

90337

VOLUME II

PROGRAM LISTING

DEVELOPMENT OF MINERAL
INVENTORY DATABASE

FOR

DEPTT. OF MINES/INDIAN BUREAU OF MINES

Thapar Institute of Engg: & Tech:
PATIALA.
CENTRAL LIBRARY

No. Dt.

 ** This program is a main subprogram used for generating the state
 ** wise report. This is used by the program named "statwise".

```
*SET PRINT ON
*SET MARGIN TO 9
CLEAR
SELE 4
GO TOP
SEEK STOCDE
IF EOF()
  CLEAR
  @ 10,10 SAY "THERE IS NO MINERAL / COMMODITY PRESENT IN THIS STATE"
ELSE
  SLCTR = 0
  DO WHILE .NOT. EOF() .AND. STATE = STOCDE
    STORE 0 TO SLCTR1,SKCUNT,SLCTR2
    SHD = 'N'
    SFLAG = 'N'
    SFLAG1 = 'N'
    SMCD = MINERAL
    SFNAME = "MI" + SMCD
    IF SK = 0
      SELE 3
      USE 8SFNAME
      SELE 3
      SET FILTER TO STATE = STOCDE
    ELSE
      SELE 5
      SET FILTER TO MINERAL = SMCD .AND. DIST = SDCODE
    ENDIF
    GO TOP
    IF EOF()
      SELE 4
      IF .NOT. EOF()
        SKIP
        LCCP
      ELSE
        EXIT
      ENDIF
    ELSE
      IF SK = 0
        COPY TO TEMP1
        SELE 6
        USE TEMP1
      ELSE
        COPY TO TEMP2
        SELE 7
        USE TEMP2
      ENDIF
    ENDIF
    SELE 1
    GO TOP
    I = 11
    LOCATE FOR MINERAL=SMCD
    SMINDEX = M_DES
    DO WHILE MINERAL = SMCD
      II=STR(I,2)
      SG8II = G_CODE
      SD8II = G_DES
      SA8II = ACDS
      I = I + 1
    ENDWHILE
  ENDWHILE
```



```

      I = I+1
      SKIP
ENDDC
SI = I
IF SK = C
  ST = '6'
ELSE
  ST = '7'
ENDIF
SELE 8ST
REST FROM 8SMCD ADDITIVE
I = 1
DO WHILE I < 7
  II = STR(I,1)
  IF TCTAL&II = 0
    STORE C.CCCCCC1 TO TCTAL&II
  ENDIF
  I = I+1
ENDDC
UNITDES = SPACE(4)
DO UNIT
IF SLCTR # 0
  IF (SLCTR+50) < 78
    ?
    ? REPLICATE("=",123)
    ? SMINDS,SPACE(9),SUCES
    ? REPLICATE("=",123)
    SLCTR = SLCTR+4
    SLCTR2 = SLCTR
  ELSE
    SLCTR = 0
    ? SPACE(58),"COMPUTERISED BY NATIONAL INFORMATICS CENTRE,"
    ?? " PLANNING COMMISSION"
  ENDIF
ENDIF
STATUS = SPACE(8)
DO WHILE SKCUNT < 19
  DO FILTER
  DO TCP
  IF ECF()
    SKCUNT = SKCUNT + 1
    LCCF
  ENDIF
  I = 1
  DO WHILE I < 21
    J = STR(I,2)
    IF I < 10
      J = STR(I,1)
    ENDIF
    STORE C TO IPRCV&J, IFROB&J, IFCSS&J, RPROV&J, RPRCB&J, RFCSS&J
    I = I + 1
  ENDDC
  STOR C TO IPRCV,IFROE,IFCSS,RPROV,RPRCB,RFCSS,PMPROV,;
  FMPCRB,PMFOSS,SMPROV,SMPCRB,SMPOSS,PRCG,FROS,CTOTAL
  DO WHILE .NOT. ECF()
    IF SK = 0
      SINTR = IS
      SIN = 'IS'
    ELSE
      SINTR = ID
      SIN = 'ID'
    ENDIF
    IF SINTR = '1'
      DO IGRADADD

