2944	150.0	154.0	4.0	2.6	132.0	-18.0	-12.0
2756	108.0	145.7	37.7	34.9	133.0	25.0	23.1
2951	111.0	155.8	44.8	40.4	147.0	36.0	32.4
2775	99.0	130.5	31.5	31.9	132.0	33.0	33.3
1021	146.0	156.8	10.8	7.4	154.0	8.0	5.4
3004	221.0	178.1	-42.9	-19.4	215.0	-6.0	-2.7
1750	81.0	94.1	13.1	16.2	98.0	17.0	20.9
2604	105.0	90.6	-14.4	-13.7	64.0	-31.0	-29.5
3107	104.0	119.5	15.5	14.9	55.0	-49.0	-47.1
3125	88.0	120.7	32.7	37.2	109.0	21.0	23.8
2073	160.0	133.4	-26.6	-16.6	95.0	-65.0	-40.6
3136	141.0	166.8	25.8	18.3	189.0	48.0	34.0
2221	151.0	121.9	-29.1	-19.2	118.0	-33.0	-21.8
2990	125.0	131.4	6.4	5.1	95.0	-30.0	-24.0
1750	97.0	95.0	-2.0	-2.1	98.0	1.0	1.0
3157	200.0	162.5	-37.5	-18.7	118.0	-82.0	-41.0
1588	99.0	100.9	1.9	1.9	66.0	-33.0	-33.3
2680	116.0	137.3	21.3	18.4	168.0	52.0	44.8
89	120.0	169.4	49.4	41.2	153.0	33.0	27.5
1878	165.0	155.8	-9.2	- 5.6	158.0	-7.0	-4.2
2024	94.0	131.7	37.7	40.1	138.0	44.0	46.8
2756	169.0	141.2	-27.8	-16.4	133.0	-36.0	-21.3
2440	173.0	146.7	-26.3	-15.2	153.0	-20.0	-11.5
3094	145.0	142.3	-2.7	-1.9	177.0	32.0	22.0
2024	142.0	134.6	-7.4	-5.2	138.0	-4.0	-2.8
1304	150.0	125.3	-24.7	-16.5	86.0	-64.0	-42.6
2849	166.0	159.0	-7.0	-4.2	184.0	28.0	16.8
1588	89.0	93.2	4.2	4.8	66.0	-23.0	-25.8
2756	189.0	133.2	-55.8	-29.5	133.0	-56.0	-29.6

2494	176.0	154.3	-21.7	-12.3	176.0	0.0	0.0
2556	135.0	120.1	-14.9	-11.0	128.0	-7.0	-5.1
2495	128.0	124.1	-3.9	-3.0	115.0	-13.0	-10.1
2974	88.0	110.9	22.9	26.0	61.0	-27.0	-30.6
2936	110.0	112.9	2.9	2.7	113.0	3.0	2.7
2978	68.0	81.6	13.6	20.0	65.0	-3.0	-4.4
2713	74.0	91.9	17.9	24.2	68.0	-6.0	-8.1
2992	116.0	85.7	-30.3	-26.1	97.0	-19.0	-16.3
2790	58.0	41.9	-16.1	-27.8	46.0	-12.0	-20.6
2936	120.0	102.4	-17.6	-14.7	113.0	-7.0	-5.8
2786	90.0	77.2	-12.8	-14.2	81.0	-9.0	-10.0
2978	70.0	77.1	7.1	10.2	65.0	-5.0	-7.1
2621	89.0	103.5	14.5	16.3	83.0	-6.0	-6.7
2899	108.0	120.8	12.8	11.8	104.0	-4.0	-3.7
3001	76.0	87.6	11.6	15.3	54.0	-22.0	-28.9
2495	114.0	121.0	7.0	6.2	115.0	1.0	0.8
2500	147.0	132.7	-14.3	-9.7	118.0	-29.0	-19.7
2000	177.0	180.4	3.4	1.9	172.0	-5.0	-2.8
2537	211.0	225.4	14.4	6.8	214.0	3.0	1.4
3116	93.0	104.1	11.1	12.0	75.0	-18.0	-19.3
1693	158.0	128.4	-29.6	-18.8	118.0	-40.0	-25.3
1930	200.0	184.3	-15.7	-7.9	148.0	-52.0	-26.0
2984	88.0	84.9	-3.1	-3.5	81.0	-9.0	-10.2
2323	200.0	226.2	26.2	13.1	214.0	14.0	7.0
1693	150.0	134.0	-16.0	-10.7	118.0	-32.0	-21.3
2537	230.0	239.5	9.5	4.1	214.0	-16.0	-6.9
2849	205.0	195.1	-9.9	-4.8	184.0	-21.0	-10.2
3153	118.0	142.5	24.5	20.8	168.0	50.0	42.3
2606	113.0	121.3	8.3	7.3	102.0	-11.0	-9.7
1693	140.0	128.7	-11.3	-8.0	118.0	-22.0	-15.7
3135	183.0	213.0	30.0	16.4	179.0	-4.0	-2.1
1713	105.0	122.1	17.1	16.3	143.0	38.0	36.1
2606	115.0	127.4	12.4	10.8	102.0	-13.0	-11.3
3037	150.0	152.1	2.1	1.4	137.0	-13.0	-8.6
3159	67.0	65.7	-1.3	-1.9	61.0	-6.0	-8.9

2000	178.0	194.2	16.2	9.1	172.0	-6.0	-3.3
2537	229.0	208.3	-20.7	-9.0	214.0	-15.0	-6.5
2606	116.0	111.4	-4.6	-4.0	102.0	-14.0	-12.0
2984	77.0	61.7	-15.3	-19.8	81.0	4.0	5.1
2537	240.0	227.9	-12.1	-5.0	214.0	-26.0	-10.8
2500	139.0	123.3	-15.7	-11.3	118.0	-21.0	-15.1
1693	136.0	136.2	0.2	0.1	118.0	-18.0	-13.2
2537	252.0	227.2	-24.8	-9.8	214.0	-38.0	-15.0
2606	108.0	119.9	11.9	11.1	102.0	-6.0	-5.5
1693	142.0	139.3	-2.7	-1.9	118.0	-24.0	-16.9
1713	115.0	116.1	1.1	1.0	143.0	28.0	24.3
1079	169.0	140.6	-28.4	-16.8	115.0	-54.0	-31.9
3087	203.0	182.3	-20.7	-10.2	153.0	-50.0	-24.6
2000	174.0	166.7	-7.3	-4.2	172.0	-2.0	-1.1
2990	135.0	123.4	-11.6	-8.6	95.0	-40.0	-29.6
2323	250.0	221.4	-28.6	-11.4	214.0	-36.0	-14.4
2606	101.0	114.5	13.5	13.3	102.0	1.0	0.9
2984	79.0	72.4	-6.6	-8.4	81.0	2.0	2.5
2537	262.0	226.4	-35.6	-13.6	214.0	-48.0	-18.3
1693	140.0	138.0	-2.0	-1.4	118.0	-22.0	-15.7
2465	240.0	222.2	-17.8	-7.4	211.0	-29.0	-12.0
3094	181.0	163.1	-17.9	-9.9	177.0	-4.0	-2.2
3037	134.0	159.0	25.0	18.7	137.0	3.0	2.2
3088	145.0	135.1	-9.9	-6.8	141.0	-4.0	-2.7
2465	230.0	225.8	-4.2	-1.8	211.0	-19.0	-8.2
2301	114.0	152.1	38.1	33.5	157.0	43.0	37.7
1912	176.0	176.4	0.4	0.3	197.0	21.0	11.9
3037	156.0	165.3	9.3	6.0	137.0	-19.0	-12.1
2301	162.0	179.1	17.1	10.5	157.0	-5.0	-3.0
3088	162.0	151.5	-10.5	-6.5	141.0	-21.0	-12.9
2646	233.0	223.8	-9.2	-3.9	211.0	-22.0	-9.4
2465	244.0	226.0	-18.0	-7.4	211.0	-33.0	-13.5
3037	136.0	162.8	26.8	19.7	137.0	1.0	0.7
3000	142.0	199.0	57.0	40.1	152.0	10.0	7.0
2911	77.0	86.0	9.1	11.8	65.0	-12.0	-15.5

2454	54.0	49.1	-4.9	-9.0	45.0	-9.0	-16.6
3141	46.0	77.1	31.1	67.7	52.0	8.0	17.3
2773	46.0	61.5	15.5	33.7	48.0	2.0	4.3
3155	53.0	65.5	12.5	23.5	45.0	-8.0	-15.0
2128	41.0	46.4	5.4	13.2	43.0	2.0	4.8
3132	62.0	64.3	2.3	3.7	81.0	19.0	30.6
3160	30.0	33.6	3.6	11.9	38.0	8.0	26.6
2350	57.0	59.3	2.3	4.0	62.0	5.0	8.7
3028	62.0	65.1	3.1	4.9	64.0	2.0	3.2
2056	70.0	66.5	-3.5	-5.1	72.0	2.0	2.8
3132	82.0	78.6	-3.4	-4.1	81.0	-1.0	-1.2
2989	110.0	90.4	-19.6	-17.8	116.0	6.0	5.4
2350	60.0	61.8	1.8	3.0	62.0	2.0	3.3
1407	80.0	58.6	-21.4	-26.7	88.0	8.0	10.0
2128	49.0	54.6	5.6	11.5	43.0	-6.0	-12.2
3028	97.0	66.6	-30.4	-31.3	64.0	-33.0	-34.0
2055	34.0	47.2	13.2	38.9	44.0	10.0	29.4
2056	61.0	65.8	4.8	7.9	72.0	11.0	18.0
2477	48.0	48.1	0.1	0.2	46.0	-2.0	-4.1
2350	54.0	59.8	5.8	10.8	62.0	8.0	14.8
2100	95.0	78.0	17.0	17.9	67.0	-27.0	-28.4
3028	54.0	58.6	4.6	8.5	64.0	10.0	18.5
2477	47.0	40.4	-6.6	-14.1	46.0	-1.0	-2.1
3141	40.0	74.9	34.9	87.2	52.0	12.0	30.0
2911	68.0	79.4	11.4	16.8	65.0	-3.0	-4.4
2454	51.0	47.7	-3.3	- 6.5	45.0	-6.0	-11.7
2350	51.0	57.5	6.5	12.8	62.0	11.0	21.5
3160	42.0	32.4	-9.6	-23.0	38.0	-4.0	-9.5
2914	69.0	77.6	8.6	12.4	63.0	-6.0	-8.6
3025	117.0	126.0	9.0	7.7	94.0	-23.0	-19.6
2958	62.0	53.3	-8.7	-14.0	46.0	-16.0	-25.8
2983	116.0	120.5	4.5	3.9	91.0	-25.0	-21.5
2999	61.0	45.7	-15.3	-25.1	49.0	-12.0	-19.6
2931	158.0	155.8	-2.2	- 1.4	141.0	-17.0	-10.7
3024	77.0	56.4	-20.6	-26.8	63.0	-14.0	-18.1

2953	160.0	114.9	-45.1	-28.2	85.0	-75.0	-46.8
2995	166.0	148.1	-17.9	-10.8	132.0	-34.0	-20.5
3025	124.0	127.2	3.2	2.6	94.0	-30.0	-24.1
3132	74.0	86.6	12.6	17.0	81.0	7.0	9.4
3029	52.0	63.3	11.3	21.8	49.0	-3.0	-5.7
2915	58.0	60.3	2.3	3.9	63.0	5.0	8.6
3033	144.0	129.5	-14.5	-10.1	115.0	-29.0	-20.1
2966	62.0	70.4	8.4	13.5	62.0	0.0	0.0
3025	123.0	126.9	3.9	3.2	94.0	-29.0	-23.5
2477	59.0	70.7	11.7	19.9	46.0	-13.0	-22.0
2983	116.0	115.5	-0.5	-0.5	91.0	-25.0	-21.5
3158	117.0	144.4	27.4	23.4	134.0	17.0	14.5
2585	97.0	100.0	3.0	3.1	77.0	-20.0	-20.6
3025	117.0	135.0	18.0	15.4	94.0	-23.0	-19.6
2927	62.0	54.2	-7.8	-12.6	61.0	-1.0	-1.6
2911	75.0	96.4	21.4	28.5	65.0	-10.0	-13.3
2915	68.0	63.5	-4.5	-6.6	63.0	-5.0	-7.3
3033	132.0	130.3	-1.7	-1.3	115.0	-17.0	-12.8
2968	140.0	127.1	-12.9	-9.2	116.0	-24.0	-17.1
2958	55.0	47.5	-7.5	-13.6	46.0	-9.0	-16.3
2983	108.0	114.5	6.5	6.0	91.0	-17.0	-15.7
2523	167.0	161.5	-5.5	-3.3	141.0	-26.0	-15.5
2920	174.0	152.1	-21.9	-12.6	142.0	-32.0	-18.3
2931	170.0	159.6	-10.4	-6.1	141.0	-29.0	-17.0
1803	41.0	47.0	6.0	14.6	47.0	6.0	14.6
2088	24.0	29.5	5.5	23.1	40.0	16.0	66.0
2646	50.0	54.3	4.3	8.6	48.0	-2.0	-4.0
3161	26.0	33.1	7.1	27.2	41.0	15.0	57.6
1284	37.0	41.3	4.3	11.5	29.0	-8.0	-21.6
2056	72.0	-57.8	-14.2	19.7	72.0	0.0	0.0
2330	53.0	56.6	3.6	6.8	58.0	5.0	9.4
1285	40.0	40.5	0.5	1.3	36.0	-4.0	-10.0
1133	51.0	53.1	2.1	4.1	68.0	17.0	33.3
1803	53.0	-44.7	-8.3	15.7	47.0	-6.0	-11.3
3140	36.0	39.4	3.4	9.4	45.0	9.0	25.0

3081	57.0	-51.3	-5.7	10.0	54.0	-3.0	-5.2
2077	48.0	-44.0	-4.0	8.4	48.0	0.0	0.0
1989	57.0	57.4	0.4	0.6	52.0	-7.0	-12.2
2690	42.0	-36.8	-5.2	12.4	46.0	4.0	9.5
2560	9.5	-7.7	-1.8	19.0	9.0	-0.5	-5.2
2385	32.0	32.5	0.5	1.7	27.0	-5.0	-15.6
2848	40.0	-38.7	-1.3	3.2	36.0	-4.0	-10.0
2516	26.0	36.7	10.7	41.3	37.0	11.0	42.3
2269	54.0	58.1	4.1	7.6	50.0	-4.0	-7.4
2702	34.0	-33.5	-0.5	1.3	36.0	2.0	5.8
2569	63.0	-58.0	-5.0	7.2	52.0	-11.0	-17.4
2651	37.0	-36.1	-0.9	2.5	24.0	-13.0	-35.1
2269	54.0	58.1	4.1	7.5	50.0	-4.0	-7.4
2625	61.0	-58.1	-2.9	4.7	52.0	-9.0	-14.7
2918	40.0	40.8	0.8	2.0	37.0	-3.0	-7.5
2569	58.0	-57.8	-0.2	0.3	52.0	-6.0	-10.3
2319	16.0	17.2	1.2	7.2	12.0	-4.0	-25.0
2848	41.0	-38.2	-2.8	6.8	36.0	-5.0	-12.1
2702	41.0	-33.3	-7.7	18.7	36.0	-5.0	-12.1
2656	47.0	-41.3	-5.7	12.2	41.0	-6.0	-12.7
2385	30.0	32.9	2.9	9.6	27.0	-3.0	-10.0
2702	36.0	-32.9	-3.1	8.5	36.0	0.0	0.0
2269	60.0	-57.6	-2.4	4.1	50.0	-10.0	-16.6
2385	33.0	32.3	-0.7	-2.1	27.0	-6.0	-18.1
2564	45.0	41.5	-3.5	-7.9	37.0	-8.0	-17.7
2386	34.0	32.6	-1.4	-4.0	27.0	-7.0	-20.5
2625	60.0	58.2	-1.8	-3.0	52.0	-8.0	-13.3
2088	26.0	19.5	-6.5	-25.0	40.0	14.0	53.8
2848	36.0	37.7	1.7	4.7	36.0	0.0	0.0
3095	26.0	35.5	9.5	36.4	36.0	10.0	38.4
2918	74.0	83.0	9.0	12.1	74.0	0.0	0.0
3041	40.0	42.5	2.5	6.3	31.0	-9.0	-22.5
2564	74.0	73.3	-0.7	-1.0	94.0	20.0	27.0
2872	64.0	65.1	1.1	1.7	58.0	-6.0	-9.3
2319	36.0	43.4	7.4	20.6	31.0	-5.0	-13.8

3043	61.0	72.6	11.6	19.1	37.0	-24.0	-39.3
2569	112.0	107.7	-4.3	-3.8	98.0	-14.0	-12.5
3041	41.0	41.0	0.0	0.0	31.0	-10.0	-24.3
2269	115.0	105.7	-9.3	-8.1	94.0	-21.0	-18.2
3043	67.0	74.3	7.3	10.9	37.0	-30.0	-44.7
3044	70.0	73.2	3.2	4.6	37.0	-33.0	-47.1
2625	99.0	107.7	8.7	8.7	94.0	-5.0	-5.0
2872	65.0	64.7	-0.3	-0.4	58.0	-7.0	-10.7
2848	61.0	69.1	8.1	13.3	68.0	7.0	11.4
3102	38.0	33.1	-4.9	-13.0	31.0	-7.0	-18.4
2269	105.0	109.4	4.4	4.2	94.0	-11.0	-10.4
3163	70.0	60.6	-9.4	-13.4	58.0	-12.0	-17.1
2386	69.0	59.6	-9.4	-13.6	50.0	-19.0	-27.5
3057	93.0	88.2	-4.8	-5.1	40.0	-53.0	-56.9
2269	108.0	109.3	1.3	1.2	94.0	-14.0	-12.9
2385	60.0	60.7	0.7	1.1	50.0	-10.0	-16.6
2319	40.0	46.8	6.8	17.1	31.0	-9.0	-22.5
3058	95.0	92.7	-2.3	-2.4	40.0	-55.0	-57.9
2321	78.0	81.6	3.6	4.6	68.0	-10.0	-12.8
2569	114.0	107.1	-6.9	-6.1	98.0	-16.0	-14.0
2918	62.0	83.0	21.0	33.9	74.0	12.0	19.3

Appendix X

Examples of various 'LP' outputs produced by the computerized scheduling system.

FILE OF JOBS TO BE SCHEDULED IN THE FORGE SHOP.

DATE 05/05/72

	JOB NAME	PRI.	QTY REQ.	DD	EPSD	PT	HAMMER CHOICES A	B	C	Z
1	3025	1	7000	09/06/72	**/**/**	14.0	5	0	0	0
2	3083	3	1400	09/06/72	**/**/**	3.5	6	0	0	0
3	3208	3	550	09/06/72	**/**/**	2.0	6	0	0	0
4	2914	1	7000	07/04/72	**/**/**	7.0	6	0	0	0
5	2477	1	1000	05/05/72	**/**/**	3.0	6	0	0	0
6	2478	1	7000	05/05/72	**/**/**	4.0	6	0	0	0
7	13168	1	7000	09/06/72	**/**/**	6.0	6	0	0	0
8	2088	3	200	05/05/72	**/**/**	1.5	7	0	0	0
9	1133	3	800	07/07/72	**/**/**	2.5	7	0	0	0
10	2879	2	350	09/06/72	**/**/**	2.0	7	0	0	0
11	3225	1	5000	08/05/72	**/**/**	9.0	7	0	0	0
12	2648	3	460	05/05/72	**/**/**	1.5	7	0	0	0
13	3205	2	500	09/06/72	**/**/**	2.0	7	0	0	0
14	3196	2	500	09/06/72	**/**/**	1.5	7	0	0	0
15	1708	3	300	09/06/72	**/**/**	1.0	7	0	0	0
16	3102	3	2000	10/04/72	**/**/**	5.0	8	0	0	0
17	2385	2	1000	07/07/72	**/**/**	3.5	8	0	0	0
18	2656	1	700	24/05/72	**/**/**	1.5	8	0	0	0
19	23244	3	10000	05/05/72	**/**/**	10.5	8	0	0	0
20	2379	3	600	05/05/72	**/**/**	2.5	8	0	0	0
21	3206	3	500	14/04/72	**/**/**	2.0	8	0	0	0
22	3043	1	7000	09/06/72	**/**/**	10.0	8	0	0	0
23	3044	1	7000	09/06/72	**/**/**	10.0	8	0	0	0
24	2918	1	7000	09/06/72	**/**/**	8.0	9	0	0	0
25	2912	1	7000	09/06/72	**/**/**	8.0	9	0	0	0

FORGE SHOP SCHEDULE.

HAMMER 7

	JOB.	QTY REQ	BALANCE	EPSD	DD	LSSD		LSFD		PT	PRI.	HAMMER CHOICES A	B	C	Z
1	23168	7000	150	08/05/72	29/02/72	04/05/72	1	09/05/72	1	3.0	3	7	0	0	0
2	2088	200	200	**/**/**	05/05/72	09/05/72	1	10/05/72	2	1.5	3	7	0	0	0
3	2648	460	460	**/**/**	05/05/72	10/05/72	2	12/05/72	1	1.5	3	7	0	0	0
4	3225	5000	5000	**/**/**	08/05/72	12/05/72	1	25/05/72	1	9.0	1	7	0	0	0
5	2879	350	350	**/**/**	09/06/72	25/05/72	1	29/05/72	1	2.0	2	7	0	0	0
6	3205	500	500	**/**/**	09/06/72	29/05/72	1	31/05/72	1	2.0	2	7	0	0	0
7	3196	500	500	**/**/**	09/06/72	31/05/72	1	01/06/72	2	1.5	2	7	0	0	0
8	1708	300	300	**/**/**	09/06/72	01/06/72	2	02/06/72	2	1.0	3	7	0	0	0
9	1133	800	800	**/**/**	07/07/72	02/06/72	2	07/06/72	1	2.5	3	7	0	0	0

HAMMER 8

	JOB.	QTY REQ	BALANCE	EPSD	DD	LSSD		LSFD		PT	PRI.	HAMMER CHOICES A	B	C	Z
1	13244	10000	200	08/05/72	07/04/72	21/04/72	1	08/05/72	1	10.5	3	8	0	0	0
2	3206	500	500	**/**/**	14/04/72	08/05/72	1	09/05/72	1	2.0	3	8	0	0	0
3	2379	600	600	**/**/**	05/05/72	09/05/72	1	10/05/72	2	2.5	3	8	0	0	0
4	3102	2000	2000	**/**/**	10/04/72	10/05/72	2	12/05/72	4	5.0	3	8	0	0	0
5	23244	10000	10000	**/**/**	05/05/72	12/05/72	4	22/05/72	1	10.5	3	8	0	0	0
6	2656	700	700	**/**/**	24/05/72	22/05/72	1	22/05/72	4	1.5	1	8	0	0	0
7	3043	7000	7000	**/**/**	09/06/72	22/05/72	4	29/05/72	4	10.0	1	8	0	0	0
8	3044	7000	7000	**/**/**	09/06/72	29/05/72	4	05/06/72	4	10.0	1	8	0	0	0
9	2335	1000	1000	**/**/**	07/07/72	05/06/72	4	07/06/72	3	3.5	2	8	0	0	0

JOBS LIKELY TO BE LATE.

HAMMER 1

JOB NO.	SCHED. FINISH DATE	DAYS LATE (UNWEIGHTED)	PRI.
729	12/05/72	-5	3
1567	15/05/72	-6	3
3248	16/05/72	-7	3
2964	25/05/72	-1	1

HAMMER 2

JOB NO.	SCHED. FINISH DATE	DAYS LATE (UNWEIGHTED)	PRI.
2256	12/05/72	-12	1
3087	16/05/72	-6	3

HAMMER 3

JOB NO.	SCHED. FINISH DATE	DAYS LATE (UNWEIGHTED)	PRI.
3259	09/05/72	-2	3
2301	12/05/72	-15	2

DAILY REPORT OF FORGE SHOP PRODUCTION.

DATE
02/01/72

B NO. 1001 COMPLETED, ON HAMMER NO, 1 PUNCTUALITY 5

B NO. 1004 ON HAMMER NO. 1 REMOVED FROM FURTHER CONSIDERATION(JOB CANCELLED,MATL. UNAVAILABLE E.T.C.)

B NO. 2001 ON HAMMER NO, 2 NOT STARTED/CONTINUED AS PER SCHEDULE, REPLACED BY JOB NO. 4005

HAMMER NO, 6 FAILED!!! E.R.D. IS 19/01/72

HAMMER NO, 7 FAILED!!! E.R.D. IS 01/02/72

B NO. 9001 COMPLETED, ON HAMMER NO, 9 PUNCTUALITY 10

3058	22/09/72	##/##/##	1																		
				1	1.5	2	1.5	5	0.5	7	6.0	12	2.0								
				0	0.0	0	0.0	0	0.0	8	6.0	13	2.0								
				0	0.0	0	0.0	0	0.0	0	0.0	14	2.0								
				0	0.0	0	0.0	0	0.0	0	0.0	15	2.0								

NEXT:-	SEQ.	M/C	PT.	OP.6		OP.7		OP.8		OP.9		OP.10	
	2	2	1.5	11	1.0	0	0.0	0	0.0	0	0.0	0	0.0
				10	1.0	0	0.0	0	0.0	0	0.0	0	0.0
				0	0.0	0	0.0	0	0.0	0	0.0	0	0.0
				0	0.0	0	0.0	0	0.0	0	0.0	0	0.0

JOB.	DD	EPSD	PRI.	OP.1		OP.2		OP.3		OP.4		OP.5	
2872	01/09/72	01/09/72	1	6	2.0	4	0.5	5	0.5	14	2.0	17	0.5
				0	0.0	0	0.0	0	0.0	12	2.0	16	0.5
				0	0.0	0	0.0	0	0.0	13	2.0	0	0.0
				0	0.0	0	0.0	0	0.0	15	2.0	0	0.0

NEXT:-	SEQ.	M/C	PT.	OP.6		OP.7		OP.8		OP.9		OP.10	
	2	4	0.5	0	0.0	0	0.0	0	0.0	0	0.0	0	0.0
				0	0.0	0	0.0	0	0.0	0	0.0	0	0.0
				0	0.0	0	0.0	0	0.0	0	0.0	0	0.0
				0	0.0	0	0.0	0	0.0	0	0.0	0	0.0

JOB.	DD	EPSD	PRI.	OP.1		OP.2		OP.3		OP.4		OP.5	
3160	29/09/72	10/07/72	1	14	3.0	18	1.0	0	0.0	0	0.0	0	0.0
				0	0.0	19	1.0	0	0.0	0	0.0	0	0.0
				0	0.0	0	0.0	0	0.0	0	0.0	0	0.0
				0	0.0	0	0.0	0	0.0	0	0.0	0	0.0

NEXT:-	SEQ.	M/C	PT.	OP.6		OP.7		OP.8		OP.9		OP.10	
	2	18	1.0	0	0.0	0	0.0	0	0.0	0	0.0	0	0.0
				0	0.0	0	0.0	0	0.0	0	0.0	0	0.0
				0	0.0	0	0.0	0	0.0	0	0.0	0	0.0
				0	0.0	0	0.0	0	0.0	0	0.0	0	0.0

MACHINE
17

JOB.	OPERATION	LSSD		LSFD		BEGUN?
3286	1	18/09/72	1	19/09/72	1	NO
3037	1	19/09/72	1	20/09/72	1	
1930	2	20/09/72	1	21/09/72	1	
2872	5	22/09/72	1	22/09/72	2	
2704	1	22/09/72	2	26/09/72	2	
3276	4	26/09/72	2	28/09/72	2	

MACHINE
18

JOB.	OPERATION	LSSD		LSFD		BEGUN?
3273	1	18/09/72	1	20/09/72	1	NO
3160	2	20/09/72	1	21/09/72	1	
12523	2	21/09/72	1	25/09/72	1	
3283	3	25/09/72	1	27/09/72	1	
2918	3	27/09/72	1	04/10/72	2	
3027	3	04/10/72	2	06/10/72	2	
3279	2	06/10/72	2	10/10/72	2	

MACHINE
19

JOB.	OPERATION	LSSD		LSFD		BEGUN?
2918	2	25/09/72	1	26/09/72	1	NO
3102	1	26/09/72	1	28/09/72	1	
3088	2	28/09/72	1	29/09/72	1	
3282	5	29/09/72	2	03/10/72	2	
3145	1	03/10/72	2	06/10/72	2	
2915	3	06/10/72	2	09/10/72	2	
1803	3	09/10/72	2	10/10/72	2	

DIE SHOP SCHEDULE, OUTPUT IN JOB ORDER.

JOB.	OP.	M/C.	LSSD.		LSFD.		REQ.DD.	BEST DD.	PUNCT.
2965	1	5	18/09/72	2	19/09/72	1			
	2	8	22/09/72	1	25/09/72	1			
	3	15	25/09/72	1	26/09/72	1			
	4	16	27/09/72	1	27/09/72	2	31/10/72	28/09/72	23
3013	1	10	22/09/72	1	25/09/72	1	29/09/72	25/09/72	1
3025	1	16	22/09/72	1	27/09/72	1			
	2	12	27/09/72	1	29/09/72	1			
	3	10	29/09/72	1	02/10/72	1	03/10/72	02/10/72	1
3027	1	8	20/09/72	1	22/09/72	1			
	2	12	26/09/72	1	27/09/72	1			
	3	18	04/10/72	2	06/10/72	2	31/10/72	09/10/72	16
3028	1	14	22/09/72	1	26/09/72	1			
	2	10	26/09/72	1	28/09/72	1	29/09/72	28/09/72	1
3029	1	10	19/09/72	1	21/09/72	1	16/08/72	21/09/72	-9
3056	1 *	1	15/09/72	1	18/09/72	2			
	2	2	19/09/72	1	20/09/72	2			
	3	5	21/09/72	1	21/09/72	2			
	4	7	21/09/72	2	29/09/72	2			
	5	15	29/09/72	2	03/10/72	2			
	6	10	03/10/72	2	04/10/72	2	22/09/72	05/10/72	-9

DIES LIKELY TO BE LATE.

JOB NO.	SCHED. FINISH DATE	DAYS LATE (UNWEIGHTED)
2872	25/09/72	-16
2918	05/10/72	-17
3029	21/09/72	-26
3036	05/10/72	-9
3037	26/09/72	-7
3160	21/09/72	-53
3168	20/09/72	-14
3273	20/09/72	-14
3280	20/09/72	-31
3282	04/10/72	-31
3283	27/09/72	-26
3286	19/09/72	-22
12523	28/09/72	-9
22523	25/09/72	-6

DAILY REPORT OF DIE SHOP PRODUCTION.