```



```

      DO RGRADE ADD
      IF FMGRADE # SPACE(32)
        FMPCV = FMPCV + PMPCVED
        FMPCB = FMPCB + PMPCBEABLE
        FMPCSS = FMPCSS + PMPCSSIBLE
      ENDIF
      IF SMGRADE # SPACE(32)
        SMPV = SMPV + SMPVED
        SMPB = SMPB + SMPBEABLE
        SMPSS = SMPSS + SMPSSIBLE
      ENDIF
      PRG = PRG + PROGGTY
      PRCS = PRCS + PROSGTY
    ELSE
      SFLAG = 'Y'
    ENDIF
    SKIP
  ENDDC
  DO ALLADD
  I = 11
  DO WHILE I < SI
    II = STR(I,2)
    IF SA8II = 'N'
      J = VAL(SG8II)
      SJ = STR(J,2)
      IF J < 10
        SJ = STR(J,1)
      ENDIF
      IPRV = IPRV - IPRV&SJ
      IPRB = IPRB - IPRB&SJ
      IPCSS = IPCSS - IPCSS&SJ
      RPRV = RPRV - RPRV&SJ
      RPRB = RPRB - RPRB&SJ
      RPCSS = RPCSS - RPCSS&SJ
    ENDIF
    I = I + 1
  ENDDC
  CTOTAL = IPRV+IPRB+IPCSS+RPRV+RPRB+RPCSS+FMPCV+FMPCB
  CTOTAL = CTOTAL + FMPCSS+SMPV+SMPB+SMPSS+PRG+PRCS
  CTOTAL1 = CTOTAL - (RPRV+RPRB+RPCSS+PRCS+PRG)
  IF SLCTR = 0
    IF SKCNT < 4 .OR. CTOTAL # 0
      *EJECT
      IF SSP = 1
        ? ".FL S6"
        ? ".MT C"
        ? ".MB C"
        ? ".CP"
      ELSE
        ? ".FA"
      ENDIF
      DO STATHEAD
    ENDIF
  ENDIF
  DO CASE
  CASE SKCNT = 0
    IF CTOTAL # 0 .OR. SFLAG = 'N'
      IF SK = 0
        ? "GRAND TOTAL OF STATE "
        ? "-----"
      ELSE
        ? "GRAND TOTAL OF DISTRICT"
        ? "-----"
      ENDIF
    ENDIF
  ENDIF

```



```

      SLCTR = SLCTR + 2
      IF (IPRCV+IPRCE+IFCSS) # 0
        ? "TOTAL RESOURCES (INSITU + CONDITIONAL)"
        ? SPACE(21),STR(IPRCV+FMPCOV+SMPROV,15,3)
        ?? SPACE(1)+STR(IPRCE+FMPCOE+SMPROE,15,3)
        ?? SPACE(1)+STR(IPCSS+FMPCOSS+SMPOSS,15,3)
        ?? SPACE(1)+STR(CTOTAL1,19,3)+SPACE(2)
        ?? "TOTAL OF ALL GRADES"
        SLCTR = SLCTR + 2
      ENDIF
    ELSE
      SFLAG1 = 'Y'
    ENDIF
    IF CTOTAL = 0
      IF SFLAG = 'N'
        ? SPACE(50),"----- NOT ESTIMATED -----"
        ? LINE
        SLCTR = SLCTR+2
      ENDIF
      EXIT
    ENDIF
CASE SKCUNT > 0
  IF SKCUNT < 4
    IF SHC = 'N'
      IF SK = 0
        ? "TOTAL OF STATE"
        ? "-----"
      ELSE
        ? "TOTAL OF DISTRICT"
        ? "-----"
      ENDIF
      SHC = 'Y'
      SLCTR = SLCTR+2
    ENDIF
    ? STATUS
    ?
    SLCTR = SLCTR + 2
    IF CTOTAL = 0
      ? SPACE(50),"----- NOT ESTIMATED -----"
      ? LINE
      SKCUNT = SKCUNT + 1
      SLCTR = SLCTR+2
      LOOP
    ENDIF
  ELSE
    IF CTOTAL # 0
      ? STATUS
      ?
      SLCTR = SLCTR + 2
    ELSE
      SKCUNT = SKCUNT+1
      LOOP
    ENDIF
  ENDIF
ENDCASE
IF IPRCV+IPRCE+IFCSS # 0
  ? "INSITU RESERVES"
  ? SPACE(21),STR(IPROV,15,3),STR(IPRCE,15,3),STR(IFCSS,15,3)
  ?? SPACE(1)+STR(IPROV+IPRCE+IFCSS,19,3)+SPACE(2)
  ?? "TOTAL OF ALL GRADES"
  ? SPACE(29)+"("+STR(IPROV/TOTAL1*100,6,2)+"%)" +SPACE(7)
  ?? "("+STR(IPRCE/TOTAL2*100,6,2)+"%)"
  ?? SPACE(7)+"("+STR(IFCSS/TOTAL3*100,6,2)+"%)" +SPACE(11)
  ?? "("+STR((IPROV+IPRCE+IFCSS)/TOTAL7*100,6,2)+"%)"

```



```

      SLCTR = SLCTR + 2
      IF (IPRCV+IPRCE+IFCSS) # 0
        ? "TOTAL RESOURCES (INSITU + CONDITIONAL)"
        ? SPACE(21),STR(IPRCV+FMPCOV+SMPROV,15,3)
        ?? SPACE(1)+STR(IPRCE+FMPCOE+SMPROE,15,3)
        ?? SPACE(1)+STR(IPCSS+FMPCOSS+SMPOSS,15,3)
        ?? SPACE(1)+STR(CTOTAL1,19,3)+SPACE(2)
        ?? "TOTAL OF ALL GRADES"
        SLCTR = SLCTR + 2
      ENDIF
    ELSE
      SFLAG1 = 'Y'
    ENDIF
    IF CTOTAL = 0
      IF SFLAG = 'N'
        ? SPACE(50),"----- NOT ESTIMATED -----"
        ? LINE
        SLCTR = SLCTR+2
      ENDIF
      EXIT
    ENDIF
CASE SKCNT > 0
  IF SKCNT < 4
    IF SHC = 'N'
      IF SK = 0
        ? "TOTAL OF STATE"
        ? "-----"
      ELSE
        ? "TOTAL OF DISTRICT"
        ? "-----"
      ENDIF
      SHC = 'Y'
      SLCTR = SLCTR+2
    ENDIF
    ? STATUS
    ?
    SLCTR = SLCTR + 2
    IF CTOTAL = 0
      ? SPACE(50),"----- NOT ESTIMATED -----"
      ? LINE
      SKCNT = SKCNT + 1
      SLCTR = SLCTR+2
      LOOP
    ENDIF
  ELSE
    IF CTOTAL # 0
      ? STATUS
      ?
      SLCTR = SLCTR + 2
    ELSE
      SKCNT = SKCNT+1
      LOOP
    ENDIF
  ENDIF
ENDCASE
IF IPRCV+IPRCE+IFCSS # 0
  ? "INSITU RESERVES"
  ? SPACE(21),STR(IPROV,15,3),STR(IPRCE,15,3),STR(IFCSS,15,3)
  ?? SPACE(1)+STR(IPROV+IPRCE+IFCSS,19,3)+SPACE(2)
  ?? "TOTAL OF ALL GRADES"
  ? SPACE(29)+"("+STR(IPROV/TOTAL1*100,6,2)+"%)" +SPACE(7)
  ?? "("+STR(IPRCE/TOTAL2*100,6,2)+"%)"
  ?? SPACE(7)+"("+STR(IFCSS/TOTAL3*100,6,2)+"%)" +SPACE(11)
  ?? "("+STR((IPROV+IPRCE+IFCSS)/TOTAL7*100,6,2)+"%)"