DATE 18/01/72

JOB. NO. 9917 OPERATION NO. 1 STARTED ON M/C 2

JOB. NO. 9912 OPERATION NO. 1 STARTED ON M/C 3

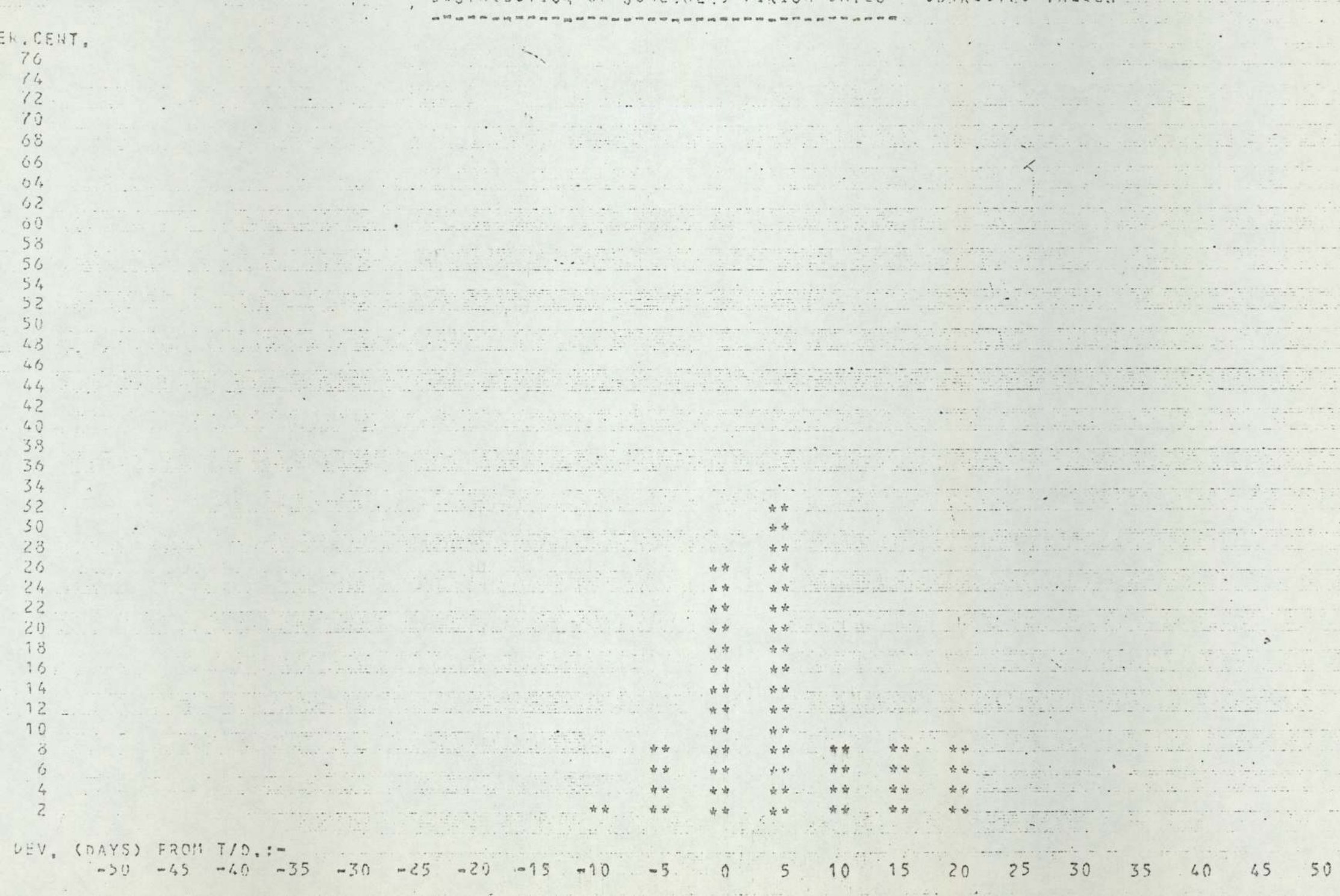
M/C 5 BROKEN DOWN, E.R.D. IS 11/02/72

JOB. NO. 1003 OPERATION NO. 2 STARTED ON M/C 6

JOB. NO. 1007 OPERATION NO. 2 STARTED ON M/C 7

JOB. NO. 1006 OPERATION NO. 3 NOT STARTED ON M/C 8 AS PER SCHEDULE, REPLACED BY JOB NO. 1002

JOB. NO. 1004 OPERATION NO. 1 NOT STARTED ON M/C 10 AS PER SCHEDULE, REPLACED BY JOB NO. 9916


EK. CENT.
76
74
72
70
68
66
64
62
60
58
56
54
52
50
48
46
44
42
40
38
36
34
32
30
28
26
24
22
20
18
16
14
12
10
8
6
4
2
DEV. (DAYS) FROM T/O.:-
-50 -45 -40 -35 -30 -25 -20 -15 -10 -5 0 5 10 15 20 25 30 35 40 45 50

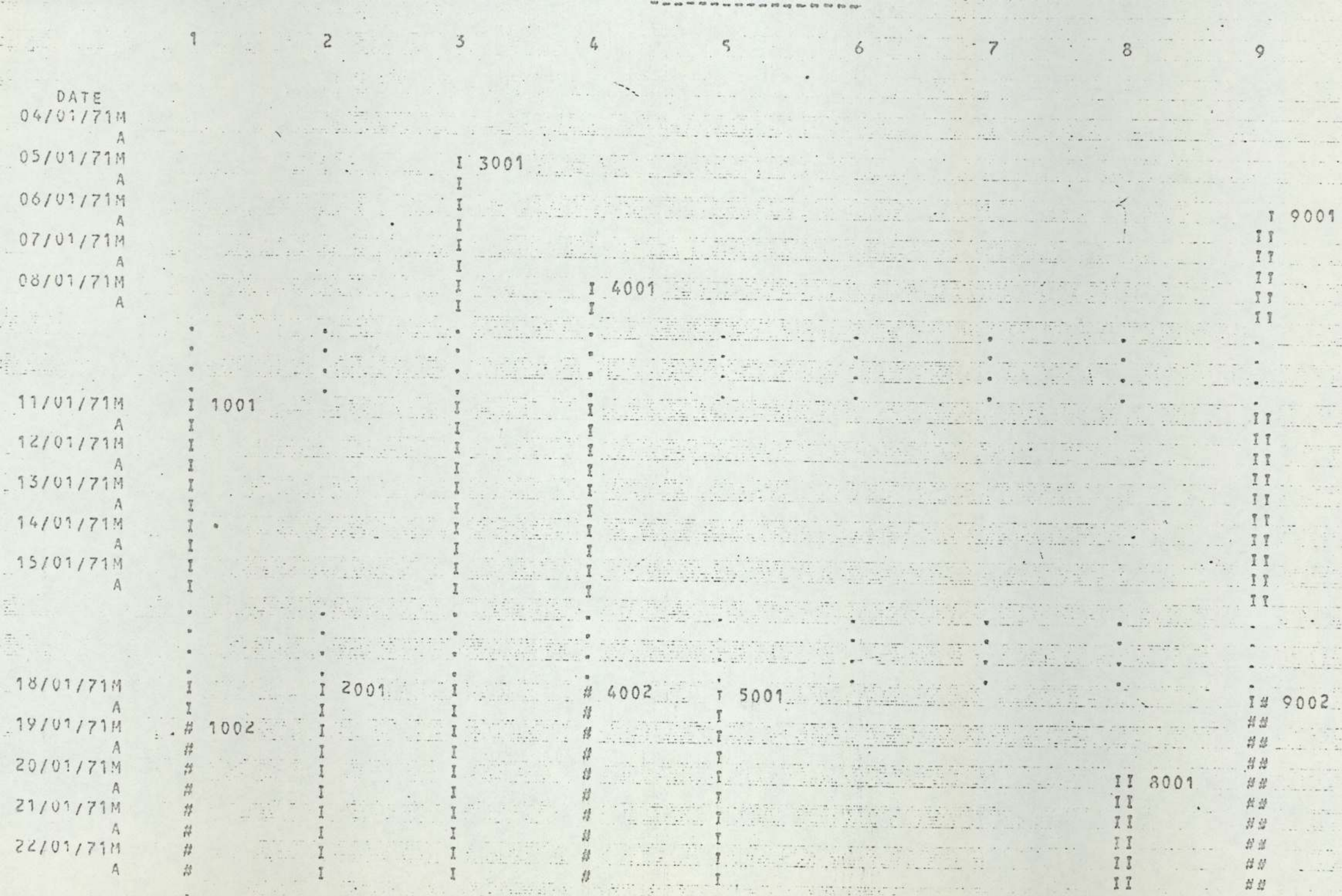
PERFORMANCE ANALYSIS (Continued from previous page.)

EV. (DAYS) FROM T/D.:-	-50	-45	-40	-35	-30	-25	-20	-15	-10	-5	0	5	10	15	20	25	30	35	40	45	50
RUENCY DISTRIBUTION :-	0	0	0	0	0	0	0	0	1	3	9	11	3	3	3	0	0	0	0	0	0
DITTO (PERCENT) :-	0	0	0	0	0	0	0	0	3	9	27	33	9	9	9	0	0	0	0	0	0
UMALATIVE FREQUENCY :-	0	0	0	0	0	0	0	0	1	4	8	22	27	28	33	33	33	33	33	33	33
DITTO (PERCENT) :-	0	0	0	0	0	0	0	0	3	12	24	67	82	85	100	100	100	100	100	100	100

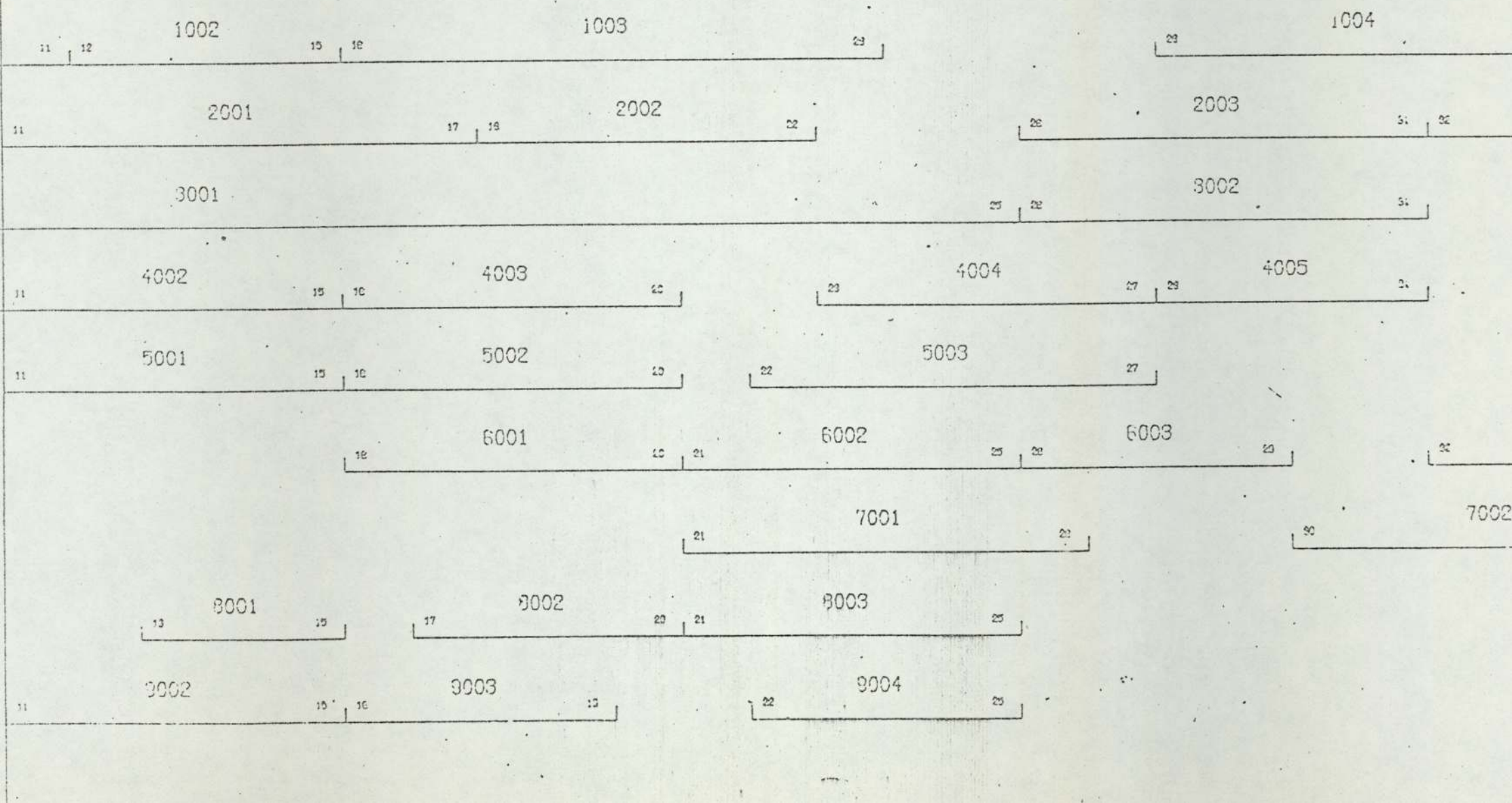
MEAN TARDINESS 5

M/C UTILIZATION PERCENT

1	2	3	4	5	6	7	8	9
76	100	89	100	94	83	100	93	87


1
2
3
4
5
6
7
8
9
DATE
04/01/71M
A
05/01/71M
A
06/01/71M
A
07/01/71M
A
08/01/71M
A
11/01/71M
A
12/01/71M
A
13/01/71M
A
14/01/71M
A
15/01/71M
A
18/01/71M
A
19/01/71M
A
20/01/71M
A
21/01/71M
A
22/01/71M
A
3001
4001
1001
2001
4002
5001
9001
9002
1002
8001

GANTT-CHART PRESENTATION ON AN x-y PLOTTER.



Appendix XI

Listing of computer programme segment for producing Gantt-chart representation of schedules.

```
'BEGIN'
  'INTEGER'YES,ORIGIN,CUDATE,CUSHIFT,H,N,D,NOTE,ONO,STARTED,X,DUMMY,Y,
  DWND,DWN,MAXDWN,FLAGYEAR,AD,BD,DDATE,MDATE,YDATE;
  'INTEGER''ARRAY'NO[1:9],SHIFTWORK[1:9],NN[1:9],FINISH[1:9],FM[71:80];
  'REAL''ARRAY'NAMFILE[1:9,1:20,1:12];
'PROCEDURE' DAY2MONTH(DWN,YEAR);
'INTEGER'YEAR,DWN;
'BEGIN'
     'INTEGER'LY,ITRAD,TRAD;
     'REAL'RTRAD,REM;
      RTRAD:=DWN/5;
      ITRAD:=ENTIER(RTRAD+0.000001);
      REM:=ABS(RTRAD-ITRAD);
      'IF'REM'GT'0.00001'THEN'REM:=(REM*5)+2
      'ELSE'
      REM:=0;
      TRAD:=(ITRAD*7)+REM;
      TRAD:=TRAD+(FM[YEAR1-3);
      'IF'YEAR/4=ENTIER(YEAR/4)'THEN'LY:=1
      'ELSE'
      LY:=0;
      'IF'TRAD'LE'31'THEN'
                        'BEGIN'
                           DDATE:=TRAD-0;
                           MDATE:=1;
                        'END'
                        'ELSE'
      'IF'TRAD'LE'59+LY'THEN'
                        'BEGIN'
                           DDATE:=TRAD-31;
                           MDATE:=2;
                        'END'
                        'ELSE'
      'IF'TRAD'LE'90+LY'THEN'
                        'BEGIN'
```

```
                    DDATE:=TRAD-59-LY;
                    MDATE:=3;
                'END'
                'ELSE'
'IF'TRAD'LE'120+LY'THEN'
                'BEGIN'
                    DDATE:=TRAD-90-LY;
                    MDATE:=4;
                'END'
                'ELSE'
'IF'TRAD'LE'151+LY'THEN'
                'BEGIN'
                    DDATE:=TRAD-120-LY;
                    MDATE:=5;
                'END'
                'ELSE'
'IF'TRAD'LE'181+LY'THEN'
                'BEGIN'
                    DDATE:=TRAD-151-LY;
                    MDATE:=6;
                'END'
                'ELSE'
'IF'TRAD'LE'212+LY'THEN'
                'BEGIN'
                    DDATE:=TRAD-181-LY;
                    MDATE:=7;
                'END'
                'ELSE'
'IF'TRAD'LE'243+LY'THEN'
                'BEGIN'
                    DDATE:=TRAD-212-LY;
                    MDATE:=8;
                'END'
                'ELSE'
'IF'TRAD'LE'274+LY'THEN'
```

```
                    'BEGIN'
                       DDATE:=TRAD-243-LY;
                       MDATE:=9;
                    'END'
                    'ELSE'
          'IF'TRAD'LE'304+LY'THEN'
                    'BEGIN'
                       DDATE:=TRAD-274-LY;
                       MDATE:=10;
                    'END'
                    'ELSE'
          'IF'TRAD'LE'334+LY'THEN'
                    'BEGIN'
                       DDATE:=TRAD-304-LY;
                       MDATE:=11;
                    'END'
                    'ELSE'
                    'BEGIN'
                       DDATE:=TRAD-334-LY;
                       MDATE:=12;
                    'END';
'END';
'PROCEDURE' OUTDATE(DWND);
'INTEGER'DWND;
'BEGIN'
      'IF'DWND =0'THEN'
            'BEGIN'
               WRITETEXT('('**/**/**')');
               'GOTO' SKIP2;
            'END';
      'IF'DWND'GT'MAXDWN'THEN'
           'BEGIN'
              YDATE:=FLAGYEAR+1;
              DWN:=DWND-MAXDWN;
           'END'
```

```
'ELSE'
'BEGIN'
   YDATE:=FLAGYEAR;
   DWN:=DWND;
'END';
DAY2MONTH(DWN,YDATE);
'IF'MDATE=12'THEN'
    'BEGIN'
      'IF'DDATE>31'THEN'
            'BEGIN'
              MDATE:=1;
              DDATE:=DDATE-31;
              YDATE:=FLAGYEAR+1;
            'END';
    'END';
'IF'DDATE'GE'30'THEN'
            'BEGIN'
              AD:=3;
              BD:=DDATE-30;
            'END'
            'ELSE'
'IF'DDATE'GE'20'THEN'
            'BEGIN'
              AD:=2;
              BD:=DDATE-20;
            'END'
            'ELSE'
'IF'DDATE'GE'10'THEN'
            'BEGIN'
              AD:=1;
              BD:=DDATE-10;
            'END'
            'ELSE'
            'BEGIN'
              AD:=0;
```

```
                BD:=DDATE;
              'END';
CODE('('AD')');
CODE('('BD')');
PRINTCH(AD);
PRINTCH(BD);
WRITETEXT('('/')');
'IF'MDATE'GE'10'THEN'
              'BEGIN'
                AD:=1;
                BD:=MDATE-10;
              'END'
              'ELSE'
              'BEGIN'
                AD:=0;
                BD:= MDATE;
              'END';
CODE('('AD')');
CODE('('BD')');
PRINTCH(AD);
PRINTCH(BD);
WRITETEXT('('/')');
'IF'YDATE'GE'80'THEN'
              'BEGIN'
                AD:=8;
                BD:=YDATE-80;
              'END'
              'ELSE'
              'BEGIN'
                AD:=7;
                BD:=YDATE-70;
              'END';
CODE('('AD')');
CODE('('BD')');
PRINTCH(AD);
```

```
      PRINTCH(RD);
SKIP2:
'END';
'PROCEDURE'CHECKMULT(X,Y);
'INTEGER'X,Y;
'BEGIN'
  'INTEGER'INTGN;
  'REAL'REALN,DIFF;
     REALN:=X/Y+0.000001;
     INTGN:=ENTIER(REALN);
     DIFF:=ABS(REALN-INTGN);
     'IF'DIFF'LE'0.00001'THEN'YES:=1
     'ELSE'
     YES:=0;
'END';
'COMMENT' READ IN FILES TO BE PLOTED AND TRANSFORM FROM EXT.TO INT.CODE;
     ORIGIN:=READ;
     'FOR'H:=1'STEP'1'UNTIL'9'DO'
       'BEGIN'
         SHIFTWORK[H]:=READ;
         NO[H]:=READ;
         DUMMY:=READ;
         DUMMY:=READ;
         'FOR'N:=1'STEP'1'UNTIL'NO[H]'DO'
               'BEGIN'
            'FOR'D:=1'STEP'1'UNTIL'12'DO'
                  HAMFILE[H,N,D]:=READ;
                  'IF'SHIFTWORK[H]=1'THEN'
                    'BEGIN'
                      'IF'HAMFILE[H,N,7]=2'THEN'HAMFILE[H,N,7]:=3;
                      'IF'HAMFILE[H,N,9]=2'THEN'HAMFILE[H,N,9]:=3;
                    'END';
               'END';
       'END';
     'FOR'H:=1'STEP'1'UNTIL'9'DO'
```

```
        'BEGIN'
           NN[H]:=1;
           FINISH[H]:=0;
        'END';
       CUSHIFT:=1;
       CUDATE:=ORIGIN;
       NOTE:=9;
       'COMMENT' IN PRACTICE , FOLLOWING TWO LINES WOULD NOT BE REQ. ;
       MAXDWN:=260;
       FLAGYEAR:=71;
      FM[71]:=4;
      FM[72]:=3;
      FM[73]:=1;
      FM[74]:=7;
      FM[75]:=6;
      FM[77]:=3;
      FM[78]:=2;
      FM[79]:=1;
      FM[80]:=7;
       PAPERTHROW;
       WRITETEXT('('''('55S')'UNIT%NUMBER.'('C')''('52S')'-----------'('2
C')''('12S')'')');
       'FOR'N:=1'STEP'1'UNTIL'9'DO'
         'BEGIN'
           PRINT(N,2,0);
           SPACE(6);
         'END';
       WRITETEXT('('''('3C')'%%%DATE'('C')'')');
AD1:   H:=0;
       'IF'CUSHIFT'LE'2'THEN'
        'BEGIN'
          OUTDATE(CUDATE);
          WRITETEXT('('M')');
        'END'
       'ELSE'
```

```
      WRITETEXT('('%%%%%%%%A')');
      SPACE(5);
AD2:  H:=H+1;
        STARTED:=0;
AD3:  'COMMENT' IS THIS THE LAST JOB ON THIS UNIT ?***                    ****;
      'IF'NN[H]>NO[H]'THEN'
        'BEGIN'
          FINISH  [H]:=FINISH  [H]+1;
          'IF'FINISH  [H]=1'THEN'NOTE:=NOTE-1;
          'IF'NOTE'LE'0'THEN''GOTO'STOPP;
          CHECKMULT(CUSHIFT,2);
          'IF'YES=1'THEN'
            'BEGIN'
              CUSHIFT:=CUSHIFT-1;
              SPACE(10);
            'END'
          'ELSE'
          SPACE(11);
          'GOTO'FOCUS;
        'END';
AD4:  'COMMENT' HAS THE PRESENT JOB FINISHED ON,OR BEFORE, TODAY?*******;
      'IF'HAMFILE[H,NN[H],8]<CUDATE'THEN''GOTO'A1LP;
      'IF'HAMFILE[H,NN[H],8]=CUDATE'THEN'
        'BEGIN'
          'IF'HAMFILE[H,NN[H],9]'LE'CUSHIFT'THEN'
A1LP:       'BEGIN'
              NN[H]:=NN[H]+1;
              'GOTO'AD3;
            'END';
        'END';
      'COMMENT' CAN THE NEXT JOB BE STARTED BY TODAY? ****          ******;
      'IF'HAMFILE[H,NN[H],6]>CUDATE'THEN''GOTO'A2LP;
      'IF'HAMFILE[H,NN[H],6]=CUDATE'THEN'
        'BEGIN'
          'IF'HAMFILE[H,NN[H],7]>CUSHIFT'THEN'
```

```
A2LP:       'BEGIN'
               SPACE(1);
               'IF'SHIFTWORK[H]=2'THEN'
                 'BEGIN'
                    CHECKMULT(CUSHIFT,2);
                    'IF'YES=1'THEN''GOTO'EVEN1
                    'ELSE'
                    CUSHIFT:=CUSHIFT+1;
                    'GOTO'AD4;
                 'END';
               SPACE(10);
               'GOTO'FOCUS
            'END';
         'END';
      'COMMENT' HAS THIS JOB STARTED TODAY ? ******                    ****;
      'IF'HAMFILE[H,NN[H],6]=CUDATE'THEN'
         'BEGIN'
           'IF'HAMFILE[H,NN[H],7]=CUSHIFT'THEN'
            'BEGIN'
               ONO:=NN[H]/2-0,01;
               'IF'ONO=NN[H]/2'THEN'WRITETEXT('('#')')
               'ELSE'
               WRITETEXT('('I')');
               'IF'SHIFTWORK[H]=2'THEN'
                 'BEGIN'
                    CHECKMULT(CUSHIFT,2);
                    'IF'YES=1'THEN''GOTO'EVEN2
                    'ELSE'
                    CUSHIFT:=CUSHIFT+1;
                    STARTED:=1;
                    'GOTO'AD4;
                 'END';
               PRINT(HAMFILE[H,NN[H],1],4,0);
               SPACE(3);
               'GOTO'FOCUS;
```

```
            'END';
         'END';
      'COMMENT' THEN THIS JOB MUST BE IN PROGRESS TODAY !!  **      *****;
      ONO:=NN[H]/2-0.01;
      'IF'ONO=NN[H]/2'THEN'WRITETEXT('('#')')
      'ELSE'
      WRITETEXT('('T')');
      'IF'SHIFTWORK[H]=2'THEN'
         'BEGIN'
            CHECKMULT(CUSHIFT,2);
            'IF'YES=1'THEN''GOTO'EVEN3
            'ELSE'
            CUSHIFT:=CUSHIFT+1;
            'GOTO'AD4;
         'END';
      SPACE(10);
      'COMMENT'CONCIDER NEXT UNIT AND/OR TIME PERIOD. *****       ******;
FOCUS:'IF'H=9'THEN'
         'BEGIN'
           'IF'CUSHIFT'LE'2'THEN'
                 WRITETEXT('('%%%%N')')
           'ELSE'
           WRITETEXT('('%%%%A')');
           NEWLINE(1);
           'IF'CUSHIFT'GT'2'THEN'
              'BEGIN'
                 CHECKMULT(CUDATE,5);
                 'IF'YES=1'THEN'
                  'FOR'Y:=1'STEP'1'UNTIL'4'DO'
                    'BEGIN'
                       SPACE(14);
                       'FOR'X:=1'STEP'1'UNTIL'9'DO'
                         WRITETEXT('('.'('10S')'')');
                         NEWLINE(1);
                    'END';
```

```
            'END';
          CUSHIFT:=CUSHIFT+2;
          'IF'CUSHIFT=5'THEN'
            'BEGIN'
              CUDATE:=CUDATE+1;
              CUSHIFT:=1;
            'END';
          'GOTO'AD1;
        'END'
      'ELSE'
      'GOTO'AD2;
EVEN1;SPACE(9);
      CUSHIFT:=CUSHIFT-1;
      'GOTO'FOCUS;
EVEN2:
      PRINT(NAMFILE[H,NN[H],1],4,0);
      SPACE(2);
      CUSHIFT:=CUSHIFT-1;
      'GOTO'FOCUS;
EVEN3:
      'IF'STARTED=1'THEN'
          'BEGIN'
            STARTED:=0;
            PRINT(NAMFILE[H,NN[H],1],4,0);
            SPACE(2);
          'END'
      'ELSE'
      SPACE(9);
      CUSHIFT:=CUSHIFT-1;
      'GOTO'FOCUS;
STOPP:
     PAPERTHROW;
'COMMENT'RECODE SHIFT FROM INT.TO EXT.FORM.*****                    ******;
     'FOR'H:=1'STEP'1'UNTIL'9'DO'
       'FOR'N:=1'STEP'1'UNTIL'NO[H]'DO'
```

```
            'BEGIN'
              'IF'SHIFTWORK[H]=1'THEN'
                'BEGIN'
                  'IF'HAMFILE[H,N,7]=3'THEN'HAMFILE[H,N,7]:=2;
                  'IF'HAMFILE[H,N,9]=3'THEN'HAMFILE[H,N,9]:=2;
                'END';
            'END';
'END'
```

```
'BEGIN'
     'INTEGER' H,N,D,CUDATE,DUMMY,ORIGDATE,YES,X,JOB;
     'INTEGER''ARRAY'NO[1:9],HAMFILE[1:9,1:10,1:8];
     'REAL'XS,XF,XMID;
     'PROCEDURE' OPENPLOT; 'EXTERNAL';
     'PROCEDURE' CLOSEPLOT; 'EXTERNAL';
     'PROCEDURE' HGPLOT(X,Y,A,B); 'REAL' X,Y; 'INTEGER' A,B; 'EXTERNAL';
     'PROCEDURE' HGPNUMBER(X,Y,HT,FL,THETA,I,IP,IQ); 'VALUE' X,Y,HT,
       FL,THETA,I,IP,IQ; 'INTEGER' I,IP,IQ; 'REAL' X,Y,HT,FL,THETA;
       'EXTERNAL';
         CUDATE:=READ;
         'FOR'H:=1'STEP'1'UNTIL'9'DO'
         'BEGIN'
                   NO[H]:=READ;
                   DUMMY:=READ;
                   DUMMY:=READ;
                   DUMMY:=READ;
                   DUMMY:=READ;
                   'FOR'N:=1'STEP'1'UNTIL'NO[H]'DO'
                        'FOR'D:=1'STEP'1'UNTIL'8'DO'
                        HAMFILE[H,N,D]:=READ;
         'END';
       ORIGDATE:=CUDATE;
       OPENPLOT;
       YES:=0;
       'GOTO'BBB;
FALSE:   DUMMY:=999;
BBB:   HGPLOT(-10.0,12.0,1,4);
       HGPLOT(0.0,-1.0,3,0);
       HGPLOT(0.0,10.0,2,0);
       HGPNUMBER(-0.2,-1.2,0.1,ORIGDATE,0,0,0,3,0);
       X:=-1;
       YES:=YES+1;
AAA:   X:=X+1;
       H:=9-X;
```

```
        HGPNUMBER(-7.0,0.0,0.1,H,0.0,0,2,0);
        'FOR'N:=1'STEP'1'UNTIL'NO[H]'DO'
        'BEGIN'
              XS:=HAMFILE[H,N,7]-ORIGDATE-1;
              XS:=XS/2;
              XF:=HAMFILE[H,N,8]-ORIGDATE;
              XF:=XF/2;
              HGPLOT(XS,0.1,3,0);
              HGPLOT(XS,0.0,2,0);
              HGPLOT(XF,0.0,2,0);
              HGPLOT(XF,0.1,2,0);
              HGPNUMBER(XS,0.1,0.05,HAMFILE[H,N,7],0.0,0,3,0);
              HGPNUMBER(XF-0.3,0.1,0.05,HAMFILE[H,N,8],0.0,0,3,0);
              XMID:=(XF-XS)/2+XS-0.3;
              HGPNUMBER(XMID,0.2,0.1,HAMFILE[H,N,1],0.0,0,4,0);
        'END';
        HGPLOT(0.0,0.0,3,0);
        HGPLOT(0.0,-0.6,1,4);
        'IF'H>1'THEN''GOTO'AAA'ELSE'
        'IF'YES=1'THEN'
           'BEGIN'
                HGPLOT(-10.0,-8.0,3,0);
                'GOTO'FALSE;
           'END'
        'ELSE'
        CLOSEPLOT;
'END'
```