```



```

SLCTR = SLCTR + 3
I = 11
DO WHILE I < SI
  II = STR(I,2)
  J = VAL(SG8II)
  SJ = STR(J,2)
  IF J < 10
    SJ = STR(J,1)
  ENDIF
  IF (IPROV8SJ + IPROE8SJ + IPCSS8SJ) # 0
    ? SPACE(21),STR(IPROV8SJ,15,3),STR(IPROE8SJ,15,3)
    ?? SPACE(1) +STR(IPCSS8SJ,15,3)
    ?? SPACE(1)+STR(IPROV8SJ+IPROE8SJ+IPCSS8SJ,19,3)
    ?? SPACE(2)+SD8II
    ? SPACE(29)+'(' +STR(IPROV8SJ/TOTAL1*100,6,2)+'%)'
    ??SPACE(7)+'(' +STR(IPROE8SJ/TOTAL2*100,6,2)+'%)'
    ?? SPACE(7)+'(' +STR(IPCSS8SJ/TOTAL3*100,6,2)+'%)'+SPACE(11)
    ?? '(' +STR((IPROV8SJ+IPROE8SJ+IPCSS8SJ)/TOTAL7*100,6,2)+'%)'
    SLCTR = SLCTR + 2
  ENDIF
  I = I + 1
ENDDO
ELSE
  IF SKCUNT = 0
    ? 'INSITU RESERVES'
    ? SPACE(50),"----- NOT ESTIMATED -----"
    SLCTR = SLCTR + 2
  ENDIF
ENDIF
IF (RPROV+RPROB+RPOSS) # 0
  ? 'RECOVERABLE RESERVES'
  ?SPACE(21),STR(RPROV,15,3),STR(RPROB,15,3),STR(RPOSS,15,3)
  ?? SPACE(1)+STR(RPROV+RPROB+RPOSS,19,3),SPACE(1)+'TOTAL OF ALL GRADES'
  ? SPACE(29)+'(' +STR(RPROV/TOTAL4*100,6,2)+'%)'+SPACE(7)+'('
  ??STR(RPROB/TOTAL5*100,6,2)+'%)'
  ?? SPACE(7)+'(' +STR(RPOSS/TOTAL6*100,6,2)+'%)'+SPACE(11)
  ?? '(' +STR((RPROV+RPROB+RPOSS)/TOTAL8*100,6,2)+'%)'
  SLCTR = SLCTR + 3
  I = 11
  DO WHILE I < SI
    II = STR(I,2)
    J = VAL(SG8II)
    SJ = STR(J,2)
    IF J < 10
      SJ = STR(J,1)
    ENDIF
    IF (RPROV8SJ + RPROE8SJ + RPOSS8SJ) # 0
      ?SPACE(21),STR(RPROV8SJ,15,3),STR(RPROE8SJ,15,3),STR(RPOSS8SJ,15,3)
      ??SPACE(1)+STR(RPROV8SJ+RPROE8SJ+RPOSS8SJ,19,3),SPACE(1)+SD8II
      ? SPACE(29)+'(' +STR(RPROV8SJ/TOTAL4*100,6,2)+'%)'+SPACE(7)+'(' +STR(RPROE8SJ/T
      ?? SPACE(7)+'(' +STR(RPOSS8SJ/TOTAL6*100,6,2)+'%)'+SPACE(11)
      ?? '(' +STR((RPROV8SJ+RPROE8SJ+RPOSS8SJ)/TOTAL8*100,6,2)+'%)'
      SLCTR = SLCTR + 2
    ENDIF
    I = I + 1
  ENDDO
ENDIF
IF PMPROV+PMPROB+PMPCSS # 0
  ? 'CONDITIONAL (PARAMARGINAL)'
  ? SPACE(21),STR(PMPROV,15,3),STR(PMPROB,15,3),STR(PMPCSS,15,3)
  ?? SPACE(1)+STR(PMPROV+PMPROB+PMPCSS,19,3)+SPACE(2)+'TOTAL OF ALL GRADES'
  SLCTR = SLCTR + 2
ENDIF
IF SMPROV+SMPROB+SMPCCS # 0

```



```

? 'CONDITIONAL (SLEMARGINAL)'
? SPACE(21),STR(SMPROV,15,3),STR(SMPRCE,15,3),STR(SMPCSS,15,3)
?? SPACE(1)+STR(SMPROV+SMPRCB+SMPCSS,19,3)+SPACE(2)+'TOTAL OF ALL GRADES'
SLCTR = SLCTR + 2
ENDIF
IF PRCS # 0
? 'PROSPECTIVE RESCURES'+SPACE(49)+STR(P RCS,19,3)+SPACE(2)+'TCTAL OF ALL GRADES'
SLCTR = SLCTR + 1
ENDIF
IF PRCG # 0
? 'PROGNOSTIC RESCURES'+SPACE(50)+STR(PRCG,19,3)
SLCTR = SLCTR + 1
ENDIF
?LINE
SLCTR = SLCTR + 1
IF SKCUNT = 0
SLCTR1 = SLCTR + 2 - SLCTR2
ENDIF
IF (SLCTR + SLCTR1) > 77
SLCTR = 0
? SPACE(58),'COMPUTERISED BY NATIONAL INFORMATICS CENTRE, PLANNING COMMISSION'
ENDIF
SKCUNT = SKCUNT + 1
ENDDC
IF SFLAG = 'Y'
SET FILTER TO 8SIN # '1'
GC TCF
DC WHILE .NOT. EOF()
DC = DEPOSIT
DC INTRDIST
SELE 8ST
DC WHILE .NOT. EOF() .AND. DEPOSIT # DC
SKIP
ENDDC
ENDDC
SET FILTER TO
IF SFLAG1 = 'Y'
IF SK = 0
? "GRAND TOTAL OF STATE"
?
ELSE
? "GRAND TOTAL OF DISTRICT"
?
ENDIF
? SPACE(50),"----- NOT ESTIMATED -----"
? LINE
SLCTR = SLCTR+3
ENDIF
ENDIF
IF SK = 0
LSE
SELE 5
APPEND FROM TEMP1
ENDIF
SELE 4
SKIP
ENDDC
IF SLCTR # 0
? SPACE(58),'COMPUTERISED BY NATIONAL INFORMATICS CENTRE, PLANNING COMMISSION'
ENDIF
ENDIF
*SET PRINT OFF

```


PROGRAM NAME :- CCATA.prc

```
*****
** This program creates the pre requisite database named 'ccata'
** which keeps only the codes for all the commodities.
*****
```

```
USE CCATA
ZAP
APPEND FROM MINFILE1
GC TCP
DO WHILE .NOT. ECF()
  MIN = MINERAL
  SKIP
  IF .NOT. ECF()
    DO WHILE MINERAL = MIN
      DELETE
      SKIP
    ENDDO
  ENDIF
ENDDO
FACK
USE MINFILE
INDEX ON MINERAL TO MINFILE
USE
```


PROGRAM NAME :- CMDTY.prg

```

*****
** This program is for generating the commodity wise report. this is
** the main program for this report which calls some other sub programs.
*****
*** CLCTR      : LINE COUNTER; CPCTR : PAGE COUNTER          ***
*** CKOUNT     : FOR PROCESSING DIFFERENT STATUSES          ***
*** CSFLAG     : STATE PROCESSING SWITCH, 1 WHEN STATE FRCESSING STAR ***
*** CS1        : IS 1 WHEN STATE IS BEING PROCESSED         ***
*** CCD1       : IS 1 WHEN DISTRICT IS BEING PROCESSED      ***
*** FLAG       : IF 0 SUBHEADINGS ARE PRINTED; CFLAG : IS 1 IF IS > 1 ***
*** DFLAG      : IS 1 IF ID > 1; NFLAG : IS 1 IF TCTAL = 0 AND (IS OR ID =1) ***
***           : INCASE OF CKOUNT = 0; CS : CURRENT STATE CODE ***
*** CD         : CURRENT DISTRICT CODE; CDIST : CURRENT DISTRICT NAME ***
*** CSTATE     : CLURRENT STATE NAME; FN : IS CLURRENT DATA FILE SELECTION NC. ***
*** CLINE      : LONG LINE; CMDTY : CODE OF COMMODITY BEING PROCESSED ***
*** COMPUTER   : FOOT NOTE; CFILEN : NAME OF DATA FILE FOR CURRENT COMMODITY ***
*****
*** INITIALIZE VARIABLES
STOR C TO CLCTR,CPCTR,CKOUNT,CSFLAG,CS1,CCD1,FLAG,CFLAG,DFLAG,NFLAG
STORE SPACE(0) TO CS,CD,CDIST,CSTATE
FN = '1'
CS_DATE = '01/01/1985'
CMDTY = SPACE(3)
CLINE = REPLICATE('-',123)
COMPUTER = 'COMPUTERISED BY NATIONAL INFORMATICS CENTRE, PLANNING COMMISSION'
SET PROC TO MICPROC
*** GET COMMODITY CODE
DO SELECT1
CMDTY = UPPER(CMDTY)
CFILEN = "MI"+CMDTY
*** SELECT FILES
SELE 1
USE &CFILEN
SELE 2
USE MINFILE
SELE 3
USE STATFILE
SELE 4
USE CSTATE
ZAP
SELE 6
USE INTRFILE
SET ALTERNATE TO &CFILEN
*** STORE ALL GRADE OF THE COMMODITY
SELE 2
SET FILTER TO MINERAL = CMDTY
GO TOP
CMIN = M_DES
I = 0
DO WHILE .NOT. ECF()
    I = I+1
    J = STR(I,2)
    IF I < 10
        J = STR(I,1)
    ENDOF
    CG&J = G_CODE
    CD&J = G_DES
    CADD&J = ACDS
    SKIP
ENDDO
*** CGCTR IS THE GRADE COUNT
CGCTR = 1+1