Appendix XII

Listing of computer programme segment for producing performance-analysis of a schedule.

```
'BEGIN'
  'INTEGER'LINA,DAY,SUB,LEVEL;
  'REAL'UPPER,LOWER;
  'REAL''ARRAY'HLZNT[1,20],PCUL[-10:10,1:2],PNODAY[-10:10,1:2],
MEANTARD[1:2];
  'INTEGER''ARRAY'NODAY[-10:10,1:2],CUL[-10:10,1:2],SUM[1:2];
'PROCEDURE'HCUT(CODE,RAYS);
'INTEGER'CODE;
'REAL''ARRAY'RAYS;
 'BEGIN' 'INTEGER'P,Q,R;      'INTEGER''ARRAY'LIMB[1:20];
                'REAL'GAP,DGAP,SGAP,FINS,FRAC;
    'IF'CODE=1'THEN'
      'BEGIN'
        LINA:=9;
        P:=6;
        Q:=7;
        R:=8;
        S:=9;
        'FOR'R:=1'STEP'1'UNTIL'LINA'DO'
          LIMB[R]:=NO[R];
      'END'
    'ELSE'
      'BEGIN'
        LINA:=10;
        P:=3;
        Q:=4;
        R:=5;
        S:=6;
        'FOR'R:=1'STEP'1'UNTIL'LINA'DO'
          LIMB[R]:=DNO[R];
      'END';
    'FOR'R:=1'STEP'1'UNTIL'LINA'DO'
      'BEGIN'
       'IF'SRTSTWORK[R]=1'THEN'FRAC:=0.50
       'ELSE'FRAC:=0.25;
```

```
            GAP:=0;
            'FOR'N:=2'STEP'1'UNTIL'LIMB[H]'DO'
              'BEGIN'
                DGAP:=RAYS[H,N,P]-RAYS[H,N-1,R];
                SGAP:=RAYS[H,N,Q]-RAYS[H,N-1,S];
                SGAP:=SGAP*FRAC;
                DGAP:=SGAP+DGAP;
                GAP:=GAP+DGAP;
                FINS:=RAYS[H,N,S]-1;
                FINS:=FINS*FRAC;
                FINS:=RAYS[H,N,P]+FINS;
              'END';
            ULZU[H]:=(((FINS-(CUDATE+1))-GAP)/(FINS-(CUDATE+1)))*100;
          'END';
      'END';
        DAY:=-55;
A14:    DAY:=DAY+5;
        SUB:=DAY/5;
        NODAY[SUB,1]:=NODAY[SUB,2]:=SUM[1]:=SUM[2]:=0;
        CUL[SUB,1]:=CUL[SUB,2]:=0;
        'IF'DAY=-50'THEN'LOWER:=-999
        'ELSE'
        LOWER:=DAY-2.5000;
        'IF'DAY=50'THEN'UPPER:=999
        'ELSE'
        UPPER:=DAY+2.5000;
        'FOR'S:=1'STEP'1'UNTIL'TT'DO'
          'BEGIN'
            'IF'LATES[S,2]'LE'DAY'THEN'CUL[SUB,1]:=CUL[SUB,1]+1;
            'IF'LATES[S,3]'LE'DAY'THEN'CUL[SUB,2]:=CUL[SUB,2]+1;
            'IF'LATES[S,2]'LE'UPPER'THEN'
              'BEGIN'
                'IF'LATES[S,2]'GT'LOWER'THEN'NODAY[SUB,1]:=NODAY[SUB,1]
                                                          +1;
              'END';
```

```
            'IF'DATES[S,3]'LE'UPPER'THEN'
              'BEGIN'
                'IF'DATES[S,3]'GT'LOWER'THEN'NODAY[SUB,2]:=NODAY[SUB,2]
                                                                    +1;
              'END';
            SUM[1]:=SUM[1]+DATES[S,2];
            SUM[2]:=SUM[2]+DATES[S,3];
          'END';
        'IF'DAY'LT'50'THEN''GOTO'A1A;
        'FOR'DAY:=-50'STEP'5'UNTIL'50'DO'
          'BEGIN'
            SUB:=DAY/5;
            PCUL[SUB,1]:=CUL[SUB,1]/TT*100;
            PCUL[SUB,2]:=CUL[SUB,2]/TT*100;
            PNODAY[SUB,1]:=NODAY[SUB,1]/TT*100;
            PNODAY[SUB,2]:=NODAY[SUB,2]/TT*100;
          'END';
        MEANTARD[1]:=SUM[1]/TT;
        MEANTARD[2]:=SUM[2]/TT;
        NCU1(1,NAME[1]);
        CNT:=1;
A2A:    PAPERTHROW;
        NEWLINE(1);
        WRITETEXT('('''('3S')'DISTRIBUTION%OF%SCHEDULED%FINISH%DATES'('3S
                                                          ')''')');
        'IF'CNT=1'THEN'WRITETEXT('('UNCORRECTED%VALUES'('C')''')')
        'ELSE'
        WRITETEXT('('CORRECTED%VALUES'('C')''')');
        WRITETEXT('('''('138')'----------------------------------------'('C
                                                          ')''')');
        WRITETEXT('('%DER.CENT.'('C')''')');
        LEVEL:=7A;
A3A:    PRINT(LEVEL,5,0);
        'FOR'DAY:=-50'STEP'5'UNTIL'50'DO'
          'BEGIN'
```

```
        SPACE(3);
        'IF'NODAY[DAY/5,CNT]'GE'LEVEL'THEN'WRITETEXT('('**')')
        'ELSE'
        SPACE(2);
    'END';
  NEWLINE(1);
    LEVEL:=LEVEL-2;
    'IF'LEVEL'GT'0'THEN''GOTO'AGA
    'ELSE'
    NEWLINE(1);
    SPACE(3);
  WRITETEXT('('DAY%%(DAYS)%FROM%T/D.:-%')');
    NEWLINE(1);
    SPACE(10);
    'FOR'DAY:=-50'STEP'5'UNTIL'50'DO'
    PRINT(DAY,2,0);
    NEWLINE(4);
  WRITETEXT('('%FREQUENCY%DISTRIBUTION%:-%')');
    NEWLINE(1);
    SPACE(10);
    'FOR'DAY:=-50'STEP'5'UNTIL'50'DO'
    PRINT(NODAY[DAY/5,CNT1,2,0);
    NEWLINE(1);
    WRITETEXT('('%%%%%%DITTO%(PERCENT)%%%:-%')');
    NEWLINE(1);
    SPACE(10);
    'FOR'DAY:=-50'STEP'5'UNTIL'50'DO'
    PRINT(PNODAY[DAY/5,CNT],2,0);
    NEWLINE(2);
    WRITETEXT('('%%%CUMALATIVE%FREQUENCY%:-%')');
    NEWLINE(1);
    SPACE(10);
    'FOR'DAY:=-50'STEP'5'UNTIL'50'DO'
    PRINT(CUL[DAY/5,CNT1,2,0);
    NEWLINE(1);
```

```
            WRITETEXT('(''%%%%%DITTO%(PERCENT)%%%:-%'')');
            NEWLINE(1);
            SPACE(10);
            'FOR'DAY:=-50'STEP'5'UNTIL'50'DO'
            PRINT(DOUBLEDAY/5,CNT1,2,0);
            NEWLINE(2);
            WRITETEXT('(''('5=5')'MEAN%TARDINESS%')');
            PRINT(MEANTARDY[CNT],4,0);
            NEWLINE(2);
            WRITETEXT('(''('5=5')'M/C%UTILIZATION%PERCENT%%')');
            NEWLINE(1);
            SPACE(30);
            'FOR'M:=1'STEP'1'UNTIL'LIMA'DO'
            PRINT(M,X,0);
            NEWLINE(2);
            SPACE(30);
            'FOR'M:=1'STEP'1'UNTIL'LIMA'DO'
            PRINT(UTZN[M],3,0);
            CNT:=CNT+1;
            PAPERTHROW;
            'IF'CNT'<'5'THEN''GOTO'A2A;
'END';
```

Appendix XIII

Listing of computer programmes for Forge- and Die-Shop scheduling.

```
'PROGRAM'(AXXX)
'INPUT'0=CR0
'OUTPUT'0=LP0
'BEGIN''COMMENT'G.HOMER PROD.PLAN.1;
      'INTEGER'N,H,D,CUDATE,NOORDS,S,T,TT,SS,FIND,CNT,
                LIMIT,BESTJOB,DATEON,TEMPDATE,B,A,X,Y,Z,LOADATE,
                INSHIFT,SHIFTON,DUMMY,FNDATE,FNSFT,DAYS,FINISHDATE,
                DATEFIN,SFTFIN,INSDATE,FNDAY,TEMPSHIFT,LOADSHFT,CUSHIFT,
                ADJUST,LOOP,COUNT,CRITICAL,ZN,NZN,HAM,OH ,DATECU, ON,
               STAGE,START,TEMPCHOICE,TEMP,TRYDATE,INT,FLAGYEAR,MAXDWN
               ,DWN,DWND,DCUDATE,MCUDATE,YCUDATE,DDATE,MDATE,YDATE,
               AD,BD,AB,NXN,NUMTODAY,FOUNDJOB,TIMEDIFF,HALT,DAYFN,SFTFN,
               MARKDATE,F,FF,RESETEPSD,WW;
      'INTEGER''ARRAY'NO[1:20],NN[1:20],SEQ[1:4],ERD[1:20],
                CHOICE[1:20,1:25],ZNO[1:20],SHIFTWORK[1:20],
                TCHOICE[1:20,1:25],OFF[1:20,1:6],FM[71:80],
               WORK[1:40,1:4],HOLDDD[1:100,1:2],HOLDATA[1:100,1:3];
      'REAL'ALATE,BLATE,CLATE,MINPRI,MINSCR,SUMS,LATE,REMSFT,
                PENALTY,COST,TARDY,SUMLT,PT,SFTS;
    'REAL''ARRAY'SCORE[1:24],HAMFILE[1:20,0:25,1:16],NEWORDS[0:100,1:12]
                ,LATES[1:20,0:25],SUMLATE[1:20],HOLDFILE[1:20,0:25,1:16],
                TEMPFILE[0:25,1:16];
'PROCEDURE' OUTDATE(DWND);
'INTEGER'DWND;
'BEGIN'
      'IF'DWND =0'THEN'
            'BEGIN'
               WRITETEXT('('**/**/**')');
               'GOTO' SKIP2;
            'END';
      'IF'DWND'GT'MAXDWN'THEN'
           'BEGIN'
              YDATE:=FLAGYEAR+1;
              DWN:=DWND-MAXDWN;
           'END'
```

```
'ELSE'
'BEGIN'
   YDATE:=FLAGYEAR;
   DWN:=DWND;
'END';
DAY2MONTH(DWN,YDATE);
'IF'MDATE=12'THEN'
    'BEGIN'
      'IF'DDATE>31'THEN'
            'BEGIN'
              MDATE:=1;
              DDATE:=DDATE-31;
              YDATE:=FLAGYEAR+1;
            'END';
    'END';
'IF'DDATE'GE'30'THEN'
            'BEGIN'
              AD:=3;
              BD:=DDATE-30;
            'END'
            'ELSE'
'IF'DDATE'GE'20'THEN'
            'BEGIN'
              AD:=2;
              BD:=DDATE-20;
            'END'
            'ELSE'
'IF'DDATE'GE'10'THEN'
            'BEGIN'
              AD:=1;
              BD:=DDATE-10;
            'END'
            'ELSE'
            'BEGIN'
              AD:=0;
```

```
                BD:=DDATE;
              'END';
CODE('('AD')');
CODE('('BD')');
PRINTCH(AD);
PRINTCH(BD);
WRITETEXT('('/')');
'IF'MDATE'GE'10'THEN'
              'BEGIN'
                AD:=1;
                BD:=MDATE-10;
              'END'
              'ELSE'
              'BEGIN'
                AD:=0;
                BD:= MDATE;
              'END';
CODE('('AD')');
CODE('('BD')');
PRINTCH(AD);
PRINTCH(BD);
WRITETEXT('('/')');
'IF'YDATE'GE'80'THEN'
              'BEGIN'
                AD:=8;
                BD:=YDATE-80;
              'END'
              'ELSE'
              'BEGIN'
                AD:=7;
                BD:=YDATE-70;
              'END';
CODE('('AD')');
CODE('('BD')');
PRINTCH(AD);
```

```
      PRINTCH(RD);
SKIP2;
'END';
'PROCEDURE' DAY2MONTH(DWN,YEAR);
'INTEGER'YEAR,DWN;
'BEGIN'
     'INTEGER'LY,ITRAD,TRAD;
     'REAL'RTRAD,REM;
      RTRAD:=DWN/5;
      ITRAD:=ENTIER(RTRAD+0.000001);
      REM:=ABS(RTRAD-ITRAD);
      'IF'REM'GT'0.00001'THEN'REM:=(REM*5)+2
      'ELSE'
      REM:=0;
      TRAD:=(ITRAD*7)+REM;
      TRAD:=TRAD+(FM[YEAR]-3);
      'IF'YEAR/4=ENTIER(YEAR/4)'THEN'LY:=1
      'ELSE'
      LY:=0;
      'IF'TRAD'LE'31'THEN'
                    'BEGIN'
                       DDATE:=TRAD-0;
                       MDATE:=1;
                    'END'
                    'ELSE'
      'IF'TRAD'LE'59+LY'THEN'
                    'BEGIN'
                       DDATE:=TRAD-31;
                       MDATE:=2;
                    'END'
                    'ELSE'
      'IF'TRAD'LE'90+LY'THEN'
                    'BEGIN'
                       DDATE:=TRAD-59-LY;
                       MDATE:=3;
```

```
                    'END'
                    'ELSE'
'IF'TRAD'LE'120+LY'THEN'
                    'BEGIN'
                       DDATE:=TRAD-90-LY;
                       MDATE:=4;
                    'END'
                    'ELSE'
'IF'TRAD'LE'151+LY'THEN'
                    'BEGIN'
                       DDATE:=TRAD-120-LY;
                       MDATE:=5;
                    'END'
                    'ELSE'
'IF'TRAD'LE'181+LY'THEN'
                    'BEGIN'
                       DDATE:=TRAD-151-LY;
                       MDATE:=6;
                    'END'
                    'ELSE'
'IF'TRAD'LE'212+LY'THEN'
                    'BEGIN'
                       DDATE:=TRAD-181-LY;
                       MDATE:=7;
                    'END'
                    'ELSE'
'IF'TRAD'LE'243+LY'THEN'
                    'BEGIN'
                       DDATE:=TRAD-212-LY;
                       MDATE:=8;
                    'END'
                    'ELSE'
'IF'TRAD'LE'273+LY'THEN'
                    'BEGIN'
                       DDATE:=TRAD-243-LY;
```

```
                        MDATE:=9;
                    'END'
                    'ELSE'
        'IF'TRAD'LE'304+LY'THEN'
                    'BEGIN'
                        DDATE:=TRAD-273-LY;
                        MDATE:=10;
                    'END'
                    'ELSE'
        'IF'TRAD'LE'334+LY'THEN'
                    'BEGIN'
                        DDATE:=TRAD-304-LY;
                        MDATE:=11;
                    'END'
                    'ELSE'
                    'BEGIN'
                        DDATE:=TRAD-334-LY;
                        MDATE:=12;
                    'END';
'END';
'PROCEDURE' TESTIME(START,FINISH,M);
'REAL'      START,FINISH;
'INTEGER'M;
'BEGIN'
        'FOR'AB:=1'STEP'2'UNTIL'5'DO'
          'BEGIN'
            'IF'START'LT'OFF[M,AB]'THEN'
              'BEGIN'
                'IF'FINISH'GT'OFF[M,AB]'THEN'
                  'BEGIN'
                    START:=START;
                    FINISH:=FINISH+(OFF[M,AB+1]-OFF[M,AB])+1;
                  'END';
              'END';
            'IF'START'GE'OFF[M,AB]'THEN'
```

```
                'BEGIN'
                  'IF'START'LE'OFF[M,AB+1]'THEN'
                    'BEGIN'
                      FINISH:=FINISH+(OFF[M,AB+1]-START)+1;
                      START:=OFF[M,AB+1]+1;
                    'END';
                'END';
          'END';
'END';
'PROCEDURE' MONTH2DAY(DATE,MONTH,YEAR);
'INTEGER'DATE,MONTH,YEAR;
'BEGIN'
     'INTEGER'TRAD,IDWN;
     'REAL'RDWN,REM;
      TRAD:='IF'MONTH=1'THEN'0'ELSE'
            'IF'MONTH=2'THEN'31'ELSE'
            'IF'MONTH=3'THEN'59'ELSE'
            'IF'MONTH=4'THEN'90'ELSE'
            'IF'MONTH=5'THEN'120'ELSE'
            'IF'MONTH=6'THEN'151'ELSE'
            'IF'MONTH=7'THEN'181'ELSE'
            'IF'MONTH=8'THEN'212'ELSE'
            'IF'MONTH=9'THEN'243'ELSE'
            'IF'MONTH=10'THEN'273'ELSE'
            'IF'MONTH=11'THEN'304'ELSE'334;
      'IF'MONTH'GE'3'THEN'
        'BEGIN'
          'IF'YEAR/4=ENTIER(YEAR/4)'THEN'TRAD:=TRAD+1;
        'END';
      TRAD:=TRAD+DATE;
      RDWN:=(TRAD-(REM[YEAR]-3))/7;
      'IF'RDWN*7'LT'3'THEN'
        'BEGIN'
          DWN:=RDWN;
          'GOTO'C1;
```

```
        'END';
      IDWN:=ENTIER(RDWN+0.000001);
      REM:=ABS(RDWN-IDWN);
      'IF'REM'GT'0.00001'THEN'REM:=(REM*7)-2
      'ELSE'
      REM:=0;
      DWN:=(IDWN*5)+REM;
C1:
'END';
'PROCEDURE' READATE(ANSWER);
'INTEGER'ANSWER;
'BEGIN'
      DDATE:=READ;
      MDATE:=READ;
      YDATE:=READ;
      'IF'YDATE=00'THEN'
            'BEGIN'
               DWN:=0;
               'GOTO' SKIP1;
            'END';
      MONTH2DAY(DDATE,MDATE,YDATE);
      'IF'FLAGYEAR'LT'YDATE'THEN'ANSWER:=DWN+MAXDWN
      'ELSE'
SKIP1;ANSWER:=DWN;
'END';
'COMMENT'NEXT 13 LINES,PROCEDURE TO EVALUATE DAY + SHIFT,THAT THE JOB IN
  *******QUESTION IS DUE TO FINISH,USING THE  GIVEN START TIME*********;
                  'PROCEDURE'LSFD(PT,STDAY,STSFT,FNDAY,FNSFT);
                  'REAL'PT;
                  'INTEGER'STDAY,STSFT,FNDAY,FNSFT;
                     'BEGIN'
                        DAYS:=ENTIER(PT);
                        REMSFT:=PT-DAYS;
                        REMSFT:=REMSFT/0.25;
                        FNDAY:=STDAY+DAYS;
```

```
                    FNSFT:=STSFT+REMSFT;
                    'IF'FNSFT'GT'4'THEN'
                      'BEGIN'
                        FNDAY:=FNDAY+1;
                        FNSFT:=FNSFT-4;
                      'END'
                    'ELSE'
                    FNSFT:=FNSFT;
                  'END';
'COMMENT'NEXT 31 LINES,PROCEDURE TO EVALUATE THE SCORES OF THE VARIOUS
        WAYS OF PERMUTATING THE 3 JOBS HAVING THE LOWESTPRIORITY FACTOR
 *******AT THE CURRENT MOMENT IN TIME**********************************;
          'PROCEDURE'PERM(A,X,Y,Z);
          'INTEGER'A,X,Y,Z;
            'BEGIN''INTEGER''ARRAY'SQU[1:4];
                  'INTEGER'W;
                   TEMPDATE:=CUDATE;
                   TEMPSHIFT:=CUSHIFT;
                   SCORE[A]:=0;
                   SQU[1]:=SEQ[X];
                   SQU[2]:=SEQ[Y];
                   SQU[3]:=SEQ[Z];
                   'FOR'W:=1'STEP'1'UNTIL'3'DO'
                   'BEGIN'
                   'IF'NEWORDS[SQU[W],5]'LE'TEMPDATE'THEN'
                     'BEGIN'
                       LOADATE:=TEMPDATE;
                       LOADSHFT:=TEMPSHIFT;
                     'END'
                   'ELSE'
                     'BEGIN'
                       LOADATE:=NEWORDS[SQU[W],5];
                       LOADSHFT:=1;
                     'END';
                    'IF'SHIFTWORK[H]=2'THEN'PT:=NEWORDS[SQU[W],6]/2
```

```
                    'ELSE' PT:=NEWORDS[SQU[W],6];
                    ISFD(PT,LOADATE,LOADSHFT,DATEFIN,SFTFIN);
                   TESTIME(LOADATE,DATEFIN,H);
                   'IF'SFTFIN=1'THEN'ADJUST:=0'ELSE'ADJUST:=1;
                   ALATE:=NEWORDS[SQU[W],4]-(DATEFIN+ADJUST);
                  'IF'ALATE'GE'0'THEN'ALATE:=0
                   'ELSE'
                   ALATE:=-ALATE/NEWORDS[SQU[W],2];
                   TEMPDATE:=DATEFIN;
                   TEMPSHIFT:=SFTFIN;
                   SCORE[A]:=SCORE[A]+ALATE;
                   'END';
      'END';
    STAGE:=1;
    CRITICAL:=-4;
    WW:=0;
    FM[71]:=4;
    FM[72]:=3;
    FM[73]:=1;
    FM[74]:=7;
    FM[75]:=6;
    FM[76]:=5;
    FM[77]:=3;
    FM[78]:=2;
    FM[79]:=1;
    FM[80]:=7;
    MAXDWN:=260;
    FLAGYEAR:=READ;
    'IF'FM[FLAGYEAR]=1'THEN'MAXDWN:=265;
    'IF'FLAGYEAR/4=ENTIER(FLAGYEAR/4)'THEN'
                    'BEGIN'
                      'IF'FM[FLAGYEAR]=2'THEN'MAXDWN:=265;
                    'END';
'FOR'H:=1'STEP'1'UNTIL'WW'DO'
  'BEGIN'
```

```
          PRINT(H,5,0);
            NEWLINE(2);
          'FOR'D:=1'STEP'2'UNTIL'5'DO'
            'BEGIN'
              OFF[H,D]:=READ;
              OUTDATE(OFF[H,D]);
              SPACE(4);
              OFF[H,D+1]:=READ;
              OUTDATE(OFF[H,D+1]);
              NEWLINE(1);
            'END';
            NEWLINE(5);
          'END';
      PAPERTHROW;
'COMMENT' READ IN ISFD,S OF DIES , ( MAY BE EPSD OF FORGE.).**********;
       TT:=READ;
       'FOR'S:=1'STEP'1'UNTIL'TT'DO'
        'BEGIN'
          HOLDATA[S,1]:=READ;
          HOLDATA[S,2]:=READ;
          HOLDATA[S,3]:=READ;
        'END';
'COMMENT'NEXT 20 LINES,READS+OUTPUTS EXISTING HAMFILES,LINES 20 TO 35,
          CORRECTS EXISTING HAMFILE RECORDS TO ALLOW FOR DOUBLE SHIFT
          WORKING ON CERTAIN HAMMERS,CHANGES PROD,TIMES AND STARTSHIFT/
  *******FINISHSHIFT TIMES TO UN-CONVENTIONAL FORM**********************;
       'FOR'H:=1'STEP'1'UNTIL'WW'DO'
          'BEGIN'
              SHIFTWORK[H]:=READ;
              NO[H]:=READ;
              ERD[H]:=READ;
              NEWLINE(6);
             WRITETEXT('('HAMMER'('C')'')');
              PRINT(H,2,0);
WRITETEXT('('''('2C')''('7S')'JOB.%%QTY%REQ%%BALANCE%%EPSD'('7S')'DD'('7S
```

```
')'LSSD'('8S')'LSFD'('8S')'PT'('7S')'PRI.'('7S')'HAMMER%CHOICES'('C')'
'('93S')'A%%%%%B%%%%%C%%%%%Z'('C')'')');
                'FOR'N:=1'STEP'1'UNTIL'NO[H]'DO'
                  'BEGIN'
                      NEWLINE(1);
                      PRINT(N,2,0);
                    'FOR'D:=1'STEP'1'UNTIL'16'DO'
                        HAMFILE[H,N,D]:=READ;
      'FOR'D:=1'STEP'1'UNTIL'3'DO'
       PRINT(HAMFILE[H,N,D],5,0);
      'FOR'D:=4,5'DO'
       'BEGIN'
         INT:=HAMFILE[H,N,D];
         OUTDATE(INT);
         SPACE(2);
       'END';
      'FOR'D:=6,8'DO'
       'BEGIN'
         INT:=HAMFILE[H,N,D];
         OUTDATE(INT);
         PRINT(HAMFILE[H,N,D+1],1,0);
       'END';
      PRINT(HAMFILE[H,N,10],2,1);
      'FOR'D:=11,12'DO'
        PRINT(HAMFILE[H,N,D],5,0);
        'FOR'D:=13,14,15'DO'
          PRINT(HAMFILE[H,N,D],3,0);
              'IF'SHIFTWORK[H]=1'THEN'
              'BEGIN'
LL:           'IF'HAMFILE[H,N,7]=2'THEN'HAMFILE[H,N,7]:=3'ELSE'
              HAMFILE[H,N,7]:=HAMFILE[H,N,7];
              'IF'HAMFILE[H,N,9]=2'THEN'HAMFILE[H,N,9]:=3'ELSE'
              HAMFILE[H,N,9]:=HAMFILE[H,N,9];
              'END'
              'ELSE'
```

```
            DUMMY:=DUMMY;
              'END';
    'END':
PAPERTHROW:
      RESETEPSD:=READ;
      DCUDATE:=READ;
      MCUDATE:=READ;
      YCUDATE:=READ;
      MONTH2DAY(DCUDATE,MCUDATE,YCUDATE);
      CUDATE:=DWN;
      'IF'FLAGYEAR'LT'YCUDATE'THEN'CUDATE:=CUDATE+MAXDWN;
      'IF'CUDATE'GE'391'THEN'
                  'BEGIN'
                    CUDATE:=CUDATE-MAXDWN;
'FOR'H:=1'STEP'1'UNTIL'WW'DO'
 'FOR'N:=1'STEP'1'UNTIL'NO[H]'DO'
  'FOR'D:=4,5,6,8,16'DO'
   'BEGIN'
    HAMFILE[H,N,D]:=HAMFILE[H,N,D]-MAXDWN;
    'IF'HAMFILE[H,N,D]'LE'0'THEN'HAMFILE[H,N,D]:=0;
   'END';
'FOR'H:=1'STEP'1'UNTIL'WW'DO'
 'BEGIN'
  ERD[H]:=ERD[H]-MAXDWN;
  'IF'ERD[H]'LE'0'THEN'ERD[H]:=0;
  'FOR'D:=1'STEP'1'UNTIL'6'DO'
   'BEGIN'
    OFF[H,D]:=OFF[H,D]-MAXDWN;
    'IF'OFF[H,D]'LE'0'THEN'OFF[H,D]:=0;
   'END';
 'END';
'FOR'S:=1'STEP'1'UNTIL'TT'DO'
 'BEGIN'
  HOLDATA[S,2]:=HOLDATA[S,2]-MAXDWN;
  'IF'HOLDATA[S,2]'LE'0'THEN'HOLDATA[S,2]:=0;
```

```
      'END';
FLAGYEAR:=FLAGYEAR+1;
MAXDWN:=260;
'IF'FM[FLAGYEAR]=1'THEN'MAXDWN:=265;
'IF'FLAGYEAR/4=ENTIER(FLAGYEAR/4)'THEN'
                    'BEGIN'
                     'IF'FM[FLAGYEAR]=2'THEN'MAXDWN:=265;
                    'END';
                         'END';


'COMMENT'         BEGINING OF MODIFIED PROD. CONT. BLOCK.    ************;


'COMMENT' NEXT 4 LINES READS IN TODAYS PRODUCTION DATA.*********;
NUMTODAY:=READ;
'IF'NUMTODAY=0'THEN''GOTO' STOP;
WRITETEXT('('%%%%%%%%DATE%%%')');
OUTDATE(CUDATE);
NEWLINE(3);
'FOR'A:=1'STEP'1'UNTIL'NUMTODAY'DO'
    'BEGIN'
        WORK[A,1]:=READ;
        WORK[A,2]:=READ;
        'IF'WORK[A,1]=0000'THEN'READATE(WORK[A,3])
        'ELSE'
        WORK[A,3]:=READ;
        'IF'NEXTCH=CODE('('F')')'THEN'WORK[A,4]:=1'ELSE'WORK[A,4]:=0;
    'END';
     H:=0;
'COMMENT'******* BEGINS TO SIMULATE TODAYS WORK DONE.*****************;
CY:   H:=H+1;
CZ:   FOUNDJOB:=0;
'COMMENT'******* NEXT 9 LINES CHECKS TO SEE IF ANY OF THE JOBS PRODUCED
      TODAY HAVE USED THE UNIT UNDER CONSIDERATION,IF SO, GOTO M,ELSE.
```

```
        CHECK NEXT JOB ON WORK DONE TODAY INPUT FILE.*********************;
      'FOR'A:=1'STEP'1'UNTIL'NUMTODAY'DO'
         'BEGIN'
               'IF'WORK[A,2]=H'THEN'
                  'BEGIN'
                        FOUNDJOB:=1;
                        'GOTO'M;
                  'END'
               'ELSE''GOTO'DUM1;
DUM1  :    'END';
      'IF'FOUNDJOB=0'THEN''GOTO'DUM5'ELSE''GOTO'M;
DUM5; 'IF'H<WU'THEN''GOTO'CY
      'ELSE''GOTO'STOP;
'COMMENT'******* IF NEXT JOB NAME HELD ON THIS NAMFILE DOES NOT AGREE
       WITH THE JOB NAME INPUT AS HAVING USED THIS UNIT TODAY,THEN,IF
       JOB NAME INPUT AS HAVING USED THIS UNIT TODAY = 0000,(I.E. THE
       UNIT HAS BROKEN DOWN),THEN GOTO HB  ,ELSE,(I.E. ANOTHER JOB HAS
       BEEN FORCED INTO THE POLE POSITION ON THIS UNIT ),GOTO CB .
          IF THE TWO JOB NAMES DO AGREE, THEN CARRY ON NORMALLY(GOTO W*;
M;    'IF'WORK[A,3]=-9'THEN''GOTO'DELETE;
      'IF'WORK[A,1]#NAMFILE[H,1,1]'THEN''GOTO'DUM6
      'ELSE' 'GOTO' CW;
DUM6; 'IF'WORK[A,1]=0'THEN''GOTO'HB
      'ELSE''GOTO'CB;
CW:
      ERD[H]:=0;
'COMMENT'******* DEDUCT TODAYS PRODUCTION FROM THE BALANCE REQUIRED,  :
      NAMFILE[H,1,3]:=NAMFILE[H,1,3]-WORK[A,3];
      'IF'NAMFILE[H,1,3]>0'THEN'
         'BEGIN'
           'IF'WORK[A,4]#1'THEN'
             'BEGIN'
      WRITETEXT('('('5S')'JOB%NO.%%%')');
      PRINT(NAMFILE[H,1,1],4,0);
      WRITETEXT('(',BALANCE%REMAINING%=%%%')');
```

```
        PRINT(HAMFILE[H,1,3],5,0);
        NEWLINE(3);
'IF'CUDATE'LT'HAMFILE[H,1,6]'THEN'
'BEGIN'
HAMFILE[H,1,6]:=CUDATE;
HAMFILE[H,1,8]:=HAMFILE[H,1,8]-1;
'END';
            'IF'HAMFILE[H,1,9]=1'THEN'TRYDATE:=HAMFILE[H,1,8]-1
            'ELSE'
            TRYDATE:=HAMFILE[H,1,8];
            'IF'TRYDATE'LE'CUDATE'THEN'HAMFILE[H,1,8]:=HAMFILE[H,1,8]+1;
            'FOR'D:=1'STEP'1'UNTIL'3'DO'
               WORK[A,D]:=-999;
            'GOTO'C7;
              'END';
          'END';
        TIMEDIFF:=HAMFILE[H,1,5]-CUDATE-1;
        WRITETEXT('('''('5S')'JOB%NO,%%%')');
        PRINT(HAMFILE[H,1,11,4,0);
        WRITETEXT('('''('5S')'COMPLETED,%ON%HAMMER%NO,%%%')');
        PRINT(H,2,0);
        WRITETEXT('('PUNCTUALITY%%')');
        PRINT(TIMEDIFF,3,0);
        NEWLINE(3);
'COMMENT'******* DELETE THIS 'WORK-DONE-TODAY' RECORD FROM THE FILE,
                                                    WORK[A,D]**;
Q:      'FOR'D:=1'STEP'1'UNTIL'3'DO'
          WORK[A,D]:=-999;
'COMMENT'******* DELETE THIS COMPLETED JOB FROM THE RESPECTIVE HAMFILE,
         AND SHUNT ALL THE FOLLOWING JOBS ON THIS FILE,ONE POSITION FORWAR
                                                                    ;
        'FOR'N:=1'STEP'1'UNTIL'NO[H]-1'DO'
          'FOR'D:=1'STEP'1'UNTIL'16'DO'
           HAMFILE[H,N,D]:=HAMFILE[H,N+1,D];
        NO[H]:=NO[H]-1;
```

```
      'GOTO'CZ;
'COMMENT' FOLLOWING BLOCK DELETES ALL RECORD OF A JOB , AS REQUESTED
          BY USER . ********************                          *******;
DELETE:   WRITETEXT('('('SS')'JOB%NO.%%%')');
          PRINT(WORK[A,1],4,0);
          WRITETEXT('('('SS')'ON%HAMMER%NO,')');
          PRINT(H,2,0);
          WRITETEXT('('REMOVED%FROM%FURTHER%CONSIDERATION(JOB%CANCELLED
          ,MATL.%UNAVAILABLE%E.T.C.)')');
          NEWLINE(3);
           'FOR'N:=1'STEP'1'UNTIL'NO[H]'DO'
              'IF'HAMFILE[H,N,1]=WORK[A,1]'THEN'
                 NXN:=N;
       'FOR'N:=NXN'STEP'1'UNTIL'NO[H]-1'DO'
         'FOR'D:=1'STEP'1'UNTIL'16'DO'
            HAMFILE[H,N,D]:=HAMFILE[H,N+1,D];
       NO[H]:=NO[H]-1;
       'FOR'D:=1'STEP'1'UNTIL'3'DO'
         WORK[A,D]:=-999;
       'GOTO'CZ;
'COMMENT'******* CONSIDER THE CASE WHERE A JOB HAS BEEN FORCED ONTO THIS
       HAMFILE IN POLE POSITION*****                              *****;
CB:
      ERD[H]:=0;
'COMMENT'******* IF THE DISPLACED JOB HAS NOT BEEN COMPLETED, SET ITS
      "START" DATE = TODAY,SHIFT=1**                              *****;
      'IF'HAMFILE[H,1,3]<HAMFILE[H,1,2]'THEN'
              'BEGIN'
                 DAYS:=HAMFILE[H,1,8]-CUDATE;
                 SFTS:=(HAMFILE[H,1,9]-1)*0.25;
                 HAMFILE[H,1,10]:=DAYS+SFTS;
                 HAMFILE[H,1,3]:=HAMFILE[H,1,2];
              'END'
      'ELSE''GOTO'DUM3;
DUM3: WRITETEXT('('('SS')'JOB%NO.%%%')');
```

```
        PRINT(HAMFILE[H,1,11,4,0);
        WRITETEXT('('('5S')'ON%HAMMER%NO.')');
        PRINT(H,2,0);
        WRITETEXT('('('1S')'NOT%STARTED/CONTINUED%AS%PER%SCHEDULE,')');
        WRITETEXT('('%REPLACED%BY%JOB%NO.%%%')');
        PRINT(WORK[A,1],4,0);
        NEWLINE(3);
'COMMENT'******* NEXT 7 LINES IDENTIFIES ORIGIN OF'FORCED' JOB.********;
        'FOR'X:=1'STEP'1'UNTIL'WW'DO'
            'FOR'N:=1'STEP'1'UNTIL'NO[X]'DO'
              'BEGIN'
                  'IF'HAMFILE[X,N,1]=WORK[A,1]'THEN''GOTO'OUT
                  'ELSE'
                  'GOTO'DUM4;
DUM4:           'END';
OUT:    'FOR'D:=1'STEP'1'UNTIL'16'DO'
        TEMPFILE[H,D]:=HAMFILE[X,N,D];
        HALT:=N;
'COMMENT'******* REMOVE THIS JOB FROM ITS ORIGINAL HAMFILE (TEMPORARILY
         STORED IN TEMPFILE) AND BRING LATER JOBS ONE POSITION FORWARD TO
         FILL THE RESULTING GAP*****                                    ********;
        'FOR'N:=HALT'STEP'1'UNTIL'NO[X]-1'DO'
            'FOR'D:=1'STEP'1'UNTIL'16'DO'
             HAMFILE[X,N,D]:=HAMFILE[X,N+1,D];
             NO[X]:=NO[X]-1;
'COMMENT'******* PUSH ALL JOBS ON THE HAMFILE UNDER CONSIDERATION ONE
         POSITION BACK AND PLACE THE 'FORCED' JOB IN THE RESULTANT EMPTY,
         POSITION ONE, POSITION.****                                    ********;
        'FOR'N:=NO[H]+1'STEP'-1'UNTIL'2'DO'
            'FOR'D:=1'STEP'1'UNTIL'16'DO'
              HAMFILE[H,N,D]:=HAMFILE[H,N-1,D];
        'FOR'D:=1'STEP'1'UNTIL'16'DO'
            HAMFILE[H,1,D]:=TEMPFILE[H,D];
            NO[H]:=NO[H]+1;
'COMMENT'******* SET START DATE OF THE 'FORCED' JOB = TODAY,SHIFT = 1,
```

```
        NEXT 5 LINES, EVALUATES THE CORRESPONDING FINISH TIME OF THIS
        'FORCED' JOB.*******                                          ********;
      HAMFILE[H,1,6]:=CUDATE;
      HAMFILE[H,1,7]:=1;
                'IF'SHIFTWORK[H]=2'THEN'PT:=HAMFILE[H,1,10]/2
                'ELSE'
                PT:=HAMFILE[H,1,10];
                LSED(PT,HAMFILE[H,1,6],HAMFILE[H,1,7],DAYFN,SFTFN);
HAMFILE[H,1,8]:=DAYFN;HAMFILE[H,1,9]:=SFTFN;
                TESTIME(HAMFILE[H,1,6],HAMFILE[H,1,8],H);
      'GOTO' CW;
'COMMENT'******* CONSIDER THE CASE OF A UNIT BREAK-DOWN BEING REPORTED,
       OR A MORE UPTO DATE ESTIMATE OF THE E.R.D. FOR A BROKE-DOWN UNIT;
HB:   WRITETEXT('('('('27S')'HAMMER%NO.%%%')');
      PRINT(H,2,0);
      WRITETEXT('('('('5S')'FAILED!!!%%E.R.D.%IS%%')');
      OUTDATE(WORK[A,3]);
      NEWLINE(3);
         MARKDATE:=ERD[H];
         ERD[H]:=WORK[A,3];
         'IF'HAMFILE[H,1,3]<HAMFILE[H,1,2]'THEN'
           'BEGIN'
             'IF'MARKDATE=0'THEN'
               'BEGIN'
                 HAMFILE[H,1,8]:=HAMFILE[H,1,8]+(ERD[H]-CUDATE);
               'END'
             'ELSE'
               'BEGIN'
                 'IF'ERD[H]=0'THEN'MARKDATE:=0;
                 HAMFILE[H,1,8]:=HAMFILE[H,1,8]+(ERD[H]-MARKDATE);
               'END';
           'END';
           'GOTO'CY;
STOP:
```

```
'COMMENT'     END OF MODIFIED PROD. CONT. BLOCK.    **********************;

WRITETEXT('('%%%%%%%%DATE%%%')');
OUTDATE(CUDATE);
NEWLINE(3);
      CUSHIFT:=0;
      CUDATE:=CUDATE+1;
      DATECU:=CUDATE;
      'IF'RESETEPSD=1'THEN'
        'BEGIN'
          'FOR'H:=1'STEP'1'UNTIL'WW'DO'
            'FOR'N:=1'STEP'1'UNTIL'NO[H]'DO'
              'BEGIN'
                'IF'CUDATE'GE'HAMFILE[H,N,16]'THEN'
                  HAMFILE[H,N,4]:=CUDATE
                'ELSE'
                  HAMFILE[H,N,4]:=HAMFILE[H,N,16];
              'END';
          'END'
        'ELSE'
          'BEGIN'
            'FOR'H:=1'STEP'1'UNTIL'WW'DO'
              'FOR'N:=1'STEP'1'UNTIL'NO[H]'DO'
                'BEGIN'
                  'FOR'S:=1'STEP'1'UNTIL'TT'DO'
                'IF'HAMFILE[H,N,1]=HOLDATA[S,1]'THEN'
                  'BEGIN'
                    'IF'HAMFILE[H,N,16]'LT'HOLDATA[S,2]'THEN'
                      HAMFILE[H,N,4]:=HOLDATA[S,2]
                    'ELSE'
                      HAMFILE[H,N,4]:=HAMFILE[H,N,16];
                  'GOTO' SKIPB;
                'END';
```

```
SKIPB:              'END';
                  'END';
        NOORDS:=READ;
        NEWLINE(5);
        'IF'NOORDS#0'THEN'
        WRITETEXT('('('10S')'JOB%NAME'('4S')'PRI.'('1S')'
QTY%REQ.'('8S')'DD'('11S')'EPSD'('10S')'PT'('9S')'HAMMER%CHOICES
'('C')''('80S')'A'('7S')'B'('7S')'C'('7S')'Z'('C')'')');
'COMMENT'NEXT 39 LINES,READS+OUTPUTS NEWORDERS,ADDS JOBS ALREADY HELD ON
          HAMFILES(EXCEPT THOSE ALREADY IN PRODUCTION) TO THE NEWORDERS,
          FORMS A COMBINED  'NEWORDS''FILE. LINES 10 TO 14,CHANGE PROD.
          TIME F.T.C. TO UN-CONVENTIONAL FORM(AS BEFORE) FOR DOUBLE SHIFT
  *******WORKING*****************************************************;
        'FOR'S:=1'STEP'1'UNTIL'NOORDS'DO'
          'BEGIN'
                NEWLINE(1);
                PRINT(S,2,0);
                SPACE(5);
                'FOR'D:=1'STEP'1'UNTIL'3'DO'
                  'BEGIN'
                    NEWORDS[S,D]:=READ;
                    PRINT(NEWORDS[S,D],5,0);
                  'END';
                SPACE(6);
                READATE(INT);
                OUTDATE(INT);
                NEWORDS[S,41:=INT;
                SPACE(6);
                READATE(INT);
                OUTDATE(INT);
                NEWORDS[S,51:=INT;
                NEWORDS[S,61:=READ;
                PRINT(NEWORDS[S,6],8,1);
                'FOR'D:=7'STEP'1'UNTIL'10'DO'
                  'BEGIN'
```

```
                    NEWORDS[S,D]:=READ;
'COMMENT'  FOLLOWING CARD ONLY REQ INORDER TO SAVE REPUNCHING CARDS ***;
'IF'D>7'THEN'NEWORDS[S,D]:=0;
                    PRINT(NEWORDS[S,D],5,0);
                  'END';
                NEWORDS[S,11]:=NEWORDS[S,5];
            'END';
        S:=NOORDS;
        'FOR'H:=1'STEP'1'UNTIL'WW'DO'
        'FOR'N:=1'STEP'1'UNTIL'NO[H]'DO'
           'BEGIN'
               'IF'N=1'THEN'
                  'BEGIN'
                      'IF'HAMFILE[H,N,2]#HAMFILE[H,N,3]'THEN'
                         'BEGIN'
                             NN[H]:=1;
                             'GOTO'AA;
                         'END'
                      'ELSE'
                      NN[H]:=0;
                  'END'
               'ELSE'
               S:=S;
               S:=S+1;
               NEWORDS[S,1]:=HAMFILE[H,N,1];
               NEWORDS[S,2]:=HAMFILE[H,N,11];
               NEWORDS[S,3]:=HAMFILE[H,N,2];
               NEWORDS[S,4]:=HAMFILE[H,N,5];
               NEWORDS[S,5]:=HAMFILE[H,N,4];
               NEWORDS[S,6]:=HAMFILE[H,N,10];
               NEWORDS[S,7]:=HAMFILE[H,N,12];
               NEWORDS[S,8]:=HAMFILE[H,N,13];
               NEWORDS[S,9]:=HAMFILE[H,N,14];
               NEWORDS[S,10]:=HAMFILE[H,N,15];
               NEWORDS[S,11]:=HAMFILE[H,N,16];
```

```
                'FOR'D:=1'STEP'1'UNTIL'16'DO'
                HAMFILE[H,N,D]:=0;
AA:         'END';
        NEWORDS[0,4]:=900;
        NEWORDS[0,5]:=0;
        NEWORDS[0,6]:=0;
        'FOR'H:=1'STEP'1'UNTIL'WW'DO'
          HAMFILE[H,0,81:=0;
        T:=S;
        TT:=S;
'COMMENT'NEXT 9 LINES, UPDATES ONE QUARTER SHIFT/DAY.RESETS HAMMER UNDER
  *******CONCIDERATION TO ZERO*****************************************;
BB:     CUSHIFT:=CUSHIFT+1;
        'IF'CUSHIFT=5'THEN'
          'BEGIN'
                CUSHIFT:=1;
                CUDATE:=CUDATE+1;