```


*** GET UNIT DESCRIPTION IN CLUNITDES

ENDCASE

CLEAR

*** START PROCESSING

SET ALTERNATE ON

DO WHILE CKOUNT < 20

DO CASE

CASE

UNIT = ' '

CLUNITDES = SPACE(5)

CASE

UNIT = '1'

CLUNITDES = 'CARATS'

CASE

UNIT = '2'

CLUNITDES = 'KILOGRAMS'

CASE

UNIT = '3'

CLUNITDES = 'TONNES'

CASE

UNIT = '4'

CLUNITDES = "CCC"TONNES"

CASE

UNIT = '5'

CLUNITDES = "MILLION TONNES"

ENDCASE

CLEAR

*** START PROCESSING

SET ALTERNATE ON

DO WHILE CKOUNT < 20

DO CASE

CASE

CKCOUNT = 1

SET FILTER TO STATUS_A = 'F'

STATUS = 'FREEHOLD'

CASE

CKCOUNT = 2

SET FILTER TO STATUS_A = 'E'

STATUS = 'FREEHOLD & LEASEHOLD'

CASE

CKCOUNT = 3

SET FILTER TO STATUS_A = 'L'

STATUS = 'LEASEHOLD'

CASE

CKCOUNT = 4

SET FILTER TO STATUS_E = 'L'

STATUS = 'PUBLIC'

CASE

CKCOUNT = 5

SET FILTER TO STATUS_E = 'L' .AND. STATUS_C = 'C'

STATUS = 'PUBLIC CAPTIVE'

CASE

CKCOUNT = 6

SET FILTER TO STATUS_E = 'L' .AND. STATUS_C = 'B'

STATUS = 'PUBLIC CAPTIVE & NON-CAPTIVE'

CASE

CKCOUNT = 7

SET FILTER TO STATUS_E = 'L' .AND. STATUS_C = 'N'

STATUS = 'PUBLIC NON-CAPTIVE'

CASE

CKCOUNT = 8

SET FILTER TO STATUS_E = 'E'

STATUS = 'PUBLIC & PRIVATE'

CASE

CKCOUNT = 9

SET FILTER TO STATUS_E = 'E' .AND. STATUS_C = 'C'

STATUS = 'PUBLIC & PRIVATE CAPTIVE'

CASE

CKCOUNT = 10

SET FILTER TO STATUS_E = 'E' .AND. STATUS_C = 'B'

STATUS = 'PUBLIC & PRIVATE CAPTIVE & NON-CAPTIVE'

CASE

CKCOUNT = 11

SET FILTER TO STATUS_E = 'E' .AND. STATUS_C = 'N'

STATUS = 'PUBLIC & PRIVATE NON-CAPTIVE'

CASE

CKCOUNT = 12

SET FILTER TO STATUS_E = 'F'

STATUS = 'PRIVATE'

CASE

CKCOUNT = 13

SET FILTER TO STATUS_E = 'F' .AND. STATUS_C = 'C'

STATUS = 'PRIVATE CAPTIVE'

CASE

CKCOUNT = 14

SET FILTER TO STATUS_E = 'F' .AND. STATUS_C = 'B'

STATUS = 'PRIVATE CAPTIVE & NON-CAPTIVE'

CASE

CKCOUNT = 15


```

SET FILTER TO STATUS_E = 'P' .AND. STATUS_C = 'N'
STATUS = 'PRIVATE NON-CAPTIVE'
CASE CKCUNT = 16
SET FILTER TO STATUS_C = 'C'
STATUS = 'TOTAL CAPTIVE'
CASE CKCUNT = 17
SET FILTER TO STATUS_C = 'E'
STATUS = 'TOTAL CAPTIVE & NON-CAPTIVE'
CASE CKCUNT = 18
SET FILTER TO STATUS_C = 'N'
STATUS = 'TOTAL NON-CAPTIVE'
CASE CKCUNT = 19
IF CSFLAG = 0
  *** BUILD STFILE FOR PROPER STATE PROCESSING
  SELE 4
  GO TOP
  INDEX ON STATE+DIST TO CSTATE
  SET INDEX TO CSTATE
  GO TOP
  STOR SPACE(2) TO P1,P2
  *** REMOVE DUPLICATE RECORDS
  DO WHILE .NOT. EOF()
    IF STATE = P1 .AND. DIST = P2
      DELE
    ELSE
      P1 = STATE
      P2 = DIST
    ENDIF
  SKIP
  ENDDO
  PACK
  GO TOP
  JOIN WITH STATFILE TO STFILE FOR STATE=C->STATE .AND. DIST=C->DIST
  SELE 5
  USE STFILE
  INDEX ON S_DES+D_DES TO STFILE
  SET INDEX TO STFILE
  GO TOP
  CSFLAG = 1
  CS = ' '
  CD = ' '
ENDIF
  *** PROCESS INTER STATE DEPOSITS
  IF CFLAG = 1
    SET FILTER TO IS # '1'
    GO TOP
    DD = DEPOSIT
    DO IST
    DO WHILE DD # SPACE(4)
      LOCATE FOR DEPOSIT = DD
      IF .NOT. EOF()
        DO IST
      ELSE
        EXIT
      ENDIF
    ENDDO
    IF NFLAG = 1
      ? SPACE(40)+'----- NOT ESTIMATED -----'
      ? CLINE
      CLCTR = CLCTR + 2
      NFLAG = 0
    ENDIF
    CFLAG = 0
  ELSE

```



```

*** PROCESS INTER DISTRICT DEPOSITS
IF DFLAG = 1
  SET FILTER TO ID # '1'
  GO TCP
  DD = DEPOSIT
  DO IST
  DO WHILE DD # SPACE(4)
    LOCATE FOR DEPOSIT = DD
    IF .NOT. EOF()
      DO IST
    ELSE
      EXIT
    ENDIF
  ENDDO
  IF NFLAG = 1
    ? SPACE(40)+'----- NOT ESTIMATED -----'
    ? CLINE
    CLCTR = CLCTR + 2
    NFLAG = 0
  ENDIF
  DFLAG = 0
ENDIF
ENDIF
SELE 5
IF EOF()
  EXIT
ELSE
  *** SELECT STATE FOR PROCESSING
  IF STATE # CS
    CS = STATE
    CSTATE = TRIM(S_DES)
    CH1 = 'GRAND TOTAL OF '+CSTATE
    P1 = LEN(CH1)
    CL1 = REPLICATE('-',P1)
    CH11 = 'TOTAL OF '+CSTATE
    P1 = LEN(CH11)
    CL11 = REPLICATE('-',P1)
    SELE 1
    SET FILTER TO STATE = CS
    COPY TO STATE
    SELE 7
    USE STATE
    FN = '7'
    CS1 = 1
    CCD1 = 0
    IF CLCTR # 0
      ? SPACE(55)+CCOMPUTER
      CLCTR = 0
    ENDIF
  ELSE
    *** SELECT DISTRICT FOR PROCESSING
    CD = DIST
    CDIST = TRIM(D_DES)
    CH2 = 'GRAND TOTAL OF '+CDIST
    P1 = LEN(CH2)
    CL2 = REPLICATE('-',P1)
    CH22 = 'TOTAL OF '+CDIST
    P1 = LEN(CH22)
    CL22 = REPLICATE('-',P1)
    SKIP
    SELE 7
    SET FILTER TO DIST = CD
    COPY TO DISTRICT
    SELE 8
  
```



```

        USE DISTRICT
        FN = '8'
        CS1 = 0
        CCD1 = 1
        IF CLCTR > 46
            ? SPACE(55) + CCMPUTER
            CLCTR = 0
        ENDIF
        LCD = CLCTR
        IF CLCTR = 0
            LCD = 10
        ENDIF
    ENDIF
    CKCLNT = 0
    FLAG = 0
ENDIF
ENDCASE
SELE 8FN
GC TOP
IF ECF()
    CKCLNT = CKCUNT + 1
    LOCF
ENDIF
*** INITIALIZE VARIABLES FOR STORING VARIOUS TOTALS
I = 1
DO WHILE I < 21
    J = STR(I,2)
    IF I < 10
        J = STR(I,1)
    ENDIF
    STCR C TO IPRCV8J,IPRCE8J,IPCSS8J,RPRCV8J,RPRCE8J,RPCSS8J
    I = I + 1
ENDDO
STCR C TO IPRCV,IPRCB,IPOSS,RPROV,RPRCB,RFCSS,SMPRCV,SMPRCB,SMPOSS
STCR C TO FMPRCV,FMPRCB,PMFCSS,PRCG,PRCS,ITOTAL,RTOTAL,SMTOTAL
STCR C TO PMTCTAL,TOTAL
*** PROCESSING STARTS
DO WHILE .NOT. ECF()
    *** MAKE NOTE OF STATES AND DISTRICTS
    IF CKCLNT = 0 .AND. CSFLAG = 0
        IF CS # STATE .OR. CD # DIST
            CS = STATE
            CD = DIST
            SELE 4
            AFPE BLANK
            REPL STATE WITH CS,DIST WITH CD
            SELE 1
        ENDIF
    ENDIF
    *** MAKE A NOTE OF INTER STATE OR INTER DISTRICT DEPOSITS
    IF CS1 = 1 .AND. IS # '1'
        CFLAG = 1
        SKIP
        LOCF
    ELSE
        IF CCD1 = 1 .AND. ID # '1'
            CFLAG = 1
            SKIP
            LOCF
        ENDIF
    ENDIF
    *** ADD GRADE WISE RESERVES
    DO MIGRADD
    SKIP