          'END'
        'ELSE'
        CUDATE:=CUDATE;
        H:=0;
'COMMENT'NEXT 18 LINES, SELECTS HAMMER FOR CONCIDERATION,CHECKS TO SEE
          IF THIS HAMMER HAS RETURNED(AND IS NOT BROKEN DOWN),IF SO,
          GOTO''DD'' OR IF NOT,GOTO''HH''(I.E.CONCIDER NEXT HAMMER,OR,
  *******IF H=WW,UPDATE TIME BY ONE QUARTER SHIFT/DAY******************;
CC:     H:=H+1;
        TRYDATE:=CUDATE;
        TESTIME(TRYDATE,TRYDATE,H);
        'IF'TRYDATE#CUDATE'THEN''GOTO'HH;
        'IF'HAMFILE[H,NN[H],8]-CUDATE'LE'-1'THEN''GOTO'X1
        'ELSE'
        'IF'HAMFILE[H,NN[H],8]-CUDATE=0'THEN'
          'BEGIN'
                'IF'CUSHIFT'GE'HAMFILE[H,NN[H],9]'THEN''GOTO'X1
                'ELSE'
```

```
                'GOTO'HH;
          'END'
       'ELSE'
       'GOTO'HH;
X1:          'BEGIN'
                    'IF'ERD[H]  'LE'CUDATE'THEN''GOTO'DD'ELSE'
                    'GOTO'HH;
             'END';
HH:    'IF'H'GE'WW'THEN''GOTO'BB
       'ELSE'
       'GOTO'CC;
'COMMENT'NEXT 17 LINES, HAVING FOUND A HAMMER THAT HAS RETURNED,EVALUATE
         DYNAMIC PRIORITY FACTORS FOR EVERY REMAINING JOB REQUIRING THIS
  *******HAMMER*****************************************************;
DD:    FIND:=0;
NAME6:       'BEGIN''REAL''ARRAY' DYNAPRI[1:100];
       'FOR'S:=1'STEP'1'UNTIL'TT'DO'
          'BEGIN'
                         'IF'NEWORDS[S,7]=H'THEN''GOTO'PP
                         'ELSE'
                         'IF'NEWORDS[S,8]=H'THEN'
                                    'BEGIN'
                                       TEMP:=NEWORDS[S,7];
                                       NEWORDS[S,7]:=NEWORDS[S,8];
                                       NEWORDS[S,8]:=TEMP;
                                       'GOTO'PP;
                                    'END'
                         'ELSE'
                         'IF'NEWORDS[S,9]=H'THEN'
                                    'BEGIN'
                                       TEMP:=NEWORDS[S,7];
                                       NEWORDS[S,7]:=NEWORDS[S,9];
                                       NEWORDS[S,9]:=TEMP;
                                       'GOTO'PP;
                                    'END'
```

```
                                'ELSE'
                                'IF'STAGE#1'THEN'
                                  'BEGIN'
                                    'IF'NEWORDS[S,10]=H'THEN''GOTO'PP
                                    'ELSE'
                                       'BEGIN'
                                          DYNAPRI[S]:=9999;
                                          'GOTO'EE;
                                       'END'
                                  'END'
                                'ELSE'
                              'BEGIN'
                                DYNAPRI[S]:=9999;
                                'GOTO'EE;
                              'END';
PP:           FIND:=FIND+1;
  'IF'SHIFTWORK[H]=2'THEN'PT:=NEWORDS[S,6]/2
  'ELSE'
  PT:=NEWORDS[S,6];
  LSED(PT,CUDATE,CUSHIFT,ENDAY,ENSFT);
  TESTIME(CUDATE,ENDAY,H);
        'IF'ENSFT=1'THEN'ADJUST:=0'ELSE'ADJUST:=1;
  DYNAPRI[S]:=NEWORDS[S,4]-(ENDAY+ADJUST);
             'IF'DYNAPRI[S]'GE'0'THEN'DYNAPRI[S]:=DYNAPRI[S]
                                                  *NEWORDS[S,2]
             'ELSE'
             DYNAPRI[S]:=DYNAPRI[S]/NEWORDS[S,2];
             SS:=S;
EE:      'END';
'COMMENT'IF NO JOBS REQUIRE THE AVAILABLE HAMMER THEN GOTO HH,(CONCIDER
  *******NEXT HAMMER)*****************************************************;
'COMMENT'IF ONLY ONE REMAINING JOB REQUIRES THIS HAMMER,SET START TIME
  *******AND LOAD THIS JOB ONTO APPROPRIATE HAMFILE(GOTO II)***********;
           'IF'FIND=0'THEN''GOTO'XX
           'ELSE'
```

```
        'IF'FIND=1'THEN'
                    'BEGIN'
                        NN[H]:=NN[H]+1;
                        NO[H]:=NN[H];
                        'IF'NEWORDS[SS,5]'LE'CUDATE  'THEN'
                        'BEGIN'
                            DATEON:=CUDATE;
                            SHIFTON:=CUSHIFT;
                        'END'
                        'ELSE'
                        'BEGIN'
                            DATEON:=NEWORDS[SS,5];
                            SHIFTON:=1;
                        'END' ;
                            'GOTO'XX;
                    'END'
        'ELSE'
'COMMENT'NEXT 24 LINES,OF REMAINING JOBS REQUIRING THIS HAMMER,CHOOSE
  *******THE 3 MOST CRITICAL (LOWES PRIORIT FACTOR)**********************;
        CNT:=0;
        SEQ[1]:=SEQ[2]:=SEQ[3]:=SEQ[4]:=0;
FF:     CNT:=CNT+1;
        MINPRI:=999;
        'FOR'S:=1'STEP'1'UNTIL'TT'DO'
           'BEGIN'
               'IF'S=SEQ[1]'THEN''GOTO'GG'ELSE'
               'IF'S=SEQ[2]'THEN''GOTO'GG'ELSE'
               'IF'S=SEQ[3]'THEN''GOTO'GG'ELSE'
               'IF'S=SEQ[4]'THEN''GOTO'GG'ELSE'
               'IF'DYNAPRI[S]<MINPRI'THEN'
                  'BEGIN'
                      MINPRI:=DYNAPRI[S];
                      SEQ[CNT]:=S;
                  'END'
               'ELSE'
```

```
                MINPRI:=MINPRI;
GG:         'END';
          'IF'CNT<3'THEN'
             'BEGIN'
                  'IF'CNT<FIND'THEN''GOTO'FF
                  'ELSE'
                  CNT:=CNT;
             'END'
          'ELSE'
          CNT:=CNT;
XX:
'END';
'IF'FIND=0'THEN'
   'BEGIN'
     'IF'STAGE#1'THEN''GOTO'NAME5
     'ELSE'
     'GOTO'HH
   'END'
'ELSE'
'IF'FIND=1'THEN''GOTO'II
'ELSE'
'COMMENT'NEXT 6 LINES,SCHEDULE THE 3(OR 2) MOST CRITICAL JOBS IN ALL
  *******POSSIBLE PERMUTATIONS,EVALUATE THE PENALTY SCORE FOR EACH PERM;
      PERM(1,1,2,3);
      PERM(2,1,3,2);
      PERM(3,2,1,3);
      PERM(4,2,3,1);
      PERM(5,3,1,2);
      PERM(6,3,2,1);
'COMMENT'NEXT 19 LINES, FIND THE ARRANGEMENT GIVING THE LOWES PENALTY
  ********SCORE,NOTE THE FIRST JOB AT THE HEAD OF THIS ARRANGEMENT(SS)**;
      'IF'FIND'GE'3'THEN'LIMIT:=6'ELSE'LIMIT:=4;
      BESTJOB:=0;
      MINSCR:=999;
      'FOR'A:=1'STEP'1'UNTIL'LIMIT'DO'
```

```
            'BEGIN'
                'IF'SCORE[A]<MINSCR'THEN'
                  'BEGIN'
                      MINSCR:=SCORE[A];
                      BESTJOB:=A;
                  'END'
                'ELSE'
                MINSCR:=MINSCR;
            'END';
        'IF'BESTJOB=1'THEN'X:=1'ELSE'
        'IF'BESTJOB=2'THEN'X:=1'ELSE'
        'IF'BESTJOB=3'THEN'X:=2'ELSE'
        'IF'BESTJOB=4'THEN'X:=2'ELSE'
        X:=3;
        SS:=SEQ[X];
'COMMENT'NEXT 8 LINES, TENTATIVELY LOAD THE SELECTED JOB,EVALUATE THE
        START DATE FOR THIS JOB,IS THERE ANY FLOAT BETWEEN NOW AND
        START DATE?? IF NO,GOTO II(FIRM LOAD),IF YES, FIND A SUITABLE
        'INSERT' JOB,IF ANY.(LINES 9 TO 43).IF A SUITABLE JOB FOUND,
 *******LOAD THIS INSERT JOB (SS=S)*************************************;
        NN[H]:=NN[H]+1;
        NO[H]:=NN[H];
        'IF'NEWORDS[SS,5]'LE'CUDATE  'THEN'
            'BEGIN'
                DATEON:=CUDATE;
                SHIFTON:=CUSHIFT;
                'GOTO'II;
            'END'
        'ELSE'
        'BEGIN'
        DATEON:=NEWORDS[SS,5];
        SHIFTON:=1;
        'END';
        'FOR'S:=1'STEP'1'UNTIL'TT'DO'
            'BEGIN'
```

```
                'IF'NEWORDS[S,7]=H'THEN''GOTO'QQ
                'ELSE'
                'IF'NEWORDS[S,8]=H'THEN'
                      'BEGIN'
                          TEMP:=NEWORDS[S,7];
                          NEWORDS[S,7]:=NEWORDS[S,8];
                          NEWORDS[S,8]:=TEMP;
                          'GOTO' QQ;
                      'END'
                'ELSE'
                'IF'NEWORDS[S,9]=H'THEN'
                      'BEGIN'
                          TEMP:=NEWORDS[S,7];
                          NEWORDS[S,7]:=NEWORDS[S,9];
                          NEWORDS[S,9]:=TEMP;
                          'GOTO' QQ;
                      'END'
                'ELSE'
                'GOTO'RR;
QQ:             'BEGIN'
                    'IF'NEWORDS[S,5]'LE'CUDATE'THEN'
                        'BEGIN'
                            INSDATE:=CUDATE;
                            INSHIFT:=CUSHIFT;
                        'END'
                    'ELSE'
                        'BEGIN'
                            INSDATE:=NEWORDS[S,5];
                            INSHIFT:=1;
                        'END';
                    'IF'SHIFTWORK[H]=2'THEN'PT:=NEWORDS[S,6]/2
                    'ELSE'
                    PT:=NEWORDS[S,6];
                    LSED(PT,INSDATE,INSHIFT,FNDATE,FNSFT);
                      TESTIME(INSDATE,FNDATE,H);
```

```
                    'IF'ENDATE'LT'DATEON'THEN'
                      'BEGIN'
                         DATEON:=INSDATE;
                         SHIFTON:=INSHIFT;
                         SS:=S;
                         'GOTO' II;
                      'END'
                    'ELSE'
                    'IF'ENDATE'EQ'DATEON'THEN'
                       'BEGIN'
                          'IF'ENSFT'LE'SHIFTON'THEN'
                             'BEGIN'
                                  DATEON:=INSDATE;
                                  SHIFTON:=INSHIFT;
                                  SS:=S ;
                                  'GOTO'II;
                             'END'
                          'ELSE'
                           SS:=SS;
                       'END'
                     'ELSE'
                      SS:=SS;
              'END';
RR:       'END';
II;
'COMMENT'NEXT 14 LINES,LOAD SELECTED (SS) JOB ON APPROPRIATE HAMFILE,
  *******DELETE THIS JOB  FROM FURTHER CONCIDERATION*******************;
            'IF'STAGE#1'THEN'
                      'BEGIN'
                           H:=HAM;
                           HOLDFILE[H,NN[H],1]:=NEWORDS[SS,1];
                           HOLDFILE[H,NN[H],2]:=NEWORDS[SS,3];
                           HOLDFILE[H,NN[H],3]:=NEWORDS[SS,3];
                           HOLDFILE[H,NN[H],4]:=NEWORDS[SS,5];
                           HOLDFILE[H,NN[H],5]:=NEWORDS[SS,4];
```

```
                'IF'SHIFTWORK[H]=2'THEN'PT:=NEWORDS[SS,6]/2
                'ELSE'
                PT:=NEWORDS[SS,6];
                LSED(PT,DATEON,SHIFTON,FNDAY,FNSFT);
                                TESTIME(DATEON,FNDAY,H);
                              HOLDFILE[H,NN[H],6]:=DATEON;
                              HOLDFILE[H,NN[H],7]:=SHIFTON;
                              HOLDFILE[H,NN[H],8]:=FNDAY;
                              HOLDFILE[H,NN[H],9]:=FNSFT;
                             HOLDFILE[H,NN[H],10]:=NEWORDS[SS,6];
                             HOLDFILE[H,NN[H],11]:=NEWORDS[SS,2];
                             HOLDFILE[H,NN[H],12]:=NEWORDS[SS,7];
                             HOLDFILE[H,NN[H],13]:=NEWORDS[SS,8];
                             HOLDFILE[H,NN[H],14]:=NEWORDS[SS,9];
                             HOLDFILE[H,NN[H],15]:=NEWORDS[SS,10];
                             HOLDFILE[H,NN[H],16]:=NEWORDS[SS,11];
                             TCHOICE[H,NN[H]]:=NEWORDS[SS,12];
                             NEWORDS[SS,7]:=0;
                             NEWORDS[SS,8]:=0;
                             NEWORDS[SS,9]:=0;
                             NEWORDS[SS,10]:=0;
                             T:=T-1;
                             'IF'T>0'THEN''GOTO'BBB
                             'ELSE'
                             'GOTO'NAME5;
                         'END'
        'ELSE'
HAMFILE[H,NN[H],1]:=NEWORDS[SS,1];
HAMFILE[H,NN[H],2]:=NEWORDS[SS,3];
HAMFILE[H,NN[H],3]:=NEWORDS[SS,3];
HAMFILE[H,NN[H],4]:=NEWORDS[SS,5];
HAMFILE[H,NN[H],5]:=NEWORDS[SS,4];
'IF'SHIFTWORK[H]=2'THEN'PT:=NEWORDS[SS,6]/2
'ELSE'
PT:=NEWORDS[SS,6];
```

```
        LSFD(PT,DATEON,SHIFTON,FNDAY,FNSFT);
         TESTIME(DATEON,FNDAY,H);
        HAMFILE[H,NN[H],6]:=DATEON;
        HAMFILE[H,NN[H],7]:=SHIFTON;
        HAMFILE[H,NN[H],8]:=FNDAY;
        HAMFILE[H,NN[H],9]:=FNSFT;
        HAMFILE[H,NN[H],10]:=NEWORDS[SS,6];
        HAMFILE[H,NN[H],11]:=NEWORDS[SS,2];
        HAMFILE[H,NN[H],12]:=NEWORDS[SS,7];
              HAMFILE[H,NN[H],13]:=NEWORDS[SS,8];
              HAMFILE[H,NN[H],14]:=NEWORDS[SS,9];
              HAMFILE[H,NN[H],15]:=NEWORDS[SS,10];
              HAMFILE[H,NN[H],16]:=NEWORDS[SS,11];
              NEWORDS[SS,7]:=0;
              NEWORDS[SS,8]:=0;
              NEWORDS[SS,9]:=0;
              NEWORDS[SS,10]:=0;
        T:=T-1;
'COMMENT'GOES BACK TO HH,CONCIDERS NEXT HAMMER (IF THERE ARE STILL JOBS
  *******REQUIRING SCHEDULING)*****************************************;
        'IF'T>0'THEN''GOTO'HH
        'ELSE'
        PAPERTHROW;
'COMMENT'NEXT 42 LINES,OUTPUTS LATEST HAMFILES.BEFORE PRINTING,RE-CONVER
          PROD.TIME,STARTSHIFT/FINISHSHIFT TIMES TO CONVENTIONAL FORM FOR
  *******DOUBLE SHIFT HAMMERS(LINES 11 TO 23)***************************;
        'FOR'H:=1'STEP'1'UNTIL'WW'DO'
           'BEGIN'
                NEWLINE(6);
                WRITETEXT('('HAMMER'('C')'')');
                PRINT(H,2,0);
WRITETEXT('('('('2C')''('7S')'JOB.%%QTY%REQ%%BALANCE%%EPSD'('7S')'DD'('7S
')'LSSD'('8S')'LSFD'('8S')'PT'('7S')'PRI.'('7S')'HAMMER%CHOICES'('C')'
'('93S')'A%%%%%B%%%%%C%%%%%Z'('C')'')');
SUMS:=0;
```

```
                'FOR'N:=1'STEP'1'UNTIL'NO[H]'DO'
                  'BEGIN'
                 'IF'SHIFTWORK[H]=1'THEN'
                 'BEGIN'
JJ:              'IF'HAMFILE[H,N,7]=3'THEN'HAMFILE[H,N,7]:=2'ELSE'
                  HAMFILE[H,N,7]:=HAMFILE[H,N,7];
                 'IF'HAMFILE[H,N,9]=3'THEN'HAMFILE[H,N,9]:=2'ELSE'
                  HAMFILE[H,N,9]:=HAMFILE[H,N,9];
                 'END'
                 'ELSE'
                 DUMMY:=DUMMY;
                        NEWLINE(1);
                        PRINT(N,2,0);
      'FOR'D:=1'STEP'1'UNTIL'3'DO'
       PRINT(HAMFILE[H,N,D],5,0);
      'FOR'D:=4,5'DO'
       'BEGIN'
         INT:=HAMFILE[H,N,D];
         OUTDATE(INT);
         SPACE(2);
       'END';
      'FOR'D:=6,8'DO'
       'BEGIN'
         INT:=HAMFILE[H,N,D];
         OUTDATE(INT);
         PRINT(HAMFILE[H,N,D+1],1,0);
       'END';
      PRINT(HAMFILE[H,N,10],2,1);
      'FOR'D:=11,12'DO'
        PRINT(HAMFILE[H,N,D],5,0);
        'FOR'D:=13,14,15'DO'
          PRINT(HAMFILE[H,N,D],3,0);
       'IF'HAMFILE[H,N,9]=1'THEN'ADJUST:=0'ELSE'ADJUST:=1;
       LATE:=(HAMFILE[H,N,5]-(HAMFILE[H,N,8]*ADJUST))/HAMFILE[H,N,11];
PRINT(LATE,2,0);
```

```
'IF'LATE'GE'0'THEN'LATE:=0'ELSE'LATE:=LATE;
SUMS:=LATE+SUMS;
       LATES[H,N]:=LATE;
                  'END';
NEWLINE(1);
WRITETEXT('('('('1145')'----'('C')'')');
SPACE(113);
PRINT(SUMS,3,0);
       SUMLATE[H]:=SUMS;
         'END';
'COMMENT'***** THIS COMPLETES STAGE 1 , STAGE 2 BEGINS ***************;
      STAGE:=2;
      'FOR'H:=1'STEP'1'UNTIL'WW'DO'
         'FOR'N:=1'STEP'1'UNTIL'NO[H]'DO'
            'BEGIN'
               'IF'SHIFTWORK[H]=1'THEN'
               'BEGIN'
LLL:            'IF'HAMFILE[H,N,7]=2'THEN'HAMFILE[H,N,7]:=3'ELSE'
               HAMFILE[H,N,7]:=HAMFILE[H,N,7];
               'IF'HAMFILE[H,N,9]=2'THEN'HAMFILE[H,N,9]:=3'ELSE'
               HAMFILE[H,N,9]:=HAMFILE[H,N,9];
               'END'
               'ELSE'
               DUMMY:=DUMMY;
                 'IF'HAMFILE[H,N,13]=0'THEN'CHOICE[H,N]:=0
                 'ELSE'
                 'IF'HAMFILE[H,N,14]=0'THEN'CHOICE[H,N]:=1
                 'ELSE'
                 CHOICE[H,N]:=2;
            'END';
NAME4:LOOP:=0;
      'IF'STAGE=2'THEN'PENALTY:=0,0
      'ELSE'
      PENALTY:=2.0;
NAME1:COUNT:=0;
```

```
        LOOP:=LOOP+1;
        OH:=0;
L1:     OH:=OH+1;     ON:=0;
L2:     ON:=ON+1;
              'BEGIN'
                  H:=OH;
                  N:=ON;
                   'IF'CHOICE[H,N]=0'THEN''GOTO'NAME2
                   'ELSE'
                   ZN:=N;
                   NZN:=N;
NAME3:             'IF'LATEST[H,NZN]>CRITICAL'THEN'
                                          'BEGIN'
                                            'IF'NZN<NO[H]'THEN'
                                                  'BEGIN'
                                                     NZN:=NZN+1;
                                                     'GOTO'NAME3;
                                                  'END'
                                             'ELSE'
                                             'GOTO'NAME2;
                                          'END'
                    'ELSE'
'COMMENT'*****HAVING FOUND A CRITICAL JOB ON THIS UNIT,PROGRAMME TRIES
               TO RESCHEDULE THE JOB IN QUESTION,ZN,ON ITS ALTERNATE CHOI
                                                        UNIT*********;
                   COUNT:= COUNT+1;
                   'FOR'D:=1'STEP'1'UNTIL'16'DO'
                      TEMPFILE[ZN,D]:=HAMFILE[H,ZN,D];
                  TEMPCHOICE:=CHOICE[H,ZN];
                   'FOR'N:=1'STEP'1'UNTIL'ZN-1'DO'
                      'BEGIN'
                       TCHOICE[H,N]:=CHOICE[H,N];
                       'FOR'D:=1'STEP'1'UNTIL'16'DO'
                          HOLDFILE[H,N,D]:=HAMFILE[H,N,D];
                      'END';
```

```
'FOR'N:=ZN'STEP'1'UNTIL'NO[H]-1'DO'
   'BEGIN'
    TCHOICE[H,N]:=CHOICE[H,N+1];
    'FOR'D:=1'STEP'1'UNTIL'16'DO'
        HOLDFILE[H,N,D]:=NAMFILE[H,N+1,D];
   'END';
ZNO[H]:=NO[H]-1;
'FOR'N:=ZN'STEP'1'UNTIL'ZNO[H]'DO'
   'BEGIN'
       'IF'HOLDFILE[H,N,4]'LE'HOLDFILE[H,N-1,8]'THEN'
                          'BEGIN'
                             LOADATE:=HOLDFILE[H,N-1,8];
                             LOADSHFT:=HOLDFILE[H,N-1,9];
                          'END'
        'ELSE'
        'BEGIN'
             LOADATE:=HOLDFILE[H,N,4];
             LOADSHFT:=1;
        'END';
       'IF'SHIFTWORK[H]=2'THEN'PT:=HOLDFILE[H,N,10]/2
       'ELSE'
       PT:=HOLDFILE[H,N,10];
       LSED(PT,LOADATE,LOADSHFT,DATEFIN,SFTFIN);
         TESTIME(LOADATE,DATEFIN,H);
        HOLDFILE[H,N,6]:=LOADATE;
        HOLDFILE[H,N,7]:=LOADSHFT;
        HOLDFILE[H,N,8]:=DATEFIN;
        HOLDFILE[H,N,9]:=SFTFIN;
   'END';
COST:=0;
'FOR'N:=1'STEP'1'UNTIL'ZNO[H]'DO'
   'BEGIN'
       'IF'HOLDFILE[H,N,9]=1'THEN'ADJUST:=0
       'ELSE'
       ADJUST:=1;
```

```
                    TARDY:=(HOLDFILE[H,N,8]+ADJUST)-HOLDFILE[H,N,5];
                    'IF'TARDY'LE'0'THEN'TARDY:=0
                    'ELSE'
                    TARDY:=TARDY/HOLDFILE[H,N,11];
                    COST:=COST+TARDY;
                 'END';
              'IF'CHOICE[H,ZN]=2'THEN'HAM:=TEMPFILE[ZN,14]
              'ELSE'
              'IF'STAGE=3'THEN'HAM:=TEMPFILE[ZN,15]
              'ELSE'
              HAM:=TEMPFILE[ZN,13];
              CHOICE[H,ZN]:=CHOICE[H,ZN]-1;
              TEMPCHOICE:=CHOICE[H,ZN];
CUDATE:=DATECU;
H:=HAM;
'IF'HAMFILE[H,1,3]#HAMFILE[H,1,2]'THEN'
              'BEGIN'
                    TCHOICE[HAM,1]:=CHOICE[HAM,1];
                 'FOR'D:=1'STEP'1'UNTIL'16'DO'
                    'BEGIN'
                       HOLDFILE[H,1,D]:=HAMFILE[H,1,D];
                       START:=2;
                       NN[H]:=1;
                    'END';
              'END'
'ELSE'
   'BEGIN'
      START:=1;
      NN[H]:=0;
   'END';
S:=0;
'FOR'N:=START'STEP'1'UNTIL'NO[H]'DO'
   'BEGIN'
      S:=S+1;
      NEWORDS[S,1]:=HAMFILE[H,N,1];
```

```
                    NEWORDS[S,2]:=HAMFILE[H,N,11];
                    NEWORDS[S,3]:=HAMFILE[H,N,2];
                    NEWORDS[S,4]:=HAMFILE[H,N,5];
                    NEWORDS[S,5]:=HAMFILE[H,N,4];
                    NEWORDS[S,6]:=HAMFILE[H,N,10];
                    NEWORDS[S,7]:=HAMFILE[H,N,12];
                    NEWORDS[S,8]:=HAMFILE[H,N,13];
                    NEWORDS[S,9]:=HAMFILE[H,N,14];
                    NEWORDS[S,10]:=HAMFILE[H,N,15];
                    NEWORDS[S,11]:=HAMFILE[H,N,16];
                    NEWORDS[S,12]:=CHOICE[HAM,N];
                 'END';
               S:=S+1;
               NEWORDS[S,1]:=TEMPFILE[ZN,1];
               NEWORDS[S,2]:=TEMPFILE[ZN,11];
               NEWORDS[S,3]:=TEMPFILE[ZN,2];
               NEWORDS[S,4]:=TEMPFILE[ZN,5];
               NEWORDS[S,5]:=TEMPFILE[ZN,4];
               NEWORDS[S,6]:=TEMPFILE[ZN,10];
               NEWORDS[S,7]:=TEMPFILE[ZN,12];
               NEWORDS[S,8]:=TEMPFILE[ZN,13];
               NEWORDS[S,9]:=TEMPFILE[ZN,14];
               NEWORDS[S,10]:=TEMPFILE[ZN,15];
               NEWORDS[S,11]:=TEMPFILE[ZN,16];
               NEWORDS[S,12]:=TEMPCHOICE;
               NEWORDS[0,4]:=999;
               NEWORDS[0,5]:=0;
               NEWORDS[0,6]:=0;
               HOLDFILE[H,0,81:=0;
               CUSHIFT:=0;
BBB:           CUSHIFT:=CUSHIFT+1;
                 'IF'CUSHIFT=5'THEN'
                        'BEGIN'
                            CUSHIFT:=1;
                            CUDATE:=CUDATE+1;
```

```
                    'END'
                'ELSE'
                CUDATE:=CUDATE;
                  TRYDATE:=CUDATE;
                  TESTIME(TRYDATE,TRYDATE,H);
                  'IF'TRYDATE#CUDATE'THEN''GOTO'BBB;
              'IF'HOLDFILE[H,NN[H],8]-CUDATE'LE'-1'THEN''GOTO'X2
              'ELSE'
              'IF'HOLDFILE[H,NN[H],8]-CUDATE=0'THEN'
                          'BEGIN'
                              'IF'CUSHIFT'GE'HOLDFILE[H,NN[H],9]'THEN'
                                                  'GOTO'X2
                              'ELSE'
                              'GOTO'BBB;
                          'END'
              'ELSE'
              'GOTO'BBB;
X2:           'BEGIN'
                  'IF'ERD[H]'LE'CUDATE'THEN''GOTO'DDD
                  'ELSE'
                  'GOTO'BBB;
              'END';
DDD:          TT:=S;
              T:=S;
              FIND:=0;
              'GOTO'NAME6;
NAME5:            ZNO[H]:=NO[H];
                  NO[H]:=NO[H]-1;
                  H:=OH;
                  'FOR'N:=1'STEP'1'UNTIL'ZNO[HAM]'DO'
                     'BEGIN'
                         'IF'HOLDFILE[HAM,N,9]=1'THEN'ADJUST:=0
                         'ELSE'
                         ADJUST:=1;
                         TARDY:=(HOLDFILE[HAM,N,8]+ADJUST)-HOLDFILE[HAM
```

```
                                                              ,N,5];
            'IF'TARDY'LE'0'THEN'TARDY:=0
            'ELSE'
            TARDY:=TARDY/HOLDFILE[HAM,N,11];
            COST:=COST+TARDY;
        'END';
    COST:=COST+PENALTY;
    SUMLT:=SUMLATE[H]+SUMLATE[HAM];
    SUMLT:=-SUMLT;
    'IF'SUMLT>COST'THEN'
        'BEGIN'
            SUMS:=0;
            'FOR'N:=1'STEP'1'UNTIL'ZNO[H]'DO'
            'BEGIN'
            CHOICE[H,N]:=TCHOICE[H,N];
                'FOR'D:=1'STEP'1'UNTIL'16'DO'
                    HAMFILE[H,N,D]:=HOLDFILE[H,N,D];
            'IF'HAMFILE[H,N,9]=1'THEN'ADJUST:=0
            'ELSE'
            ADJUST:=1;
            LATE:=(HAMFILE[H,N,5]-(HAMFILE[H,N,8]+ADJUST))
            /HAMFILE[H,N,11];
            'IF'LATE'GE'0'THEN'LATE:=0'ELSE'LATE:=LATE;
            SUMS:=SUMS+LATE;
            LATES[H,N]:=LATE;
            'END';
            SUMLATE[H]:=SUMS;
            SUMS:=0;
            'FOR'N:=1'STEP'1'UNTIL'ZNO[HAM]'DO'
            'BEGIN'
            CHOICE[HAM,N]:=TCHOICE[HAM,N];
                'FOR'D:=1'STEP'1'UNTIL'16'DO'
                    HAMFILE[HAM,N,D]:=HOLDFILE[HAM,N,D];
            'IF'HAMFILE[HAM,N,9]=1'THEN'ADJUST:=0
            'ELSE'
```

```
                        ADJUST:=1;
                        LATE:=(HAMFILE[HAM,N,5]-(HAMFILE[HAM,N,8]+ADJUST))
                        /HAMFILE[HAM,N,11];
                        'IF'LATE'GE'0'THEN'LATE:=0'ELSE'LATE:=LATE;
                        SUMS:=SUMS+LATE;
                        LATES[HAM,N]:=LATE;
                          'END';
                          SUMLATE[HAM]:=SUMS;
                        NO[H]:=ZNO[H];
                        NO[HAM]:=ZNO[HAM];
                     'END'
                  'ELSE'
                  DUMMY:=DUMMY;
                  H:=OH;
                  N:=ON;
NAME2:       'END';
             'IF'ON<NO[OH]'THEN''GOTO' L2;
             'IF'OH<WW'THEN''GOTO' L1;
             'IF'COUNT=0'THEN''GOTO'NAME7;
             'IF'LOOP<10'THEN''GOTO'NAME1;
NAME7:       'IF'STAGE=3'THEN''GOTO'STTOP
             'ELSE'
             STAGE:=3;
'COMMENT'*****THIS COMPLETES STAGE 2 ,STAGE 3 BEGINS*******************;
             'FOR'H:=1'STEP'1'UNTIL'WW'DO'
               'FOR'N:=1'STEP'1'UNTIL'NO[H]'DO'
                  'IF'HAMFILE[H,N,15]>0'THEN'CHOICE[H,N]:=1
                  'ELSE'
                  CHOICE[H,N]:=0;
             'GOTO'NAME4;
STTOP:
             PAPERTHROW;
'COMMENT'NEXT 42 LINES,OUTPUTS LATEST HAMFILES.BEFORE PRINTING,RE-CONVER
         PROD.TIME,STARTSHIFT/FINISHSHIFT TIMES TO CONVENTIONAL FORM FOR
  *******DOUBLE SHIFT HAMMERS(LINES 11 TO 23)*****************************;
```

```
        'FOR'H:=1'STEP'1'UNTIL'WW'DO'
           'BEGIN'
                NEWLINE(6);
                WRITETEXT('('HAMMER'('C')'')');
                PRINT(H,2,0);
WRITETEXT('('('('2C')''('7S')'JOB.%%QTY%REQ%%BALANCE%%EPSD'('7S')'DD'('7S
')'LSSD'('8S')'LSFD'('8S')'PT'('7S')'PRI.'('7S')'HAMMER%CHOICES'('C')'
'('93S')'A%%%%%B%%%%C%%%%Z'('C')'')');
SUMS:=0;
                'FOR'N:=1'STEP'1'UNTIL'NO[H]'DO'
                   'BEGIN'
                 'IF'SHIFTWORK[H]=1'THEN'
                  'BEGIN'
JJJ:              'IF'HAMFILE[H,N,7]=3'THEN'HAMFILE[H,N,7]:=2'ELSE'
                  HAMFILE[H,N,7]:=HAMFILE[H,N,7];
                 'IF'HAMFILE[H,N,9]=3'THEN'HAMFILE[H,N,9]:=2'ELSE'
                  HAMFILE[H,N,9]:=HAMFILE[H,N,9];
                  'END'
                 'ELSE'
                  DUMMY:=DUMMY;
                         NEWLINE(1);
                         PRINT(N,2,0);
      'FOR'D:=1'STEP'1'UNTIL'3'DO'
       PRINT(HAMFILE[H,N,D],5,0);
      'FOR'D:=4,5'DO'
       'BEGIN'
         INT:=HAMFILE[H,N,D];
         OUTDATE(INT);
         SPACE(2);
       'END';
      'FOR'D:=6,8'DO'
       'BEGIN'
         INT:=HAMFILE[H,N,D];
         OUTDATE(INT);
         PRINT(HAMFILE[H,N,D+1],1,0);
```

```
        'END';
      PRINT(HAMFILE[H,N,10],2,1);
      'FOR'D:=11,12'DO'
        PRINT(HAMFILE[H,N,D],5,0);
        'FOR'D:=13,14,15'DO'
          PRINT(HAMFILE[H,N,D],3,0);
       'IF'HAMFILE[H,N,9]=1'THEN'ADJUST:=0'ELSE'ADJUST:=1;
       LATE:=(HAMFILE[H,N,5]-(HAMFILE[H,N,8]+ADJUST))/HAMFILE[H,N,11];
PRINT(LATE,2,0);
'IF'LATE'GE'0'THEN'LATE:=0'ELSE'LATE:=LATE;
SUMS:=LATE+SUMS;
       LATES[H,N]:=LATE*HAMFILE[H,N,11];
                    'END';
NEWLINE(1);
WRITETEXT('('('114S')'----'('C')'')');
SPACE(113);
PRINT(SUMS,3,0);
          'END';
    PAPERTHROW;
      WRITETEXT('('('50S')'JOBS%LIKELY%TO%BE%LATE.'('C')''('50S')'
----------------------'('3C')'')');
      'FOR'H:=1'STEP'1'UNTIL'WW'DO'
        'BEGIN'
          WRITETEXT('('%HAMMER')');
          PRINT(H,3,0);
          NEWLINE(2);
          WRITETEXT('('('10S')'JOB%NO.'('15S')'SCHED.%FINISH%DATE
'('15S')'DAYS%LATE%(UNWEIGHTED)'('8S')'PRI.'('3C')'')');
          'FOR'N:=1'STEP'1'UNTIL'NO[H]'DO'
            'IF'LATES[H,N]<0'THEN'
              'BEGIN'
                SPACE(9);
                PRINT(HAMFILE[H,N,1],5,0);
                SPACE(20);
                'IF'HAMFILE[H,N,9]=1'THEN'ADJUST:=0'ELSE'ADJUST:=1;
```

```
                    INT:=HAMFILE[H,N,8]+ADJUST;
                    OUTDATE(INT);
                    SPACE(22);
                    PRINT(LATES[H,N],3,0);
                    SPACE(22);
                    PRINT(HAMFILE[H,N,11],1,0);
                    NEWLINE(2);
                  'END';
                NEWLINE(5);
                'END';
              PAPERTHROW;
          'IF'RESETERSD=1'THEN'
          'BEGIN'
            F:=0;
            'FOR'H:=1'STEP'1'UNTIL'WW'DO'
              'FOR'N:=1'STEP'1'UNTIL'NO[H]'DO'
                'BEGIN'
                  F:=F+1;
                  HOLDDD[F,1]:=HAMFILE[H,N,1];
                  HOLDDD[F,2]:=HAMFILE[H,N,6];
                'END';
                  FF:=F;
          'END';
'END'
```

```
'PROGRAM'(AXXX)
'INPUT'0=CR0
'OUTPUT'0=LP0
'BEGIN''COMMENT'DIE SHOP PLAN  G.HOMER;
      'INTEGER'M,N,D,Q,S,NODIES,T,TT,CUDATE,CUSHIFT,FIND,MM,STDAY,STSFT,
   DATEFIN,SFTFIN,FINISHED,ADJUST,SS,DATEON,SHIFTON,LEGALDAY,LEGALSFT,
INSDATE,INSHIFT,FNDAY,FNSFT,DUMMY,LOADATE,LOADSFT,DAYS,CNT,BESTJOB,A,X,Y
,Z,W,LIMIT,MARK,SQ,PUNCT,XN,HOLD,SC,DD,XD,TESTMAC,TRYDATE,INT,FLAGYEAR,
MAXDWN,DWN,DWND,DCUDATE,MCUDATE,YCUDATE,DDATE,MDATE,YDATE,AD,BD,AB,
TESTEPSD,F,FF,SCHEDRUN,NUMTODAY,FOUNDOP,HALT,VV;
      'REAL''ARRAY'SUBMAST[0:50,5:24,1:4],SCORE[1:6],DLATE[1:50,1:2],
SUBGND[1:50,5:24,1:4],MASTER[0:50,1:33],GNDMAST[1:50,1:33];
      'INTEGER''ARRAY'DNO[1:20],DERD[1:20],DIESFILE[1:20,0:45,1:7],
NN[1:20],TNO[1:20],SEQ[1:3],SORT[1:50],OFF[1:20,1:6],FM[71:80],
HOLDDD[1:50,1:2],HOLDATA[1:50,1:3],DWORK[1:30,1:3];
      'REAL'PT,ALATE,REMSET,MINPRI,MINSCR,HOLD26,HOLD27;
'PROCEDURE' DEHEAD;
'BEGIN'
                    SQ:=MASTER[S,25];
        'FOR'MM:=1'STEP'1'UNTIL'VV'DO'
            'BEGIN'
              'IF'DIESFILE[MM,1,1]=DWORK[A,1]'THEN'
                'BEGIN'
                   'IF'DIESFILE[MM,1,2]'LT'SQ'THEN'
                   SCAN(MASTER,Q,DWORK,0,A);
                     'BEGIN'
                        'FOR'D:=1'STEP'1'UNTIL'7'DO'
                          DIESFILE[MM,1,D]:=0;
                     'END';
                'END';
            'END';
'END';
'PROCEDURE'EVALPT(S,M,N);
  'INTEGER'S,M,N;
  'BEGIN'
```

```
    'INTEGER'TEMPSEQ;
      TEMPSEQ:=MASTER[S,25];
      MASTER[S,25]:=DIESFILE[M,N,2];
      NEXTOP(S,MASTER,SUBMAST);
      PT:=MASTER[S,27];
      MASTER[S,25]:=TEMPSEQ;
      NEXTOP(S,MASTER,SUBMAST);
  'END';
'PROCEDURE' READATE(ANSWER);
'INTEGER'ANSWER;
'BEGIN'
      DDATE:=READ;
      MDATE:=READ;
      YDATE:=READ;
      'IF'YDATE=00'THEN'
            'BEGIN'
               DWN:=0;
               'GOTO' SKIP1;
            'END';
      MONTH2DAY(DDATE,MDATE,YDATE);
      'IF'FLAGYEAR'LT'YDATE'THEN'ANSWER:=DWN+MAXDWN
      'ELSE'
SKIP1:ANSWER:=DWN;
'END';
'PROCEDURE' TESTIME(START,FINISH,M);
'REAL'        START,FINISH;
'INTEGER'M;
'BEGIN'
      'FOR'AB:=1'STEP'2'UNTIL'5'DO'
        'BEGIN'
          'IF'START'LT'OFF[M,AB]'THEN'
            'BEGIN'
              'IF'FINISH'GT'OFF[M,AB]'THEN'
                'BEGIN'
                  START:=START;
```

```
                    FINISH:=FINISH+(OFF[M,AB+1]-OFF[M,AB])+1;
                  'END';
              'END';
            'IF'START'GE'OFF[M,AB]'THEN'
              'BEGIN'
                'IF'START'LE'OFF[M,AB+1]'THEN'
                  'BEGIN'
                    FINISH:=FINISH+(OFF[M,AB+1]-START)+1;
                    START:=OFF[M,AB+1]+1;
                  'END';
              'END';
        'END';
'END';
'PROCEDURE' OUTDATE(DWND);
'INTEGER'DWND;
'BEGIN'
      'IF'DWND =0'THEN'
            'BEGIN'
              WRITETEXT('('**/**/**')');
              'GOTO' SKIP2;
            'END';
      'IF'DWND'GT'MAXDWN'THEN'
          'BEGIN'
            YDATE:=FLAGYEAR+1;
            DWN:=DWND-MAXDWN;
          'END'
      'ELSE'
      'BEGIN'
        YDATE:=FLAGYEAR;
        DWN:=DWND;
      'END';
      DAY2MONTH(DWN,YDATE);
      'IF'MDATE=12'THEN'
          'BEGIN'
            'IF'DDATE>31'THEN'
```

```
                'BEGIN'
                  MDATE:=1;
                  DDATE:=DDATE-31;
                  YDATE:=FLAGYEAR+1;
                'END';
    'END';
'IF'DDATE'GE'30'THEN'
                'BEGIN'
                  AD:=3;
                  BD:=DDATE-30;
                'END'
                'ELSE'
'IF'DDATE'GE'20'THEN'
                'BEGIN'
                  AD:=2;
                  BD:=DDATE-20;
                'END'
                'ELSE'
'IF'DDATE'GE'10'THEN'
                'BEGIN'
                  AD:=1;
                  BD:=DDATE-10;
                'END'
                'ELSE'
                'BEGIN'
                  AD:=0;
                  BD:=DDATE;
                'END';
CODE('('AD')');
CODE('('BD')');
PRINTCH(AD);
PRINTCH(BD);
WRITETEXT('('/')');
'IF'MDATE'GE'10'THEN'
                'BEGIN'
```

```
                    AD:=1;
                    BD:=MDATE-10;
                  'END'
                  'ELSE'
                  'BEGIN'
                    AD:=0;
                    BD:= MDATE;
                  'END';
      CODE('('AD')');
      CODE('('BD')');
      PRINTCH(AD);
      PRINTCH(BD);
      WRITETEXT('('/')');
      'IF'YDATE'GE'80'THEN'
                  'BEGIN'
                    AD:=8;
                    BD:=YDATE-80;
                  'END'
                  'ELSE'
                  'BEGIN'
                    AD:=7;
                    BD:=YDATE-70;
                  'END';
      CODE('('AD')');
      CODE('('BD')');
      PRINTCH(AD);
      PRINTCH(BD);
SKIP2;
'END';
'PROCEDURE' DAY2MONTH(DWN,YEAR);
'INTEGER'YEAR,DWN;
'BEGIN'
     'INTEGER'LY,ITRAD,TRAD;
     'REAL'RTRAD,REM;
      RTRAD:=DWN/5;
```

```
ITRAD:=ENTIER(RTRAD+0.000001);
REM:=ABS(RTRAD-ITRAD);
'IF'REM'GT'0.00001'THEN'REM:=(REM*5)+2
'ELSE'
REM:=0;
TRAD:=(ITRAD*7)+REM;
TRAD:=TRAD+(FM[YEAR]-3);
'IF'YEAR/4=ENTIER(YEAR/4)'THEN'LY:=1
'ELSE'
LY:=0;
'IF'TRAD'LE'31'THEN'
                    'BEGIN'
                       DDATE:=TRAD-0;
                       MDATE:=1;
                    'END'
                    'ELSE'
'IF'TRAD'LE'59+LY'THEN'
                    'BEGIN'
                       DDATE:=TRAD-31;
                       MDATE:=2;
                    'END'
                    'ELSE'
'IF'TRAD'LE'90+LY'THEN'
                    'BEGIN'
                       DDATE:=TRAD-59-LY;
                       MDATE:=3;
                    'END'
                    'ELSE'
'IF'TRAD'LE'120+LY'THEN'
                    'BEGIN'
                       DDATE:=TRAD-90-LY;
                       MDATE:=4;
                    'END'
                    'ELSE'
'IF'TRAD'LE'151+LY'THEN'
```

```
                    'BEGIN'
                       DDATE:=TRAD-120-LY;
                       MDATE:=5;
                    'END'
                    'ELSE'
'IF'TRAD'LE'181+LY'THEN'
                    'BEGIN'
                       DDATE:=TRAD-151-LY;
                       MDATE:=6;
                    'END'
                    'ELSE'
'IF'TRAD'LE'212+LY'THEN'
                    'BEGIN'
                       DDATE:=TRAD-181-LY;
                       MDATE:=7;
                    'END'
                    'ELSE'
'IF'TRAD'LE'243+LY'THEN'
                    'BEGIN'
                       DDATE:=TRAD-212-LY;
                       MDATE:=8;
                    'END'
                    'ELSE'
'IF'TRAD'LE'273+LY'THEN'
                    'BEGIN'
                       DDATE:=TRAD-243-LY;
                       MDATE:=9;
                    'END'
                    'ELSE'
'IF'TRAD'LE'304+LY'THEN'
                    'BEGIN'
                       DDATE:=TRAD-273-LY;
                       MDATE:=10;
                    'END'
                    'ELSE'
```

```
        'IF'TRAD'LE'334+LY'THEN'
                        'BEGIN'
                           DDATE:=TRAD-304-LY;
                           MDATE:=11;
                        'END'
                        'ELSE'
                        'BEGIN'
                           DDATE:=TRAD-334-LY;
                           MDATE:=12;
                        'END';
'END';
'PROCEDURE' MONTH2DAY(DATE,MONTH,YEAR);
'INTEGER'DATE,MONTH,YEAR;
'BEGIN'
     'INTEGER'TRAD,IDWN;
     'REAL'RDWN,REM;
      TRAD:='IF'MONTH=1'THEN'0'ELSE'
            'IF'MONTH=2'THEN'31'ELSE'
            'IF'MONTH=3'THEN'59'ELSE'
            'IF'MONTH=4'THEN'90'ELSE'
            'IF'MONTH=5'THEN'120'ELSE'
            'IF'MONTH=6'THEN'151'ELSE'
            'IF'MONTH=7'THEN'181'ELSE'
            'IF'MONTH=8'THEN'212'ELSE'
            'IF'MONTH=9'THEN'243'ELSE'
            'IF'MONTH=10'THEN'273'ELSE'
            'IF'MONTH=11'THEN'304'ELSE'334;
      'IF'MONTH'GE'3'THEN'
        'BEGIN'
          'IF'YEAR/4=ENTIER(YEAR/4)'THEN'TRAD:=TRAD+1;
        'END';
      TRAD:=TRAD+DATE;
      RDWN:=(TRAD-(FM[YEAR]-3))/7;
      'IF'RDWN*7'LT'3'THEN'
        'BEGIN'
```

```
            DWN:=RDWN;
            'GOTO'C1;
          'END';
        IDWN:=ENTIER(RDWN+0.000001);
        REM:=ABS(RDWN-IDWN);
        'IF'REM'GT'0.00001'THEN'REM:=(REM*7)-2
        'ELSE'
        REM:=0;
        DWN:=(IDWN*5)+REM;
C1:
'END';
'PROCEDURE'OUTPUTMAST;
'BEGIN'
      'FOR'S:=1'STEP'1'UNTIL'Q'DO'
       'BEGIN'
         WRITETEXT('('''('7S')'JOB,'('6S')'DD'('7S')'EPSD'('5S')'PRI,'('
12S')'OP.1'('8S')'OP.2'('8S')'OP.3'('8S')'OP.4'('8S')'OP.5'('C')''')');
         SPACE(5);
         PRINT(MASTER[S,11,5,0);
         SPACE(1);
         INT:=MASTER[S,2];
         OUTDATE(INT);
         SPACE(2);
         INT:=MASTER[S,3];
         OUTDATE(INT);
         PRINT(MASTER[S,4],5,0);
         SPACE(8);
         'FOR'DD:=1'STEP'1'UNTIL'4'DO'
           'BEGIN'
             'FOR'D:=5'STEP'2'UNTIL'13'DO'
               'BEGIN'
                 PRINT(SUBMAST[S,D,DD],2,0);
                 PRINT(SUBMAST[S,D+1,DD],2,1);
               'END';
             NEWLINE(1);
```

```
                SPACE(48);
              'END';
            NEWLINE(1);
            WRITETEXT('(''('C')'%%NEXT:-%%%%%SEQ.%%%%M/C'('6S')'PT.'('18S'
)'OP.6'('8S')'OP.7'('8S')'OP.8'('8S')'OP.9'('8S')'OP.10'('C')'')');
            SPACE(9);
            PRINT(MASTER[S,25],5,0);
            PRINT(MASTER[S,26],5,0);
            PRINT(MASTER[S,27],5,1);
            SPACE(13);
            'FOR'DD:=1'STEP'1'UNTIL'4'DO'
              'BEGIN'
                'FOR'D:=15'STEP'2'UNTIL'23'DO'
                  'BEGIN'
                    PRINT(SUBMAST[S,D,DD],2,0);
                    PRINT(SUBMAST[S,D+1,DD],2,1);
                  'END';
                NEWLINE(1);
                SPACE(48);
              'END';
              NEWLINE(1);
              'FOR'X:=1'STEP'1'UNTIL'118'DO'
                WRITETEXT('('-')');
              NEWLINE(3);
          'END';
'END';
'PROCEDURE'LEGALST(S,ARAY,BRAY,CRAY);
'INTEGER'S;
'REAL''ARRAY'ARAY,CRAY;
'INTEGER''ARRAY'BRAY;
'BEGIN''INTEGER'POSTMC,PRESEQ,PREMC;
        FINISHED:=1;
        POSTMC:=ARAY[S,26];
        PRESEQ:=ARAY[S,25]-1;
        'IF'PRESEQ =0'THEN''GOTO'FSTOP;
```

```
          'FOR'DD:=1'STEP'1'UNTIL'4'DO'
            'IF'CRAY[S,(PRESEQ*2+3),DD]>0'THEN'
                                    'BEGIN'
                                       PREMC:=CRAY[S,(PRESEQ*2+3),DD];
                                       FINISHED:=0;
                                    'END';
FSTOP:LEGALSFT:=1;
          'IF'FINISHED=1'THEN'
              'BEGIN'
                'IF'TESTEPSD=1'THEN'LEGALDAY:=ARAY[S,3]
                'ELSE'
                LEGALDAY:=0;
              'END'
          'ELSE'
          'BEGIN'
             MARK:=0;
             'FOR'N:=TNO[PREMC]'STEP'-1'UNTIL'1'DO'
               'BEGIN'
                  'IF'BRAY[PREMC,N,1]=ARAY[S,1]'THEN'
                    'BEGIN'
                       MARK:=1;
                       LEGALDAY:=BRAY[PREMC,N,5];
                       LEGALSFT:=BRAY[PREMC,N,6];        'GOTO'LP;
                    'END'
                  'ELSE'
                  DUMMY:=DUMMY;
               'END';
               'IF'MARK=0'THEN'LEGALDAY:=0
               'ELSE'LEGALDAY:=LEGALDAY;
          'END';
LP:     'IF'CUDATE'GT'LEGALDAY'THEN'
              'BEGIN'
                LEGALDAY:=CUDATE;
                LEGALSFT:=CUSHIFT;
              'END'
```

```
        'ELSE'
        'IF'CUDATE'EQ'LEGALDAY'THEN'
              'BEGIN'
                'IF'CUSHIFT'GT'LEGALSFT'THEN'
                     'BEGIN'
                          LEGALDAY:=CUDATE;
                          LEGALSFT:=CUSHIFT;
                     'END'
                'ELSE'
                DUMMY:=DUMMY;
              'END'
        'ELSE'
        DUMMY:=DUMMY;
        'IF'BRAY[POSTMC,TNO[POSTMC],5]'GT'LEGALDAY'THEN'
              'BEGIN'
                LEGALDAY:=BRAY[POSTMC,TNO[POSTMC],5];
                LEGALSFT:=BRAY[POSTMC,TNO[POSTMC],6];
              'END'
        'ELSE'
        'IF'BRAY[POSTMC,TNO[POSTMC],5]'EQ'LEGALDAY'THEN'
              'BEGIN'
                'IF'BRAY[POSTMC,TNO[POSTMC],6]'GT'LEGALSFT'THEN'
                     'BEGIN'
                          LEGALDAY:=BRAY[POSTMC,TNO[POSTMC],5];
                          LEGALSFT:=BRAY[POSTMC,TNO[POSTMC],6];
                     'END'
                'ELSE'
                DUMMY:=DUMMY;
              'END'
        'ELSE'
        DUMMY:=DUMMY;
'END';
'PROCEDURE'NEXTOP(S,RAY,SUBRAY);
'INTEGER'S;
'REAL''ARRAY'SUBRAY,RAY;
```

```
'BEGIN'  'INTEGER'XD;
      FINISHED:=1;
      'IF'RAY[S,251>10.001'THEN''GOTO'SKIP3;
      RAY[S,261:=RAY[S,271:=RAY[S,28]:=RAY[S,29]:=RAY[S,30]:=
      RAY[S,311]:=RAY[S,321:=RAY[S,33]:=0;
      XD:=24;
      'FOR'DD:=1'STEP'1'UNTIL'4'DO'
        'BEGIN'
          'IF'SUBRAY[S,(ENTIER(RAY[S,25]+0.001)*2+3),DD]>0.001'THEN'
            'BEGIN'
              XD:=XD+2;
             RAY[S,XD]:=SUBRAY[S,(ENTIER(RAY[S,25]+0.001)*2+3),DD];
             RAY[S,XD+1]:=SUBRAY[S,(ENTIER(RAY[S,25]+0.001)*2+4),DD];
              FINISHED:=0;
            'END';
SKIP3;  'END';
'END';
'PROCEDURE'DPERM(A,X,Y,Z);
'INTEGER'A,X,Y,Z;
'BEGIN''REAL''ARRAY'STF[1:3,5:24,1:4],TF[1:3,1:33];
       'INTEGER''ARRAY'TEMPRET[1:19,0:45,1:6];
      SCORE[A]:=0;
      'FOR'D:=1'STEP'1'UNTIL'4'DO'
        'BEGIN'
          TF[1,D]:=MASTER[SEQ[X],D];
          TF[2,D]:=MASTER[SEQ[Y],D];
          TF[3,D]:=MASTER[SEQ[Z],D];
        'END';
      'FOR'D:=25'STEP'1'UNTIL'33'DO'
        'BEGIN'
          TF[1,D]:=MASTER[SEQ[X],D];
          TF[2,D]:=MASTER[SEQ[Y],D];
          TF[3,D]:=MASTER[SEQ[Z],D];
        'END';
      'FOR'D:=5'STEP'1'UNTIL'24'DO'
```

```
            'FOR'DD:=1'STEP'1'UNTIL'4'DO'
              'BEGIN'
                STF[1,D,DD]:=SUBMAST[SEQ[X],D,DD];
                STF[2,D,DD]:=SUBMAST[SEQ[Y],D,DD];
                STF[3,D,DD]:=SUBMAST[SEQ[Z],D,DD];
              'END';
          'FOR'MM:=1'STEP'1'UNTIL'VV'DO'
            'BEGIN'
              TNO[MM]:=NN[MM];
              'FOR'N:=0'STEP'1'UNTIL'TNO[MM]'DO'
                'FOR'D:=1'STEP'1'UNTIL'6'DO'
                  TEMPRET[MM,N,D]:=DIESFILE[MM,N,D];
            'END';
          'FOR'W:=1'STEP'1'UNTIL'3'DO'
            'IF'TF[W,11<=9000'THEN''GOTO'DX5'ELSE'
            'BEGIN'
DX3:        LEGALST(W,TF,TEMPRET,STF);
            LOADATE:=LEGALDAY;
            LOADSFT:=LEGALSFT;
            MM:=ENTIER(TF[W,26]+0.001);
            PT:=TF[W,27];
            LSFD(PT,LOADATE,LOADSFT,DATEFIN,SFTFIN);
            TESTIME(LOADATE,DATEFIN,MM);
            TNO[MM]:=TNO[MM]+1;
            TEMPRET[MM,TNO[MM],1]:=TF[W,1];
            TEMPRET[MM,TNO[MM],5]:=DATEFIN;
            TEMPRET[MM,TNO[MM],6]:=SFTFIN;
            TF[W,25]:=TF[W,25]+1;
            NEXTOP(W,TF,STF);
            'IF'FINISHED #1'THEN'
              'BEGIN'
                LOADATE:=999;
                LOADSFT:=999;
                'FOR'D:=26'STEP'2'UNTIL'32'DO'
                 'IF'TF[W,D]>0,001'THEN'
```

```
                    'BEGIN'
                      TESTMAC:=ENTIER(TF[W,D]+0.001);
                      'IF'TEMPRET[TESTMAC,TNO[TESTMAC],5]'LT'LOADATE'THEN'
                        'BEGIN'
                          LOADATE:=TEMPRET[TESTMAC,TNO[TESTMAC],5];
                          LOADSFT:=TEMPRET[TESTMAC,TNO[TESTMAC],6];
                          MM:=TESTMAC;
                          TESTIME(LOADATE,LOADATE,MM);
                          PT:=TF[W,D+1];
                        'END'
                      'ELSE'
                      'IF'TEMPRET[TESTMAC,TNO[TESTMAC],5]'EQ'LOADATE'THEN'
                        'BEGIN'
                          'IF'TEMPRET[TESTMAC,TNO[TESTMAC],6]'LT'LOADSFT
                                                              'THEN'
                            'BEGIN'
                              LOADSFT:=TEMPRET[TESTMAC,TNO[TESTMAC],6 ];
                              MM:=TESTMAC;
                              PT:=TF[W,D+1];
                            'END';
                        'END';
                    'END';
                  TF[W,26]:=MM;
                           TESTIME(LOADATE,LOADATE,MM);
                  TF[W,27]:=PT;
                  'GOTO'DX3;
                'END'
        'ELSE'
        'IF'SETFIN=1'THEN'ADJUST:=0'ELSE'ADJUST:=1;
        ALATE:=TF[W,2]-(DATEFIN+ADJUST);
        'IF'ALATE'GE'0'THEN'ALATE:=0
        'ELSE'
        ALATE:=-ALATE/TF[W,4];
        SCORE[A]:=SCORE[A]+ALATE;
DX5:
```

```
        'END';
'END';
'PROCEDURE'LSFD(PT,STDAY,STSFT,FNDAY,FNSFT);'REAL'PT;'INTEGER'STDAY,STSF
T,FNDAY,FNSFT;
'BEGIN'
      DAYS:=ENTIER(PT);
      REMSFT:=PT-DAYS;
      REMSFT:=REMSFT/0.50;
      FNDAY:=STDAY+DAYS;
      FNSFT:=STSFT+REMSFT;
      'IF'FNSFT'GT'2'THEN'
         'BEGIN'
            FNDAY:=FNDAY+1;
            FNSFT:=FNSFT-2;
         'END'
      'ELSE'
      FNSFT:=FNSFT;
'END';
'PROCEDURE'SCAN(RAY1,Q,RAY2,NM,NA);
   'INTEGER''ARRAY'RAY2;
   'REAL''ARRAY'RAY1;
   'INTEGER'Q,NM,NA;
'BEGIN'
    'FOR'S:=1'STEP'1'UNTIL'Q'DO'
     'BEGIN'
       'IF'NM=0'THEN'
             'BEGIN'
                'IF'ABS(RAY1[S,1]-RAY2[NA,1])'LE'0.001'THEN'SC:=S
                'ELSE'
                DUMMY:=DUMMY;
              'END'
       'ELSE'
       'IF'ABS(RAY1[S,1]-RAY2[NM,NA,1])'LE'0.001'THEN'SC:=S
       'ELSE'
       DUMMY:=DUMMY;
```

```
        'END';
        S:=SC;
'END';
VV:=19;
                FM[71]:=4;
                FM[72]:=3;
                FM[73]:=1;
                FM[74]:=7;
                FM[75]:=6;
                FM[76]:=5;
                FM[77]:=3;
                FM[78]:=2;
                FM[79]:=1;
                FM[80]:=7;
                MAXDWN:=260;
                FLAGYEAR:=READ;
                'IF'FM[FLAGYEAR]=1'THEN'MAXDWN:=265;
                'IF'FLAGYEAR/4=ENTIER(FLAGYEAR/4)'THEN'
                                'BEGIN'
                                   'IF'FM[FLAGYEAR]=2'THEN'MAXDWN:=265;
                                'END';
              'FOR'M:=1'STEP'1'UNTIL'VV'DO'
                'BEGIN'
                 PRINT(M,5,0);
                 NEWLINE(2);
                 'FOR'D:=1'STEP'2'UNTIL'5'DO'
                 'BEGIN'
                     OFF[M,D]:=READ;
                     OUTDATE(OFF[M,D]);
                   SPACE(4);
                     OFF[M,D+1]:=READ;
                     OUTDATE(OFF[M,D+1]);
                   NEWLINE(1);
                 'END';
                 NEWLINE(5);
```

```
            'END';
          PAPERTHROW;
        'FOR'M:=1'STEP'1'UNTIL'VV'DO'
         'BEGIN'
            DNO[M]:=READ;
            DERD[M]:=READ;
            NEWLINE(6);
            WRITETEXT('('MACHINE'('C')')');
            PRINT(M,3,0);
           WRITETEXT('('('2C')''('8S')'JOB.%%%OPERATION'('6S')'LSSD'('15S
')'LSFD'('10S')'BEGUN?'('1C')')');
            'FOR'N:=1'STEP'1'UNTIL'DNO[M]'DO'
            'BEGIN'
             NEWLINE(1);
             SPACE(6);
             'FOR'D:=1'STEP'1'UNTIL'7'DO'
                  DIESFILE[M,N,D]:=READ;
             'FOR'D:=1,2'DO'
               PRINT(DIESFILE[M,N,D],5,0);
             SPACE(4);
             OUTDATE(DIESFILE[M,N,3]);
             PRINT(DIESFILE[M,N,4],4,0);
             SPACE(4);
             OUTDATE(DIESFILE[M,N,5]);
             PRINT(DIESFILE[M,N,6],4,0);
             'IF'N=1'THEN'
               'BEGIN'
                 'IF'DIESFILE[M,1,7]=1'THEN'WRITETEXT('('%%%%YES')')
                 'ELSE'
                  WRITETEXT('('%%%%%NO')');
               'END';
           'END'
         'END';
        PAPERTHROW;
        Q:=READ;
```

```
        TT:=Q;
        'FOR'S:=1'STEP'1'UNTIL'Q'DO'
          'BEGIN'
            'FOR'D:=1'STEP'1'UNTIL'4'DO'
               MASTER[S,D]:=READ;
            'FOR'D:=5'STEP'1'UNTIL'24'DO'
              'FOR'DD:=1'STEP'1'UNTIL'4'DO'
                SUBMAST[S,D,DD]:=0;
            'FOR'D:=5'STEP'2'UNTIL'23'DO'
              'BEGIN'
                X:=NEXTCH;
                'IF'X=CODE('('F')')'THEN''GOTO'SKIP12;
                'FOR'DD:=1'STEP'1'UNTIL'4'DO'
                  'BEGIN'
                    SUBMAST[S,D,DD]:=READ;
                    SUBMAST[S,D+1,DD]:=READ;
                  'END';
              'END';
SKIP12:     SKIPCH;
            'FOR'D:=25'STEP'1'UNTIL'27'DO'
               MASTER[S,D]:=READ;
          'END';
        OUTPUTMAST;
        PAPERTHROW;
        SCHEDRUN:=READ;
        'IF'SCHEDRUN=1'THEN'
         'BEGIN'
        'FOR'M:=1'STEP'1'UNTIL'VV'DO'
         'IF'DNO[M]=0'THEN'NN[M]:=0
         'ELSE'
           'FOR'N:=1'STEP'1'UNTIL'DNO[M]'DO'
             'BEGIN'
               'IF'N=1'THEN'
                 'BEGIN'
                   'IF'DIESFILE[M,N,7]=1'THEN'
```

```
                                'BEGIN'
                                    NN[M]:=1;
                                    'GOTO'DAA;
                                'END'
                            'ELSE'
                            NN[M]:=0;
                        'END'
                    'ELSE'
                    DUMMY:=DUMMY;
                    'FOR'D:=1'STEP'1'UNTIL'7'DO'
                    DIESFILE[M,N,D]:=0;
DAA:            'END';
        'END';
            DCUDATE:=READ;
            MCUDATE:=READ;
            YCUDATE:=READ;
            MONTH2DAY(DCUDATE,MCUDATE,YCUDATE);
            CUDATE:=DWN;
            'IF'FLAGYEAR'LT'YCUDATE'THEN'CUDATE:=CUDATE+MAXDWN;
            'IF'CUDATE'GE'391'THEN'
          'BEGIN'
            CUDATE:=CUDATE-MAXDWN;
            'FOR'M:=1'STEP'1'UNTIL'VV'DO'
              'FOR'N:=1'STEP'1'UNTIL'DNO[M]'DO'
                'FOR'D:=3,5'DO'
                  'BEGIN'
                    DIESFILE[M,N,D]:=DIESFILE[M,N,D]-MAXDWN;
                    'IF'DIESFILE[M,N,D]'LE'0'THEN'DIESFILE[M,N,D]:=0;
                  'END';
            'FOR'M:=1'STEP'1'UNTIL'VV'DO'
              'BEGIN'
                DERD[M]:=DERD[M]-MAXDWN;
                'IF'DERD[M]'LE'0'THEN'DERD[M]:=0;
                'FOR'D:=1'STEP'1'UNTIL'6'DO'
                  'BEGIN'
```

```
                    OFF[M,D]:=OFF[M,D]-MAXDWN;
                    'IF'OFF[M,D]'LE'0'THEN'OFF[M,D]:=0;
                  'END';
                'END';
              'FOR'S:=1'STEP'1'UNTIL'Q'DO'
                'FOR'D:=2,3'DO'
                  'BEGIN'
                    MASTER[S,D]:=MASTER[S,D]-MAXDWN;
                    'IF'MASTER[S,D]'LE'0'THEN'MASTER[S,D]:=0;
                  'END';
              'FOR'F:=1'STEP'1'UNTIL'FF'DO'
                'BEGIN'
                  HOLDDD[F,2]:=HOLDDD[F,2]-MAXDWN;
                  'IF'HOLDDD[F,2]'LE'0'THEN'HOLDDD[F,2]:=0;
                'END';
              FLAGYEAR:=FLAGYEAR+1;
              MAXDWN:=260;
              'IF'FM[FLAGYEAR]=1'THEN'MAXDWN:=265;
              'IF'FLAGYEAR/4=ENTIER(FLAGYEAR/4)'THEN'
                  'BEGIN'
                    'IF'FM[FLAGYEAR]=2'THEN'MAXDWN:=265;
                  'END';
            'END';

'COMMENT' MODIFIED DIES CONTROL BLOCK STARTS HERE .  ******************;


       NUMTODAY:=READ;
      'IF'NUMTODAY=0'THEN''GOTO' DSTOP;
      WRITETEXT('('%%%%%%%%DATE%%%')');
      OUTDATE(CUDATE);
      NEWLINE(3);
      CUSHIFT:=1;
'COMMENT' PUTS 'DUMMY' OPS. IN POSITION N=1 ,WHERE REQUIRED **********;
```

```
          'FOR'M:=1'STEP'1'UNTIL'VV'DO'
            'BEGIN'
              'IF'DIESFILE[M,1,7]#0'THEN''GOTO'DDUM8;
              'FOR'N:=DNO[M]'STEP'-1'UNTIL'1'DO'
                'FOR'D:=1'STEP'1'UNTIL'7'DO'
                  DIESFILE[M,N+1,D]:=DIESFILE[M,N,D];
              'FOR'D:=1'STEP'1'UNTIL'7'DO'
                DIESFILE[M,1,D]:=0;
              DNO[M]:=DNO[M]+1;
DDUM8:      'END';
'COMMENT' READ IN TODAYS RECORDS OF OPERATIONS STARTED.*****          ***;
        'FOR'A:=1'STEP'1'UNTIL'NUMTODAY'DO'
          'BEGIN'
            DWORK[A,1]:=READ;
            DWORK[A,2]:=READ;
            'IF'DWORK[A,1]=0000'THEN'READATE(DWORK[A,3])
            'ELSE' DWORK[A,3]:=READ;
          'END';
        M:=0;
DY:     M:=M+1;
        FOUNDOP:=0;
DZ:     DUMMY:=DUMMY;
'COMMENT'SCAN DWORK FILE FOR AN OP, STARTED ON THE M/C UNDER CONSIDERATI
                                                                  *****;
        'FOR'A:=1'STEP'1'UNTIL'NUMTODAY'DO'
          'BEGIN'
            'IF'DWORK[A,2]=M'THEN'
              'BEGIN'
                FOUNDOP:=FOUNDOP+1;
                'GOTO'DM;
              'END'
            'ELSE'
            'GOTO'DDUM1;
DDUM1:    'END';
        'IF'M<VV'THEN''GOTO'DY
```

```
        'ELSE'
        'GOTO'DSTOP;
'COMMENT' IS THIS A REPORT OF A M/C BREAKDOWN OR A FORCED OP. ?? ******;
DM:     'IF' DWORK[A,3]=-9'THEN''GOTO'DLETE;
        'IF' DWORK[A,1]#DIESFILE[M,2,1]'THEN''GOTO'DDUM6
        'ELSE'
        'BEGIN'
           SCAN(MASTER,Q,DWORK,0,A);
           'IF'DIESFILE[M,2,2]#MASTER[S,25]'THEN'
                          'BEGIN'
                             WRITETEXT('('('4C')''('10S')'ERROR%IN
                                    %INPUT%REGARDING')');
                             PRINT(DWORK[A,1],4,0);
                             WRITETEXT('(',PREVIOUS%OP%NOT%RECORDED%
                                AS%BEING%STARTED%!!!!')');
                             NEWLINE(3);
                             'FOR'D:=1'STEP'1'UNTIL'3'DO'
                                DWORK[A,D]:=-999;
                             'IF'M<10'THEN''GOTO' DY
                             'ELSE'
                             'GOTO' DSTOP;
                          'END'
           'ELSE'
           'GOTO' DW;
        'END';
DDUM6;  'IF'DWORK[A,1]=0000'THEN''GOTO'DHB
        'ELSE'
        'GOTO'DR;
'COMMENT' I.E. NORMAL ADVANCE.*****                                    ******;
DW:      DEHEAD;
        DERD[M]:=0;
        WRITETEXT('('JOB.%NO.')');
        PRINT(DIESFILE[M,2,1],4,0);
        WRITETEXT('('OPERATION%NO.')');
        PRINT(DIESFILE[M,2,2],4,0);
```

```
        WRITETEXT('('STARTED%ON%M/C')');
        PRINT(M,2,0);  NEWLINE(3);
'COMMENT'PUSH THIS OP.INTO POSITION N=1 SLOT (I.E. OP. STARTED ) AND PUS
          ALL FOLLOWING OPS. ON THIS M/C ONE POSITION FORWARD.EVALUATE
          NEXTOP. OF THE JOB JUST REPORTED STARTED AND NOTE IT IN MASTERF
                                                                  ******;
        'FOR'N:=1'STEP'1'UNTIL'DNO[M]-1'DO'
          'FOR'D:=1'STEP'1'UNTIL'7'DO'
            DIESFILE[M,N,D]:=DIESFILE[M,N+1,D];
        DIESFILE[M,1,7]:=1;
        DNO[M]:=DNO[M]-1;
'COMMENT'HAS THIS OP. STARTED TOO EARLY,IF SO, MODIFY THE START AND FINI
  DATES OF N=1 OP. *******                                         ****;
        'IF' DIESFILE[M,1,3]'GT'CUDATE'THEN'
         'BEGIN'
            DIESFILE[M,1,3]:=CUDATE;
            DIESFILE[M,1,4]:=1;
            EVALPT(S,M,1);
            LSED(PT,DIESFILE[M,1,3],DIESFILE[M,1,4],DIESFILE[M,1,5],
                    DIESFILE[M,1,6]);
            TESTIME(DIESFILE[M,1,3],DIESFILE[M,1,5],M);
         'END';
        SCAN(MASTER,Q,DWORK,0,A);
        MASTER[S,25]:=MASTER[S,25]+1;
        NEXTOP(S,MASTER,SUBMAST);
        'IF'FINISHED=1'THEN'
          'BEGIN'
            'FOR'S:=S'STEP'1'UNTIL'Q-1'DO'
            'BEGIN'
              'FOR'D:=1'STEP'1'UNTIL'4'DO'
                MASTER[S,D1]:=MASTER[S+1,D];
              'FOR'D:=25,26,27'DO'
                MASTER[S,D1]:=MASTER[S+1,D];
            'FOR'D:=5'STEP'1'UNTIL'24'DO'
              'FOR'DD:=1'STEP'1'UNTIL'4'DO'
```

```
                SUBMAST[S,D,DD]:=SUBMAST[S,D+1,DD];
            'END';
            Q:=Q-1;
            TT:=0;
          'END';
'COMMENT'DELETE THIS PRODUCTION REPORT. *****                    ******;
        'FOR'D:=1'STEP'1'UNTIL'3'DO'
          DWORK[A,D1:=-999;
        'GOTO' DZ;
'COMMENT' NOW CONSIDER THE CASE WHERE AN OP. HAS BEEN FORCED-IN.*******;
DB:      DEHEAD;
        DERD[M]:=0;
        WRITETEXT('('JOB.%NO.')');
        PRINT(DIESFILE[M,2,1],4,0);
        WRITETEXT('('OPERATION%NO.')');
        PRINT(DIESFILE[M,2,2],4,0);
        WRITETEXT('('NOT%STARTED%ON%M/C')');
        PRINT(M,2,0);
        WRITETEXT('('AS%PER%SCHEDULE,REPLACED%BY%JOB%NO.')');
        PRINT(DWORK[A,1],4,0);  NEWLINE(3);
'COMMENT' IDENTIFY ORIGIN OF'FORCED' OP. ****                    *******;
        SCAN(MASTER,Q,DWORK,0,A);
        X:=MASTER[S,26];
        'FOR'N:=1'STEP'1'UNTIL'DNO[X]'DO'
         'BEGIN'
            'IF'DIESFILE[X,N,1]=DWORK[A,1]'THEN''GOTO'DOUT
            'ELSE'
            DUMMY:=DUMMY;
         'END';
'COMMENT' TRANSFER  'FORCED' OP. TO N=1 POSITION ON M/C,'M'      *******;
DOUT:   HALT:=N;
        'FOR'D:=1'STEP'1'UNTIL'6'DO'
          DIESFILE[M,1,D]:=DIESFILE[X,HALT,D];
        DIESFILE[M,1,7]:=1;
        DIESFILE[M,1,3]:=CUDATE;
```

```
        DIESFILE[M,1,4]:=1;
        EVALPT(S,M,1);
        LSFD(PT,DIESFILE[M,1,3],DIESFILE[M,1,4],DIESFILE[M,1,5],
            DIESFILE[M,1,6]);
      TESTIME(DIESFILE[M,1,3],DIESFILE[M,1,5],M);
'COMMENT'DELETE THIS OP. FROM ITS POSITION ON ITS ORIGINAL M/C.*********;
        'FOR'N:=HALT'STEP'1'UNTIL'DNO[X]-1'DO'
         'FOR'D:=1'STEP'1'UNTIL'7'DO'
          DIESFILE[X,N,D]:=DIESFILE[X,N+1,D];
        DNO[X]:=DNO[X]-1;
'COMMENT' UPDATE THE MASTERFILE ,SINCE THE FORCED OP. HAS BEEN STARTED*;
        SCAN(MASTER,0,DWORK,0,A);
        MASTER[S,25]:=MASTER[S,25]+1;
        NEXTOP(S,MASTER,SUBMAST);
        'IF'FINISHED=1'THEN'
          'BEGIN'
            'FOR'S:=S'STEP'1'UNTIL'Q-1'DO'
            'BEGIN'
              'FOR'D:=1'STEP'1'UNTIL'4'DO'
                MASTER[S,D1]:=MASTER[S+1,D];
              'FOR'D:=25,26,27'DO'
                MASTER[S,D1]:=MASTER[S+1,D];
            'FOR'D:=5'STEP'1'UNTIL'24'DO'
              'FOR'DD:=1'STEP'1'UNTIL'4'DO'
                SUBMAST[S,D,DD]:=SUBMAST[S,D+1,DD];
            'END';
            Q:=Q-1;
            TT:=Q;
          'END';
'COMMENT' DELETE THIS REPORT FROM THE PRODUCTION FILE,'DWORK',*********;
        'FOR'D:=1'STEP'1'UNTIL'3'DO'
          DWORK[A,D1]:=-099;
        'GOTO' DZ;
'COMMENT' CONSIDER THE CASE OF M/C BREAKDOWN,OR,UPDATED ESTIMATE OF
                                                          E.R.D.***;
```

```
DHB:    DERD[M1:=DWORK[A,3];
        WRITETEXT('('M/C')');
        PRINT(M,2,0);
        WRITETEXT('('BROKEN%DOWN,E.R.D.%IS')');
        SPACE(4);
        OUTDATE(DWORK[A,3]);
        NEWLINE(3);
'COMMENT' DELETE THIS REPORT FROM THE PRODUCTION FILE DWORK.**********;
        'FOR'D1:=1'STEP'1'UNTIL'3'DO'
          DWORK[A,D1]:=-999;
        'GOTO' DY;
'COMMENT' FOLLOWING BLOCK DELETES ALL RECORD OF A JOB ,AS REQUESTED BY
                                                              USER ***;
DLETE:  WRITETEXT('('ALL%RECORD%OF%JOB%NO.')');
        PRINT(DWORK[A,1],4,0);
        WRITETEXT('('DELETED%JOB%CANCELLED,MATL. %UNAVAILABLE% E.T.C
          ....')');
        NEWLINE(3);
        'BEGIN'
          'INTEGER''ARRAY'NXN[1:10];
DL1:      'FOR'MM:=1'STEP'1'UNTIL'VV'DO'
            NXN[MM1:=99;
          MARK:=0;
          'FOR'MM:=1'STEP'1'UNTIL'VV'DO'
            'FOR'N:=1'STEP'1'UNTIL'DNO[MM]'DO'
              'BEGIN'
               'IF'DIESFILE[MM,N,1]=DWORK[A,1]'THEN'
                  'BEGIN'
                    NXN[MM1:=N;
                    MARK:=1;
                  'END';
              'END';
          'IF'MARK=0'THEN''GOTO'DL2;
          'FOR'MM:=1'STEP'1'UNTIL'VV'DO'
              'BEGIN'
```

```
                'IF'NXN[MM]=1'THEN'
                  'BEGIN'
                    'FOR'D:=1'STEP'1'UNTIL'7'DO'
                      DIESFILE[MM,1,D]:=0;
                  'END'
                'ELSE'
                  'BEGIN'
                   'FOR'N:=NXN[MM]'STEP'1'UNTIL'DNO[MM]-1'DO'
                    'FOR'D:=1'STEP'1'UNTIL'7'DO'
                      DIESFILE[MM,N,D]:=DIESFILE[MM,N+1,D];
                   'IF'NXN[MM]#99'THEN'DNO[MM]:=DNO[MM]-1;
                  'END';
              'END';
            'GOTO'DL1;
DL2:        SCAN(MASTER,Q,DWORK,0,A) ;
            'FOR'S:=S'STEP'1'UNTIL'Q-1'DO'
              'BEGIN'
              'FOR'D:=1'STEP'1'UNTIL'4'DO'
                MASTER[S,D]:=MASTER[S+1,D];
              'FOR'D:=25,26,27'DO'
                MASTER[S,D]:=MASTER[S+1,D];
              'FOR'D:=5'STEP'1'UNTIL'24'DO'
                'FOR'DD:=1'STEP'1'UNTIL'4'DO'
                  SUBMAST[S,D,DD]:=SUBMAST[S,D+1,DD];
              'END';
            Q:=Q-1;
            TT:=0;
          'END';
          'FOR'D:=1'STEP'1'UNTIL'3'DO'
            DWORK[A,D]:=-999;
          'GOTO'DZ;
'COMMENT' REMOVES ANY 'DUMMY' OPS, REMAINING IN POSITIONS N=1 .********;
          'FOR'M:=1'STEP'1'UNTIL'VV'DO'
            'BEGIN'
              'IF'DIESFILE[M,1,1]#0'THEN''GOTO'DDUM9;
```

```
                'FOR'N:=1'STEP'1'UNTIL'DNO[M]-1'DO'
                  'FOR'D:=1'STEP'1'UNTIL'7'DO'
                    DIESFILE[M,N,D]:=DIESFILE[M,N+1,D];
                DNO[M]:=DNO[M]-1;
DDUM9:        'END';
DSTOP:


'COMMENT' END OF MODIFIED CONTROL BLOCK . ********************************;


      PAPERTHROW;
      WRITETEXT('('%%%%%%%DATE%%%')');
      OUTDATE(CUDATE);
      NEWLINE(3);
      CUSHIFT:=0;
      CUDATE:=CUDATE+1;
      TESTEPSD:=1;
      T:=TT;
      NODIES:=READ;
      'FOR'S:=0+1'STEP'1'UNTIL'Q+NODIES'DO'
        'BEGIN'
            T:=S;
              MASTER[S,1]:=READ;
              READATE(INT);
              MASTER[S,2]:=INT;
              READATE(INT);
              MASTER[S,3]:=INT;
              MASTER[S,4]:=READ;
            'FOR'D:=5'STEP'1'UNTIL'24'DO'
              'FOR'DD:=1'STEP'1'UNTIL'4'DO'
                SUBMAST[S,D,DD]:=0;
            'FOR'D:=5'STEP'2'UNTIL'23'DO'
              'BEGIN'
                X:=NEXTCH;
```

```
                    'IF'X=CODE('('F')')'THEN''GOTO'SKIP13;
                    'FOR'DD:=1'STEP'1'UNTIL'4'DO'
                      'BEGIN'
                          SUBMAST[S,D,DD]:=READ;
                          SUBMAST[S,D+1,DD]:=READ;
                      'END';
                  'END';
SKIP13:       SKIPCH:
              MASTER[S,25]:=1;
          'END';
            'FOR'M:=1'STEP'1'UNTIL'VV'DO'
              DIFSETLE[M,0,5]:=0;
            MASTER[0,1]:=-9999;
            MASTER[0,2]:=9999;
            'FOR'D:=3'STEP'1'UNTIL'33'DO'
              MASTER[0,D]:=0;
            'FOR'D:=5'STEP'1'UNTIL'24'DO'
              'FOR'DD:=1'STEP'1'UNTIL'4'DO'
                SUBMAST[0,D,DD]:=0;
            Q:=TT:=T:
            'IF'SCHEDRUN=1'THEN'
              'BEGIN'
                'FOR'S:=1'STEP'1'UNTIL'Q'DO'
                  'FOR'D:=5'STEP'2'UNTIL'23'DO'
                    'FOR'DD:=1'STEP'1'UNTIL'4'DO'
                      SUBMAST[S,D,DD]:=ABS(SUBMAST[S,D,DD]);
              'END';
NEWLINE(2);
        'FOR'S:=1'STEP'1'UNTIL'TT'DO'
          NEXTOP(S,MASTER,SUBMAST);
        'FOR'S:=1'STEP'1'UNTIL'TT'DO'
          GNDMAST[S,25]:=MASTER[S,25];
        OUTPUTMAST;
        PAPERTHROU;
        'IF'SCHEDRUN=0'THEN''GOTO' SKIPC;
```

```
'COMMENT'  READ IN DUE DATES OF DIES , PREVIOUSLY EVALUATED BY FORGE SCH
                                  RUN .                  *****************;
        FF:=READ;
        'FOR'F:=1'STEP'1'UNTIL'FF'DO'
          'BEGIN'
            HOLDDD[FF,1]:=READ;
            HOLDDD[FF,2]:=READ;
          'END';
      'FOR'S:=1'STEP'1'UNTIL'TT'DO'
        'BEGIN'
          'FOR'F:=1'STEP'1'UNTIL'FF'DO'
            'IF'MASTER[S,1]=HOLDDD[F,1]'THEN'
              'BEGIN'
                MASTER[S,2]:=HOLDDD[F,2];
                'GOTO' SKIPA;
              'END';
SKIPA:  'END';
'COMMENT'**********SCHEDULING PROPER NOW BEGINS. **********************;
DBB:  CUSHIFT:=CUSHIFT+1;
      'IF'CUSHIFT=3'THEN'
        'BEGIN'
          CUDATE:=CUDATE+1;
          CUSHIFT:=1;
        'END'
      'ELSE'
      CUDATE:=CUDATE;
      M:=0;
DCC:  M:=M+1;
      TRYDATE:=CUDATE;
      TESTIME(TRYDATE,TRYDATE,M);
      'IF'TRYDATE#CUDATE'THEN''GOTO'DHH;
      'IF'DIESFILE[M,NN[M],5]-CUDATE'LE'-1'THEN''GOTO'DX1
      'ELSE'
      'IF'DIESFILE[M,NN[M],5]-CUDATE'EQ'0'THEN'
        'BEGIN'
```

```
            'IF'CUSHIFT'GE'DIESFILE[M,NN[M],6]'THEN''GOTO'DX1
            'ELSE'
            'GOTO'DHH;
         'END'
      'ELSE'
      'GOTO'DHH;
DX1:  'BEGIN'
         'IF'DERD[M]'LE'CUDATE'THEN''GOTO'DDD
         'ELSE'
         'GOTO'DHH;
      'END';
DHH:  'IF'M'GE'VV'THEN''GOTO'DBB
      'ELSE'
      'GOTO'DCC;
'COMMENT'**********HAVING FOUND A M/C THAT HAS RETURNED , EVALUATES DYNAM
                   PRIORITY FACTORS FOR EVER JOB REQ. THIS M/C FOR ITS NE
                                                          OPERATION**;
DDD:  FIND:=0;
     'BEGIN''REAL''ARRAY' DYNAPRI[1:120],STF[1:2,5:24,1:4],TF[1:2,1:331;
           'INTEGER''ARRAY'TEMPRET[1:20,0:45,1:6];
           'FOR'MM:=1'STEP'1'UNTIL'VV'DO'
             'FOR'D:=1'STEP'1'UNTIL'6'DO'
              TEMPRET[MM,0,D]:=0;
        'FOR'S:=1'STEP'1'UNTIL'TT'DO'
          'BEGIN'
            'IF'ENTIER(MASTER[S,26]+0.001)=M'THEN''GOTO'DPP
            'ELSE'
            'IF'ENTIER(MASTER[S,28]+0.001)=M'THEN'
                                  'BEGIN'
                                     HOLD26:=MASTER[S,26];
                                     HOLD27:=MASTER[S,27];
                                     MASTER[S,26]:=MASTER[S,28];
                                     MASTER[S,27]:=MASTER[S,29];
                                     MASTER[S,28]:=HOLD26;
                                     MASTER[S,29]:=HOLD27;
```

```
                                    'GOTO'DPP;
                                'END'
            'ELSE'
            'IF'ENTIER(MASTER[S,30]+0.001)=M'THEN'
                                'BEGIN'
                                    HOLD26:=MASTER[S,26];
                                    HOLD27:=MASTER[S,27];
                                    MASTER[S,26]:=MASTER[S,30];
                                    MASTER[S,27]:=MASTER[S,31];
                                    MASTER[S,30]:=HOLD26;
                                    MASTER[S,31]:=HOLD27;
                                    'GOTO'DPP;
                                'END'
            'ELSE'
            'IF'ENTIER(MASTER[S,32]+0.001)=M'THEN'
                                'BEGIN'
                                    HOLD26:=MASTER[S,26];
                                    HOLD27:=MASTER[S,27];
                                    MASTER[S,26]:=MASTER[S,32];
                                    MASTER[S,27]:=MASTER[S,33];
                                    MASTER[S,22]:=HOLD26;
                                    MASTER[S,33]:=HOLD27;
                                    'GOTO'DPP;
                                'END'
            'ELSE'
             'BEGIN'
                 DYNAPRI[S]:=9999;
                 'GOTO'DEE;
             'END';
DPP:        FIND:=FIND+1;
            'FOR'MM:=1'STEP'1'UNTIL'VV'DO'
               'BEGIN'
                   TNO[MM]:=NN[MM];
                   'FOR'N:=0'STEP'1'UNTIL'TNO[MM]'DO'
                      'FOR'D:=1'STEP' 1'UNTIL'6'DO'
```

```
                                    TEMPRET[MM,N,D]:=DIESFILE[MM,N,D];
                        'END';
                      'FOR'D:=1'STEP'1'UNTIL'4'DO'
                        TF[1,D1:=MASTER[S,D];
                      'FOR'D:=25'STEP'1'UNTIL'33'DO'
                        TF[1,D1:=MASTER[S,D];
              'FOR'D:=5'STEP'1'UNTIL'24'DO'
                 'FOR'DD:=1'STEP'1'UNTIL'4'DO'
                      STF[1,D,DD1:=SUBMAST[S,D,DD];
DX4:                  TESTEPSD:=0;
                      LEGALST(1,TF,TEMPRET,STF);
                      TESTEPSD:=1;
                    STDAY:=LEGALDAY;
                    STSFT:=LEGALSFT;
                    MM:=ENTIER(TF[1,26]+0.001);
                    PT:=TF[1,27];
                    LSFD(PT,STDAY,STSFT,DATEFIN,SFTFIN);
                    TESTIME(STDAY,DATEFIN,MM);
                    TNO[MM]:=TNO[MM]+1;
                    TEMPRET[MM,TNO[MM],1]:=TF[1,1];
                    TEMPRET[MM,TNO[MM],5]:=DATEFIN;
                    TEMPRET[MM,TNO[MM],6]:=SFTFIN;
                    TF[1,25]:=TF[1,25]+1;
                    NEXTOP(1,TF,STF);
              'IF'FINISHED #1'THEN'
                   'BEGIN'
                      LOADATE:=999;
                      LOADSFT:=999;
                      'FOR'D:=26'STEP'2'UNTIL'32'DO'
                       'IF'TF[1,D1>0.001'THEN'
                        'BEGIN'
                          TESTMAC:=ENTIER(TF[1,D]+0.001);
                          'IF'TEMPRET[TESTMAC,TNO[TESTMAC],5]'LT'LOADATE'THEN'
                             'BEGIN'
                                LOADATE:=TEMPRET[TESTMAC,TNO[TESTMAC],5];
```

```
                        LOADSFT:=TEMPRET[TESTMAC,TNO[TESTMAC],6];
                        MM:=TESTMAC;
                        TESTIME(LOADATE,LOADATE,MM);
                        PT:=TF[1,D+1];
                    'END'
                'ELSE'
                'IF'TEMPRET[TESTMAC,TNO[TESTMAC],5]'EQ'LOADATE'THEN'
                    'BEGIN'
                      'IF'TEMPRET[TESTMAC,TNO[TESTMAC],6]'LT'LOADSFT
                                                              'THEN'
                        'BEGIN'
                          LOADSFT:=TEMPRET[TESTMAC,TNO[TESTMAC],6 ];
                          MM:=TESTMAC;
                          TESTIME(LOADATE,LOADATE,MM);
                          PT:=TF[1,D+1];
                        'END';
                    'END';
              'END';
            TF[1,26]:=MM;
            TF[1,27]:=PT;
            'GOTO'DX4;
        'END'
    'ELSE'
            'IF'SFTFIN=1'THEN'ADJUST:=0
            'ELSE'
            ADJUST:=1;
            DYNAPRI[S]:=TF[1,2]-(DATEFIN+ADJUST);
            'IF'DYNAPRI[S]'GE'0'THEN'DYNAPRI[S]:=DYNAPRI[S]*TF[1,4]
            'ELSE'
            DYNAPRI[S]:=DYNAPRI[S]/TF[1,4];
            SS:=S;
DEE:    'END';
      'IF'FIND=0'THEN''GOTO'XX
      'ELSE'
      'IF'FIND=1'THEN'
```

```
            'BEGIN'
               'FOR'MM:=1'STEP'1'UNTIL'VV'DO'
                  TNO[MM]:=NN[MM];
               NN[M]:=NN[M]+1;
               DNO[M]:=NN[M];
               LEGALST(SS,MASTER,DIESFILE,SUBMAST);
               DATEON:=LEGALDAY;
               SHIFTON:=LEGALSFT;
               'GOTO'XX;
            'END'
         'ELSE'
'COMMENT'NEXT 24 LINES,OF REMAINING JOBS REQUIRING THIS HAMMER,CHOOSE
  *******THE 3 MOST CRITICAL (LOWES PRIORIT FACTOR)**********************;
      CNT:=0;
      SEQ[1]:=SEQ[2]:=SEQ[3]:=0;
FF:   CNT:=CNT+1;
      MINPRI:=999;
      'FOR'S:=1'STEP'1'UNTIL'TT'DO'
         'BEGIN'
            'IF'S=SEQ[1]'THEN''GOTO'GG'ELSE'
            'IF'S=SEQ[2]'THEN''GOTO'GG'ELSE'
            'IF'S=SEQ[3]'THEN''GOTO'GG'ELSE'
            'IF'DYNAPRI[S]<MINPRI'THEN'
               'BEGIN'
                  MINPRI:=DYNAPRI[S];
                  SEQ[CNT]:=S;
               'END'
            'ELSE'
            MINPRI:=MINPRI;
GG:      'END';
      'IF'CNT<3'THEN'
         'BEGIN'
            'IF'CNT<FIND'THEN''GOTO'FF
            'ELSE'
            CNT:=CNT;
```

```
            'END'
        'ELSE'
        CNT:=CNT;
XX:
        'END';
        'IF'FIND=0'THEN''GOTO'DHH
        'ELSE'
        'IF'FIND=1'THEN''GOTO'DII
        'ELSE'
        DPERM(1,1,2,3);
        DPERM(2,1,3,2);
        DPERM(3,2,3,1);
        DPERM(4,2,1,3);
        DPERM(5,3,1,2);
        DPERM(6,3,2,1);
'COMMENT'NEXT 19 LINES, FIND THE ARRANGEMENT GIVING THE LOWES PENALTY
  *******SCORE,NOTE THE FIRST JOB AT THE HEAD OF THIS ARRANGEMENT(SS)**;
        'IF'FIND'GE'3'THEN'LIMIT:=6'ELSE'LIMIT:=4;
        BESTJOB:=0;
        MINSCR:=999;
        'FOR'A:=1'STEP'1'UNTIL'LIMIT'DO'
           'BEGIN'
              'IF'SCORE[A]<MINSCR'THEN'
                 'BEGIN'
                    MINSCR:=SCORE[A];
                    BESTJOB:=A;
                 'END'
              'ELSE'
              MINSCR:=MINSCR;
           'END';
        'IF'BESTJOB=1'THEN'X:=1'ELSE'
        'IF'BESTJOB=2'THEN'X:=1'ELSE'
        'IF'BESTJOB=3'THEN'X:=2'ELSE'
        'IF'BESTJOB=4'THEN'X:=2'ELSE'
        X:=3;
```

```
SS:=SEQ[X];
'FOR'MM:=1'STEP'1'UNTIL'VV'DO'
   TNO[MM]:=NN[MM];
NN[M]:=NN[M]+1;
DNO[M]:=NN[M];
LEGALST(SS,MASTER,DIESFILE,SUBMAST);
DATEON:=LEGALDAY;
SHIFTON:=LEGALSFT;
'IF'DATEON=CUDATE'EQ'0'THEN''GOTO'DII
'ELSE'
'FOR'S:=1'STEP'1'UNTIL'TT'DO'
  'BEGIN'
        'IF'ENTIER(MASTER[S,26]+0.001)=M'THEN''GOTO'DX11
        'ELSE'
        'IF'ENTIER(MASTER[S,28]+0.001)=M'THEN'
                              'BEGIN'
                                 HOLD26:=MASTER[S,26];
                                 HOLD27:=MASTER[S,27];
                                 MASTER[S,26]:=MASTER[S,28];
                                 MASTER[S,27]:=MASTER[S,29];
                                 MASTER[S,28]:=HOLD26;
                                 MASTER[S,29]:=HOLD27;
                                 'GOTO'DX11;
                              'END'
        'ELSE'
        'IF'ENTIER(MASTER[S,30]+0.001)=M'THEN'
                              'BEGIN'
                                 HOLD26:=MASTER[S,26];
                                 HOLD27:=MASTER[S,27];
                                 MASTER[S,26]:=MASTER[S,30];
                                 MASTER[S,27]:=MASTER[S,31];
                                 MASTER[S,30]:=HOLD26;
                                 MASTER[S,31]:=HOLD27;
                                 'GOTO'DX11;
                              'END'
```

```
                'ELSE'
                'IF'ENTIER(MASTER[S,32]+0.001)=M'THEN'
                                'BEGIN'
                                   HOLD26:=MASTER[S,26];
                                   HOLD27:=MASTER[S,27];
                                   MASTER[S,26]:=MASTER[S,32];
                                   MASTER[S,27]:=MASTER[S,33];
                                   MASTER[S,22]:=HOLD26;
                                   MASTER[S,33]:=HOLD27;
                                   'GOTO'DX11;
                                'END'
                'ELSE'
                'GOTO'DX12;
DX11:           'BEGIN'
                  LEGALST(S,MASTER,DIESFILE,SUBMAST);
                  INSDATE:=LEGALDAY;
                  INSHIFT:=LEGALSFT;
                  PT:=MASTER[S,27];
                  LSED(PT,INSDATE,INSHIFT,FNDAY,FNSFT);
                  TESTIME(INSDATE,FNDAY,M);
                    'IF'FNDAY'LT'DATEON'THEN'
                       'BEGIN'
                           DATEON :=INSDATE;
                           SHIFTON:=INSHIFT;
                           SS:=S;
                           'GOTO'DII;
                       'END'
                    'ELSE'
                    'IF'FNDAY=DATEON'THEN'
                       'BEGIN'
                          'IF'FNSFT'LE'SHIFTON'THEN'
                             'BEGIN'
                                DATEON:=INSDATE;
                                SHIFTON:=INSHIFT;
                                SS:=S;
```

```
                                    'GOTO'DII;
                                  'END'
                                'ELSE'
                                SS:=SS;
                              'END'
                            'ELSE'
                            SS:=SS;
                    'END';
DX12:   'END';
DII:
'COMMENT'*********LOAD NEXT OPERATION OF CHOSEN JOB ON REQ. M/C *******;
      DIESFILE[M,NN[M],1]:=MASTER[SS,1];
      DIESFILE[M,NN[M],2]:=MASTER[SS,25];
      LSFD(MASTER[SS,27],DATEON,SHIFTON,FNDAY,FNSFT);
      TESTIME(DATEON,FNDAY,M);
      DIESFILE[M,NN[M],3]:=DATEON;
      DIESFILE[M,NN[M],4]:=SHIFTON;
      DIESFILE[M,NN[M],5]:=FNDAY;
      DIESFILE[M,NN[M],6]:=FNSFT;
      DIESFILE[M,NN[M],7]:=0;
      'FOR'DD:=1'STEP'1'UNTIL'4'DO'
        'IF'ENTIER(SUBMAST[SS,(ENTIER(MASTER[SS,25]+0.001)*2+3),DD]+
0.001)#ENTIER(MASTER[SS,26]+0.001)'THEN'
          SUBMAST[SS,(ENTIER(MASTER[SS,25]+0.001)*2+3),DD]:=
         -SUBMAST[SS,(ENTIER(MASTER[SS,25]+0.001)*2+3),DD];
      MASTER[SS,25]:=MASTER[SS,25]+1;
      NEXTOP(SS,MASTER,SUBMAST);
      'IF'FINISHED=1'THEN'
      'BEGIN'
        'FOR'D:=1'STEP'1'UNTIL'4'DO'
          'BEGIN'
            GNDMAST[SS,D]:=MASTER[SS,D];
            MASTER[SS,D]:=0;
          'END';
        'FOR'D:=5'STEP'1'UNTIL'24'DO'
```

```
            'FOR'DD:=1'STEP'1'UNTIL'4'DO'
              'BEGIN'
                SUBGND[SS,D,DD]:=SUBMAST[SS,D,DD];
                SUBMAST[SS,D,DD]:=0;
              'END';
          NEXTOP(SS,GNDMAST,SUBGND);
          'FOR'D:=25'STEP'1'UNTIL'33'DO'
           MASTER[SS,D]:=0;
          T:=T-1;
      'END'
      'ELSE'
      T:=T;
      'IF'T>0'THEN''GOTO'DHH
      'ELSE'
      DUMMY:=DUMMY;
SKIPC:
      'FOR'M:=1'STEP'1'UNTIL'VV'DO'
       'BEGIN'
          NEWLINE(6);
          WRITETEXT('('MACHINE'('C')'')');
          PRINT(M,3,0);
         WRITETEXT('('('('2C')''('8S')'JOB.%%%OPERATION'('6S')'LSSD'('15S
')'LSFD'('10S')'BEGUN?'('1C')'')');
          'FOR'N:=1'STEP'1'UNTIL'DNO[M]'DO'
            'BEGIN'
             NEWLINE(1);
             SPACE(6);
            'FOR'D:=1,2'DO'
             PRINT(DIESFILE[M,N,D],5,0);
            SPACE(4);
            OUTDATE(DIESFILE[M,N,3]);
            PRINT(DIESFILE[M,N,4],4,0);
            SPACE(4);
            OUTDATE(DIESFILE[M,N,5]);
            PRINT(DIESFILE[M,N,6],4,0);
```

```
            'IF'N=1'THEN'
              'BEGIN'
                'IF'DIESFILE[M,1,7]=1'THEN'WRITETEXT('('%%%%YES')')
                'ELSE'
                WRITETEXT('('%%%%%NO')');
              'END';
            'END';
        'END';
      'IF'SCHEDRUN=0'THEN''GOTO' SKIPD;
         'FOR'S:=1'STEP'1'UNTIL'TT'DO'
          SORT[S]:=GNDMAST[S,1];
LOOP: CNT:=0;
      'FOR'S:=2'STEP'1'UNTIL'TT'DO'
         'IF'SORT[S]<SORT[S-1]'THEN'
          'BEGIN'
            HOLD:=SORT[S-1];
            SORT[S-1]:=SORT[S];
            SORT[S]:=HOLD;
            CNT:=CNT+1;
          'END';
      'IF'CNT#0'THEN''GOTO'LOOP;
'COMMENT' JOBS ARE NOW ORDERED ***********                                    **;
      PAPERTHROW;
      WRITETEXT('('%%JOB.%%%%%OP.%%%%M/C.'('7S')'LSSD.'('12S')'LSFD.'('1
'1S')'REQ.DD.%%%%BEST%DD.%%%%%PUNCT.'('C')')');
      'FOR'T:=1'STEP'1'UNTIL'TT'DO'
         'BEGIN'
          S:=T;
          MARK:=0;
          SQ:=0;
A1:       SQ:=SQ+1;
          'FOR'M:=1'STEP'1'UNTIL'VV'DO'
            'FOR'N:=1'STEP'1'UNTIL'DNO[M]'DO'
              'BEGIN'
                'IF'DIESFILE[M,N,1]=SORT[S]'THEN'
```

```
                    'BEGIN'
                      'IF'DIESFILE[M,N,2]=SQ'THEN''GOTO'PRNT;
                    'END'
                  'ELSE'
                  DUMMY:=DUMMY;
                'END';
              'IF'MARK=0'THEN''GOTO'A1
              'ELSE'
              'GOTO'A2;
PRNT:       NEWLINE(1);
            MARK:=MARK+1;
            'IF'MARK=1'THEN'
              'BEGIN'
                PRINT(DIESFILE[M,N,1],5,0);
                SPACE(2);
              'END'
            'ELSE'
            SPACE(10);
            PRINT(DIESFILE[M,N,2],2,0);
            'IF'DIESFILE[M,N,7]=1'THEN'WRITETEXT('('*')')
            'ELSE'
            SPACE(1);
            PRINT(M,3,0);
            SPACE(3);
            OUTDATE(DIESFILE[M,N,3]);
            PRINT(DIESFILE[M,N,4],3,0);
            SPACE(3);
            OUTDATE(DIESFILE[M,N,5]);
            PRINT(DIESFILE[M,N,6],3,0);
            XN:=N;
            MM:=M;
            'GOTO'A1;
A2:         SCAN(GNDMAST,TT,DIESFILE,MM,XN);
            SPACE(6);
            INT:=GNDMAST[S,2];
```

```
            OUTDATE(INT);
            'IF'DIESFILE[MM,XN,6]=1'THEN'ADJUST:=0
            'ELSE'
            ADJUST:=1;
            PUNCT:=GNDMAST[S,2]-(DIESFILE[MM,XN,5]+ADJUST);
        HOLDATA[T,1]:=GNDMAST[S,1];
        HOLDATA[T,2]:=(DIESFILE[MM,XN,5]+ADJUST);
        HOLDATA[T,3]:=PUNCT;
            PUNCT:=PUNCT/GNDMAST[S,4];
            SPACE(4);
            INT:=DIESFILE[MM,XN,5]+ADJUST;
            OUTDATE(INT);
            PRINT(PUNCT,8,0);
            NEWLINE(3);
          'END';
        PAPERTHROW;
        WRITETEXT('('('50S')'DIES%LIKELY%TO%BE%LATE.'('C')''('50S')'
-----------------------'('3C')''('10S')'JOB%NO.'('15S')'SCHED.%FINISH%
DATE'('15S')'DAYS%LATE%(UNWEIGHTED)'('3C')'')');
        'FOR'T:=1'STEP'1'UNTIL'TT'DO'
          'BEGIN'
            'IF'HOLDATA[T,3]<0'THEN'
              'BEGIN'
                SPACE(9);
                PRINT(HOLDATA[T,1],5,0);
                SPACE(20);
                OUTDATE(HOLDATA[T,2]);
                SPACE(22);
                PRINT(HOLDATA[T,3],3,0);
                NEWLINE(2);
              'END';
          'END';
        PAPERTHROW;
SKIPD;
'END'
```

Appendix XIV

Listing of computer simulation programme.

```
'BEGIN'
   'INTEGER'XBRR,DUMMY,DIST,Z,F,
S,J,SS,N,NOTESTS,FLAG;
   'REAL'T,PRBB,UPPERPROB,LOWERPROB,MINSP,STREAM,RANDOM,
        XVALUE,TOTPROF,PER,X,UPPERX,LOWERX,INTER,MINACSP,MINACPR,
MINACFW,NWT,MINSPPR,MINSPFW;
    'INTEGER''ARRAY'NOJOBS[1:6],NEGJOBS[1:6];
'REAL''ARRAY' PSTDEV[1:6],FSTDEV[1:6],TABLE[0:41,1:2],SP[1:6,1:300]
,TRUEMANCST[1:300],PROFIT[1:6,1:300],SUMPROFIT[1:6],PROB[1:6,0:22,
1:2],RAND[1:6],PRODRATE[1:6],PBIAS[1:6],FBIAS[1:6],DEBIT[1:6],
FLASHW[1:6],XBAR[1:300,1:2];
'REAL''PROCEDURE'FPMCRV(STREAM);'REAL'STREAM;'EXTERNAL';
       'PROCEDURE'GENDIST(XBRR,STDAV,DIST);
       'REAL'STDAV,XBRR; 'INTEGER'DIST;
              'BEGIN'
                  'REAL'SD;
                 'IF'STDAV=0'THEN''GOTO'FIN;
                  SD:=(XBRR*STDAV)/100;
                  INTER:=(6*SD)/20;
                  N:=0;
                 'FOR'X:=XBRR-2*SD'STEP'INTER'UNTIL'
                                        (XBRR+4*SD)+0.001'DO'
                     'BEGIN'
                        N:=N+1;
                        'IF'X<XBRR'THEN'T:=ABS(X-XBRR)/(0.5*SD)'ELSE'
                        T:=ABS(X-XBRR)/SD;
                        'FOR'Z:=1'STEP'1'UNTIL'41'DO'
                          'IF'T'GE'TABLE[Z,1]'THEN''GOTO'FOUNDPROB
                          'ELSE'
                          DUMMY:=DUMMY;
FOUNDPROB:              'IF'T-TABLE[Z,1]>TABLE[Z-1,1]-T'THEN'Z:=Z-1
                        'ELSE'
                        Z:=Z;
                        PRBB:=TABLE[Z,2];
                        'IF'X'LE'XBRR'THEN'
```

```
                    PROB[DIST,N,2]:=PRBB
                    'ELSE'
                    PROB[DIST,N,2]:=(100.0-PRBB);
                    PROB[DIST,N,1]:=X;
                    'IF'X'LE'XBRR*0.05'THEN'PROB[DIST,N,1]:=XBRR*0.05;
                      'END';
                    PROB[DIST,22,1]:=PROB[DIST,21,1];
                    PROB[DIST,22,2]:=100.0;
FIN:
              'END';
       'PROCEDURE'SAMPLEDIST(XBRR,STDAV,DIST);
'REAL'STDAV,XBRR;
'INTEGER'DIST;
              'BEGIN'
                   'IF'STDAV=0'THEN'
                    'BEGIN'
                      PRODRATE[DIST]:=XBRR;
                      'GOTO' NIF;
                    'END';
                   FPMCRV(STREAM);
                   RANDOM:=STREAM*100.0;
                   'IF'RANDOM=0.0000'THEN'RANDOM:=0.000001;
                   RAND[DIST]:=RANDOM;
                    'FOR'N:=1'STEP'1'UNTIL'21'DO'
                   'IF'RANDOM'LE'PROB[DIST,N,2]'THEN''GOTO'FOUNDX
                   'ELSE'DUMMY:=DUMMY;
FOUNDX:            UPPERX:=PROB[DIST,N,1];
                   LOWERX:=PROB[DIST,N-1,1];
                   UPPERPROB:=PROB[DIST,N,2];
                   LOWERPROB:=PROB[DIST,N-1,2];
                   XVALUE:=(RANDOM-LOWERPROB)/(UPPERPROB-LOWERPROB);
                   XVALUE:=((UPPERX-LOWERX)*XVALUE)+LOWERX;
                   PRODRATE[DIST]:=XVALUE;
NIF:
              'END';
```

```
      STREAM:=READ;
     'COMMENT'READ IN THE NORMAL TABLE ********************************;
     TABLE[0,1]:=5;
     TABLE[0,2]:=0.0;
     'FOR'Z:=1'STEP'1'UNTIL'41'DO'
        'BEGIN'
            TABLE[Z,1]:=READ;
            TABLE[Z,2]:=READ;
        'END';
'FOR'Z:=0'STEP'1'UNTIL'41'DO'
    'BEGIN'
        SPACE(3);
        PRINT(TABLE[Z,1],1,1);
        SPACE(10);
        PRINT(TABLE[Z,2],2,3);
        NEWLINE(1);
    'END';
PAPERTHROW;
     FLAG:=0;
     'COMMENT'READ IN THE VARIONS DISTRIBUTION PARAMETERS************;
     NOTESTS:=READ;
     'FOR'J:=1'STEP'1'UNTIL'NOTESTS'DO'
            XBAR[J,1]:=READ;
     'FOR'J:=1'STEP'1'UNTIL'NOTESTS'DO'
            XBAR[J,2]:=READ;
STRT:
'COMMENT' READ IN BIAS      VALUES*************************************;
     'FOR'S:=1'STEP'1'UNTIL'6'DO'
        PBIAS[S]:=READ;
     'IF'PBIAS[1]=-99.9'THEN''GOTO'ENND;
     'FOR'S:=1'STEP'1'UNTIL'6'DO'
        FBIAS[S]:=READ;
     'COMMENT'READ IN THE SD OF THE VARIOUS EST. PROCEDURES**********;
     'FOR'S:=1'STEP'1'UNTIL'6'DO'
            PSTDEV[S]:=READ;
```

```
      'FOR'S:=1'STEP'1'UNTIL'6'DO'
            FSTDEV[S]:=READ;
PAPERTHROW;
'IF'FLAG=1'THEN''GOTO'JUMP1;
WRITETEXT('('''('41S')'SUPPLIER'('C')''('41S')'========='('C')''('16S')'
1'('10S')'2'('10S')'3'('10S')'4'('10S')'5'('10S')'6'('C')''('14S')'
ST.DEV'('5S')'ST.DEV'('5S')'ST.DEV'('5S')'ST.DEV'('5S')'ST.DEV'('5S')'
ST.DEV'('6S')'N.B.%ALL%PRICES%ETC.,IN''POUNDS''.'('C')''('14S')'(RATE)
%%%%%(RATE)%%%%%(RATE)%%%%%(RATE)%%%%%(RATE)%%%%%(RATE)'('C')''('10S')''
)');
'FOR'S:=1'STEP'1'UNTIL'6'DO'
  'BEGIN'
    SPACE(3);
    PRINT(FSTDEV[S],2,2);
  'END';
WRITETEXT('('''('2C')''('14S')'ST.DEV%%%%%ST.DEV%%%%%ST.DEV%%%%%ST.DEV
%%%%%ST.DEV%%%%%ST.DEV'('C')''('14S')'(F.W.)%%%%%(F.W.)%%%%%(F.W.)
 %%%%%(F.W.)%%%%%(F.W.)%%%%%(F.W.)'('C')''('10S')''')');
'FOR'S:=1'STEP'1'UNTIL'6'DO'
  'BEGIN'
    SPACE(3);
    PRINT(FSTDEV[S],2,2);
  'END';
NEWLINE(2);
WRITETEXT('('''('13S')'PR.%BIAS'('3S')'PR.%BIAS'('3S')'PR.%BIAS'('3S')'
PR.%BIAS'('3S')'PR.%BIAS'('3S')'PR.%BIAS'('C')''('11S')''')');
'FOR'S:=1'STEP'1'UNTIL'6'DO'
  'BEGIN'
    SPACE(2);
    PRINT(PBIAS[S],2,3);
  'END';
NEWLINE(1);
WRITETEXT('('''('13S')'FW.%BIAS'('3S')'FW.%BIAS'('3S')'FW.%BIAS'('3S')'
FW.%BIAS'('3S')'FW.%BIAS'('3S')'FW.%BIAS'('C')''('11S')''')');
'FOR'S:=1'STEP'1'UNTIL'6'DO'
```

```
  'BEGIN'
    SPACE(2);
    PRINT(FBIAS[S],2,3);
  'END';
WRITETEXT('('''('8S')'BASE'('4S')'CHOSEN'('1S')'SUPPLIER%%PROFIT'('C')'
%%%JOB'('79S')'%SP%'('6S')'SP'('12S')'PER%1000'('C')'')');
JUMP1:
     'COMMENT'BEGIN SIMULATION PROPER ***********************************;
     J:=0;
     TOTPROF:=0;
     'FOR'S:=1'STEP'1'UNTIL'6'DO'
           'BEGIN' NOJOBS[S]:=0;
                   NEGJOBS[S]:=0;
                   DEBIT[S]:=0;
                   SUMPROFIT[S]:=0;
                   PROB[S,0,1]:=0;
                   PROB[S,0,2]:=0;
           'END';
     'COMMENT'CONSIDER FIRST/NEXT JOB***********************************;
ADRS:J:=J+1;
     NWT:=3*XBAR[J,2];
     MINACPR:=(XBAR[J,1]*(100+2*PSTDEV[5])/100)*((100+PBIAS[5])/100);
     MINACFW:=(XBAR[J,2]*(100-1*FSTDEV[6])/100)*((100+FBIAS[5])/100);
     MINSPPR:=(1.46/MINACPR)+(0.76/MINACPR)+(2.7/MINACPR)+(2.8/MINACPR)+
              ((NWT+XBAR[J,2])*0.053);
     MINSPFW:=(1.46/XBAR[J,1])+(0.76/XBAR[J,1])+(2.7/XBAR[J,1])+(2.8/
              XBAR[J,1])+((NWT+MINACFW)*0.053);
MINACSP:=(1.46/MINACPR)+(0.76/MINACPR)+(2.7/MINACPR)+(2.8/MINACPR)
        +((NWT+MINACFW)*0.053);
     'FOR'S:=1'STEP'1'UNTIL'6'DO'
       'BEGIN'
'COMMENT' FOR CONTINUITY,WHEN SIMULATING P.R. ONLY..FIRST CALL DUMMY
        RANDOM. WHEN SIMULATING FLASH ONLY ..CALL DUMMY RANDOM
          AFTER 'GENDIST(.....  ***********************************;
        GENDIST(XBAR[J,2],FSTDEV[S],S);
```

```
            SAMPLEDIST(XBAR[J,2],PSTDEV[S],S);
            FLASHW[S]:=XVALUE*((100+PBIAS[S])/100);
            GENDIST(XBAR[J,1],PSTDEV[S],S);
            SAMPLEDIST(XBAR[J,1],PSTDEV[S],S);
            PRODRATE[S]:=XVALUE*((100+PBIAS[S])/100);
SP[S,J]:=(NWT+FLASHW[S])*0.053+(1.46/PRODRATE[S])+(0.76/PRODRATE[S])
                    +(2.7/PRODRATE[S])+(2.80/PRODRATE[S]);
        'END';
    'COMMENT' FOLLOWING ROUTINE FINDS LOWEST SP AND 'CUSTOMER' CHOOSES
              THIS SUPPLIER ***************************************;
    MINSP:=99999;
    'FOR'S:=1'STEP'1'UNTIL'6'DO'
        'BEGIN'
            'IF'SP[S,J]'LE'MINSP'THEN'
              'BEGIN'
                MINSP:=SP[S,J];
                SS:=S;
              'END'
            'ELSE'
            MINSP:=MINSP;
        'END';
    'COMMENT' FOLLOWING ROUTINE EVALUATES THE EFFECT OF THIS JOB ON
    THE PROFITABILITY OF EACH SUPPLIER*****************************;
'IF'FLASHW[SS]'LT'MINACFW'THEN'
    'BEGIN'
        'IF'PRODRATE[SS]'LE'MINACPR'THEN'
        MINACSP:=MINSPFW;
    'END';
'IF'PRODRATE[SS]'GT'MINACPR'THEN'
    'BEGIN'
        'IF'FLASHW[SS]'GE'MINACFW'THEN'
        MINACSP:=MINSPPR;
    'END';
    PROFIT[SS,J]:=(SP[SS,J]- MINACSP )*1000;
    'FOR'S:=1'STEP'1'UNTIL'6'DO'
```

```
            'IF'S#SS'THEN'PROFIT[S,J]:=0
            'ELSE'
             PROFIT[S,J]:=PROFIT[SS,J];
        'FOR'S:=1'STEP'1'UNTIL'6'DO'
            SUMPROFIT[S]:=SUMPROFIT[S]+PROFIT[S,J];
            NOJOBS[SS]:=NOJOBS[SS]+1;
            'IF'PROFIT[SS,J]>0'THEN'TOTPROF:=TOTPROF+PROFIT[SS,J]
            'ELSE'
            NEGJOBS[SS]:=NEGJOBS[SS]+1;
'IF'FLAG=1'THEN''GOTO' SKIP;
NEWLINE(1);
WRITETEXT('('('3S')'---'('7S')'--------%%%--------%%%--------%%%
--------%%%--------%%%-------%%%%%%%%------%%%------%%%%---%%%
--------'('C')''')');
SPACE(2);
PRINT(J,2,0);
WRITETEXT('('EST.PR.')');
'FOR'S:=1'STEP'1'UNTIL'6'DO'
  'BEGIN'
    PRINT(PRODRATE[S],3,0);
    SPACE(5);
  'END';
SPACE(3);
PRINT(MINACSP,1,4);
PRINT(SP[SS,J],1,4);
SPACE(1);
PRINT(SS,2,0);
SPACE(1);
PRINT(PROFIT[SS,J],3,2);
NEWLINE(1);
SPACE(7);
WRITETEXT('('EST.FW.')');
'FOR'S:=1'STEP'1'UNTIL'6'DO'
  'BEGIN'
    PRINT(FLASHW[S],1,4);
```

```
    SPACE(2);
  'END';
NEWLINE(2);
WRITETEXT('('''('7S')'EST.SP')');
'FOR'S:=1'STEP'1'UNTIL'6'DO'
  'BEGIN'
    PRINT(SP[S,J],1,4);
    SPACE(2);
  'END';
WRITETEXT('('''('C')''('85S')' ''ACHIEVED''%PR.=')');
PRINT(XBAR[J,1],3,0);
WRITETEXT('('''('C')''('85S')''ACHIEVED''%FW.=')');
PRINT(XBAR[J,2],1,4);
NEWLINE(3);
'GOTO' SKIP;
SELECT OUTPUT(4);
        'FOR'N:=1'STEP'1'UNTIL'22'DO'
          'BEGIN'
            'FOR'S:=1'STEP'1'UNTIL'6'DO'
              'BEGIN'
                'IF'S=1'THEN'
                  'BEGIN'
                    'IF'PROB[S,N,2]=50.0'THEN'
                      WRITETEXT('('**')')
                    'ELSE'
                      SPACE(2);
                  'END';
                'IF'ESTDEV[S]=0'THEN'
                  'BEGIN'
                    PRINT(XBAR[J,1],3,0);
                    WRITETEXT('('%100.000%%')');
                  'END'
                'ELSE'
                  'BEGIN'
                    PRINT(PROB[S,N,1],3,0);
```

```
                    PRINT(PROB[S,N,2],3,3);
                  'END';
                'END';
                NEWLINE(1);
              'END';
            NEWLINE(3);
SELECT OUTPUT (0);
SKIP:   'IF'J'LT'NOTESTS'THEN''GOTO'ADRS;
FLAG:=1;
PAPERTHROW;
NEWLINE(12);
WRITETEXT('('%SUPPLIER'('9S')'1'('10S')'2'('10S')'3'('10S')'4'('10S')'5
        '('10S')'6'('C')''('17S')'---'('8S')'---'('8S')'---'('8S')'---
        '('8S')'---'('8S')'---'('2C')'')');
WRITETEXT('('%ST.DEV%(PR.)%%')');
'FOR'S:=1'STEP'1'UNTIL'6'DO'
  'BEGIN'
    PRINT(PSTDEV[S],2,2);
    SPACE(3);
  'END';
NEWLINE(1);
WRITETEXT('('%PR.BIAS'('7S')'')');
'FOR'S:=1'STEP'1'UNTIL'6'DO'
  'BEGIN'
    PRINT(PBIAS[S],2,3);
    SPACE(2);
  'END';
NEWLINE(2);
WRITETEXT('('%ST.DEV%(FW.)%%')');
'FOR'S:=1'STEP'1'UNTIL'6'DO'
  'BEGIN'
    PRINT(FSTDEV[S],2,2);
    SPACE(3);
  'END';
NEWLINE(1);
```

```
WRITETEXT('('%FW.BIAS'('7s')'')');
'FOR'S:=1'STEP'1'UNTIL'6'DO'
  'BEGIN'
    PRINT(PBIAS[S],2,3);
    SPACE(2);
  'END';
NEWLINE(2);
WRITETEXT('('%NUMBER%JOBS%%%')');
'FOR'S:=1'STEP'1'UNTIL'6'DO'
  'BEGIN'
    PRINT(NOJOBS[S],2,0);
    SPACE(6);
  'END';
NEWLINE(1);
WRITETEXT('('%%PER.%CENT.%%%')');
'FOR'S:=1'STEP'1'UNTIL'6'DO'
  'BEGIN'
    PER:=(NOJOBS[S]/NOTESTS)*100;
    PRINT(PER,3,1);
    SPACE(3);
  'END';
NEWLINE(2);
WRITETEXT('('%TOT.%PROFIT%%')');
'FOR'S:=1'STEP'1'UNTIL'6'DO'
  'BEGIN'
    PRINT(SUMPROFIT[S],4,1);
    SPACE(2);
  'END';
NEWLINE(1);
WRITETEXT('('%%PER.%CENT.%%%')');
'IF'TOTPROF=0'THEN'TOTPROF:=1000000;
'FOR'S:=1'STEP'1'UNTIL'6'DO'
  'BEGIN'
    PER:=(SUMPROFIT[S]/TOTPROF)*100;
    PRINT(PER,3,1);
```

```
    SPACE(3);
  'END';
NEWLINE(2);
WRITETEXT('('%%AVE%PROF/JOB%')');
'FOR'S:=1'STEP'1'UNTIL'6'DO'
  'BEGIN'
    'IF'NOJOBS[S]=0'THEN'WRITETEXT('('%%%0.0%%')')
    'ELSE'
    PRINT((SUMPROFIT[S])/NOJOBS[S],3,1);
    SPACE(3);
  'END';
NEWLINE(2);
WRITETEXT('('%NUM.%NEG.%JOBS')');
'FOR'S:=1'STEP'1'UNTIL'6'DO'
  'BEGIN'
    PRINT(NEGJOBS[S],2,0);
    SPACE(6);
  'END';
'GOTO'STRT;
ENND:
'END'
```

Appendix XV

Example of detailed 'LP' output produced by computer simulation programme.

	1	2	3	4	5	6
	ST.DEV (RATE) 21.00	ST.DEV (RATE) 16.00	ST.DEV (RATE) 21.00	ST.DEV (RATE) 21.00	ST.DEV (RATE) 21.00	ST.DEV (RATE) 21.00
	ST.DEV (F.W.) 47.00	ST.DEV (F.W.) 35.00	ST.DEV (F.W.) 47.00	ST.DEV (F.W.) 47.00	ST.DEV (F.W.) 47.00	ST.DEV (F.W.) 47.00
	PR. BIAS -7.000	PR. BIAS 0.000	PR. BIAS -7.000	PR. BIAS -7.000	PR. BIAS -7.000	PR. BIAS -7.000
	FW. BIAS 22.000	FW. BIAS -1.000	FW. BIAS 22.000	FW. BIAS 22.000	FW. BIAS 22.000	FW. BIAS 22.000

N.B. ALL PRICES ETC., IN ''POUNDS''.

							BASE SP	CHOSEN SP	SUPPLIER	PROFIT PER 100
EST.PR.	69	99	87	81	83	99	0.0874	0.1115	6	24.03
EST.FW.	0.2242	0.2222	0.1697	0.1545	0.1703	0.1861				
EST.SP	0.1482	0.1135	0.1218	0.1270	0.1261	0.1115				

''ACHIEVED'' PR.= 100
''ACHIEVED'' FW.= 0.1500

EST.PR.	159	141	108	108	138	114	0.0862	0.0913	1	5.09
EST.FW.	0.2063	0.1922	0.1682	0.3508	0.2356	0.1870				
EST.SP	0.0913	0.0968	0.1125	0.1219	0.1003	0.1097				

''ACHIEVED'' PR.= 123
''ACHIEVED'' FW.= 0.2000

EST.PR.	74	77	64	79	79	69	0.2934	0.3237	4	30.27
EST.FW.	1.1678	1.7621	1.6683	0.9008	1.4009	1.5597				
EST.SP	0.3450	0.3714	0.3868	0.3237	0.3505	0.3729				

''ACHIEVED'' PR.= 76
''ACHIEVED'' FW.= 1.1200

Appendix XVI

Specimen documents and records pertinent to production scheduling.

…DUCTION CONTROL

DIE No. A3311

…SCRIPTION	PART No.	SPEC.	TREATMENT/INSP.
…ERING ARM (R.H.)	D?7560, 22Z	EN16T	HT, SB, 100% BRN. + CRACK DETECT.

…T	WT. GRS.	SIZE	WEIGHT OF STEEL
10 CWT	2.098 oz	1 1/8" dia	

…DER No.	QTY.	DATE	DELIVERY
02447	16,000		

…TOMER

LOTUS CARS LTD.

NORWICH

NORFOLK NOR 92W

DELIVER TO

DIES | STEEL

PRODUCTION			SCRAP			DESPATCH		
…ATE	QTY.	HRS.	DATE	QTY.	HRS.	QTY.	DATE	TOTAL
1.9.72	480	11 1/4						
12.9.72	4090	12 3/4						
	4570							
13.9.72	5063	8 1/2						
	9623							
14.9.72	2685	9						
	12308							
15.9.72	3730	6 1/2						
	16038							

…ATE	LETTER	CODE	IMP.	

…TUAL PROD. RATE 341

ACTUAL GROSS WT.

Figure A4.

	Quantity/Die No.		Scrap	Shift	
1	264 752	A2194 A3006		X	K.O. 10.15 Set dies 1hr. Tools out ½hr.
0					
1	596 804	A1750 A1750	2	6-2 X 2-10	Reset dies 1¼hrs. No heat in furnace ½hr
2	800 160 1470	A3261 A887 A887		6-2 " X 2-10	K.O. 11.15 Set dies 1hr. Tools ¼hr. Hammer twice ¾hr.
3	1036	A3234		X	Burner ½hr. K.O. 4.00. ½hr
4	1134 465	A3037 A3037		X 5.30-10.00	Waiting for stripper ¼ hr. Hammer ¼hr. Press ¼ hr. Transferred from 13cwt 5.30.
	385 184	A3181 A3069	6	X	Waiting for kicker ½hr. K.O. 11.30 Set dies 1hr. Shaft off ½hr.
	890 252	A2899 A2899	6	6-2 X 2-10	Packing rod ¾hr. Shaft off 1½hrs. Transfered to 10-4cwt 5.30
	870 928	A3065 A3065	4 9	6-2 X 2-10	Tools out ½hr. Ears turned ½hr.
	905 1250	A2523 A2523	2 3	6-2 X 2-10	Hammer 4 times 2 hrs. Hammer twice 1hr.
	618 866	A3005 A3025		6-2 X 2-10	Bearing dies boat 1¼hrs. Gas on 1¼hrs Ear test ¾hr. Automatic ½hr.
	1181 1600	A3175 A3175		6-2 X 2-10	Poppet ½hr. Die off bed ½hr. Die repair twice 1¼hr. Ear test ½hr. Hammer ½hr.
3	101 786	A2920 A2920		2-10 X	Bottom die out, peg stuck in bolster, bolster out 3¼hrs. Hammer 3 times 3¼hrs. Transferred from 30cwt 9.30. K.O. bottom die ½hr.

Figure A5.

FORGE DIE No. A3311

UNIT	SPEC.	CAST LETTER	IMPRESSION No.	QTY.
	EN16T			

STEEL SIZE	STEEL LENGTH	BAR WT. LBS.	No FORGINGS PER BAR
1 1/8" dia	6 7/16		

No. BARS SET	FRG. WT. LBS	FLASH WT. LBS.	TARGET PROD. RATE
	1·171 20	·582	

TEST BAR DETAILS :- 2 TEST BARS 12" LONG

PRODUCTION METHOD PER CAST

HALF ROLL

EDGE

STAMP

CUT OFF

CLIP

CONDITION DIES & TOOLS

ISSUED	RETURNED
DIES	DIES
TOOLS	TOOLS

PRODUCTION EQUIPMENT

CLIP TOOLS ✓ POWER HAMMER

PIERCE TOOLS COMBINATION

HOT SET TOOLS

PAYMENT RATE

100%	110%	120%	130%	140%	150%

REMARKS SAFETY CRITICAL PART

CONT EST 760 PER SHIFT.

Figure 16.

STEEL STORES DIE No. A331

SIZE	SPEC.	LENGTH	QTY.
1⅛"dia	EN18T	4·71in	

CAST LETTER	UNIT	COLOUR CODE	GROSS WT.
		GREEN	2·038

ISSUES

DATE	T	C	Q	L	DATE	T	C	Q	L
ISSUE TOTAL					ISSUE TOTAL				
RETURN					RETURN				
TOTAL USED					TOTAL USED				

WEIGHT CHECK	UNIT			EST. GROSS WT.	DIE No.
				2·038 cc	A331
STAMPING WT. LBS.	EST.		ACTUAL	No FRGS SET	No BARS SET
	1·171				
FLASH WT. LBS.	·592			TAG SIZE	TAG WT. SET
SET WEIGHT	C	Q	LS.		

CHECK GROSS WT.:-

Figure A7.

T. INSPECTION DIE No. A3311

CUSTOMER		PT. No.	
·TUS CARS LTD.		RCM5001337.	
ORDER No.		QUANTITY	CODE LETTER
SPEC.	STEEL CODE	PROD UNIT	DELIVERY REQD.
EN16T.			

HEAT TREATMENT				PROCESS		
BATCH QTY.	TEMP.	METHOD COOL	SOAK	PROCESS	YES NO	%
NORM.				SHOT BLAST	✓	100
ANNEAL				PICKLE		
HARDEN ✓	860	OIL	½	CRACK DETECT	✓	100
TEMPER ✓	640/660	AIR	2½	GRIND		
BRINELL RANGE T				COIN		
% BRINELL	100%			GAUGE		

SPECIAL NOTES TEST CERTIFICATE REQ'D

SAFETY CRITICAL PART

DESCRIPTION STEERING ARM (R.H.)

DELIVER TO
NORWICH
NORFOLK
NOR 92W

DESPATCH			
DATE	QTY.	ADVICE NOTE	CARRIER

CONTINUE DESPATCH DETAILS ON REVERSE SIDE IF NECESSARY.

Figure A8.

HAMMER 18 A DATE 24-11-77 SHIFT 2-10

STARTER D.G. CREW 116 132

JOB NO A3221 QTY 429 SCRAP 9

JOB NO. QTY SCRAP

JOB NO. QTY SCRAP

REASON	TIME ON	TIME OFF	TIME	SIGNATU
TOOLS OUT	2-00			
SET PALLET DIES TOOLS OUT				
HAD TROUBLE WITH DIES ROLLER		5-00		

Figure A9.

Die No.	Steel	Die Hardness	New Dies / Resinks / Repairs
A3013			

Operation	Time	Operator	Comments
Gauges			
Marking Out ✓	2 hrs.		
Facing ✓			
Crank	2½ hrs	Skeffington	
Dovetail			
Mill	10 hrs.		
Turn			
Sink	8 hrs		
Dowel			
Flash etc.	6 hrs		
Finish	1½ hrs		

SIDE

Figure A10.

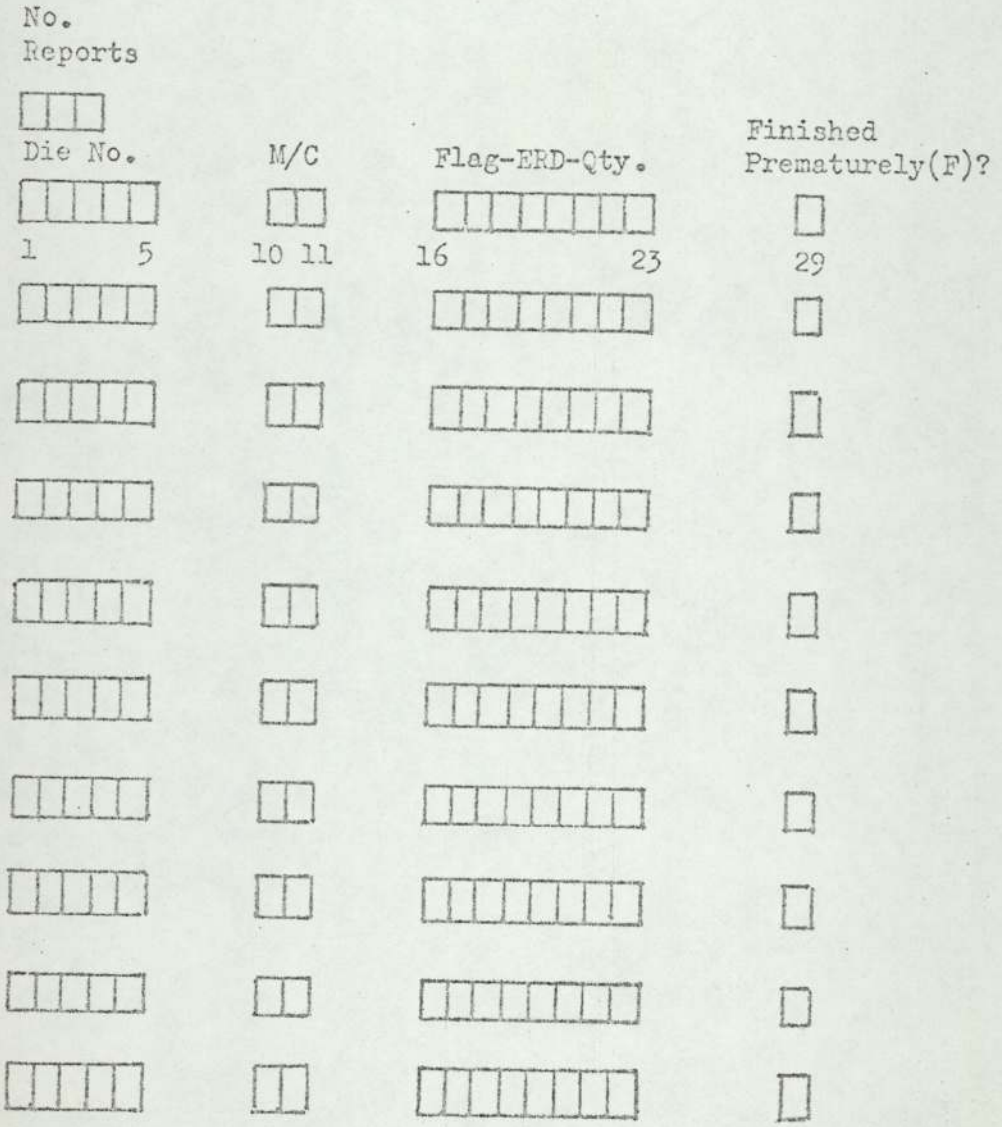


Figure All. Input Document For Daily Forge-Shop Production.

No.
Input

Hammers.

Die No.	Pri.	Qty. Req.	Due Date	EPSD.	Duration	A	B	C	D
1 5	9 10	13 18	21 28	31 38	41 44	49 50	53 54	57 58	61 62

Figure A12. Input Document For New Jobs. (Forge)

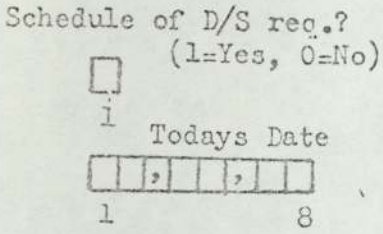


Figure A13. 'Initiate' Document.(Die-Shop and Forge-Shop)

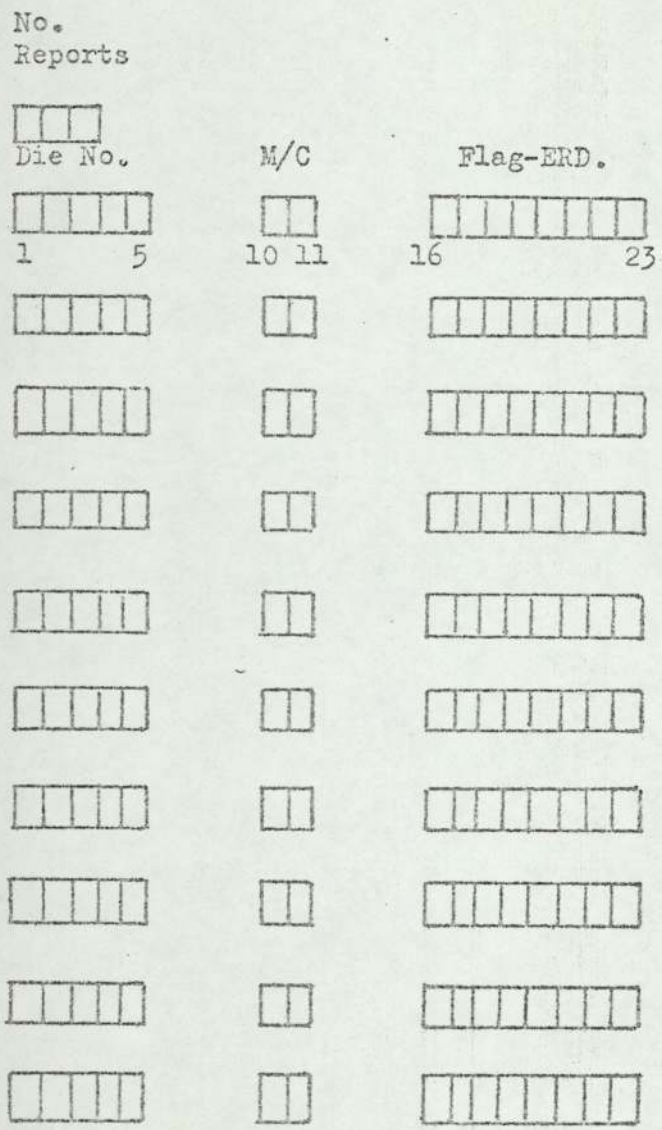


Figure A14. Input Document For Daily Die-Shop Production.

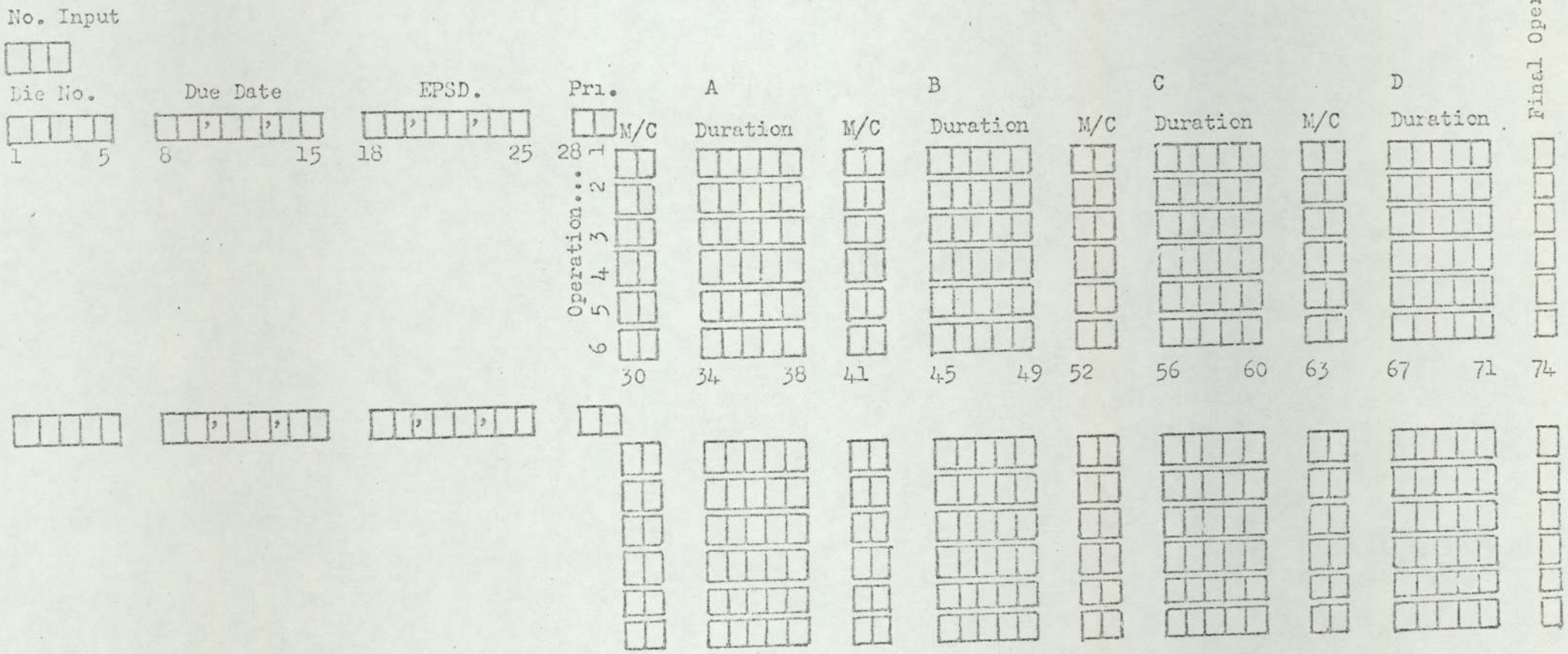


Figure A15. Input Document For New Jobs. (Die-Shop)

ACKNOWLEDGMENTS.

The author wishes to thank Dr. J.L.Aston, the project supervisor, for his guidance throughout the period of this project. The valuable assistance of Dr. S.S.Chang and Mr. T.B.Tate as I.H.D. and associate supervisors, respectfully, is also noted.

My gratitude is also due to the management and staff of Anslow Ltd. for the useful discussion and criticism offered by them.

Finally, the financial support given by both The Science Research Council and Anslow Ltd. must be gratefully acknowledged, without which this project would not have been possible.