```



```

ENDDO
*** DC FINAL TCTALLING
DC MIAED
*** GENERATE VARIABLES FOR PERCENTAGE COMPLTATIONS
IF CKCNT = 0 .AND. CSFLAG = 0
  TOTAL1 = IFROV
  TOTAL2 = IPROB
  TOTAL3 = IFCSS
  TOTAL7 = ITCTAL
  TOTAL4 = RPROV
  TOTAL5 = RPROB
  TOTAL6 = RFCSS
  TOTAL8 = RTCTAL
ENDIF
I = 1
DC WHILE I < 7
  II = STR(I,1)
  IF TOTAL8II = 0
    STORE 0.0000001 TO TOTAL8II
  ENDIF
  I = I + 1
ENDDO
*** PRINTING STARTS
*** PRINT TCP HEADING
IF CLCTR = 0
  IF TOTAL # 0 .OR. CKCNT < 4
    DC MINHEAD
  ENDIF
ENDIF
DC CASE
  *** PRINT MAIN HEADINGS
  CASE CKCNT = 0
    IF CS1 = 1
      ? CH1
      ? CL1
      CLCTR = CLCTR + 2
    ELSE
      IF CCD1 = 1
        IF CLCTR > 10
          ?
          ? CH2
          ? CL2
          CLCTR = CLCTR + 3
        ELSE
          ? CSTATE
          ? CH2
          ? CL2
          CLCTR = CLCTR + 3
        ENDIF
      ELSE
        ? 'ALL INDIA GRAND TCTAL '
        ? '-----'
        CLCTR = CLCTR + 2
      ENDIF
    ENDIF
  *** PRINT TOTAL RESOURCES
  IF ITCTAL # 0
    ? 'TOTAL RESOURCES (INSITU + CONDITIONAL)'
    ? SPACE(21),STR((IFROV+SMPROV+FMPCOV),15,3)+SPACE(1)
    ?? STR(IPROB+SMFROB+PMFROB,15,3) + SPACE(1)
    ?? STR(IPOSS+SMFCSS+PMFCSS,15,3) + SPACE(1)
    ?? STR(ITOTAL+SMTOTAL+FMTOTAL,19,3)
    ?? SPACE(2)+'TCTAL OF ALL GRADES'
    CLCTR = CLCTR + 2
  ENDIF

```



```

ENDIF
CASE CKCOUNT > 0
  *** PRINT SUB HEADINGS
  IF TOTAL # 0 .OR. CKCOUNT < 4
    IF FLAG = 0
      IF CS1 = 1
        IF CLCTR > 10
          ?
          ? CH11
          ? CL11
          CLCTR = CLCTR + 3
        ELSE
          ? CH11
          ? CL11
          CLCTR = CLCTR + 2
        ENDIF
      ELSE
        IF CCC1 = 1
          IF CLCTR > 10
            ?
            ? CH22
            ? CL22
            CLCTR = CLCTR + 3
          ELSE
            ? CSTATE
            ? CH22
            ? CL22
            CLCTR = CLCTR + 3
          ENDIF
        ELSE
          ? 'ALL INDIA TOTAL '
          ? '-----'
          CLCTR = CLCTR + 2
        ENDIF
      ENDIF
    ENDIF
    FLAG = 1
  ENDIF
  ? STATUS
  ?
  CLCTR = CLCTR + 2
ENDIF
ENDCASE
IF TOTAL = 0
  IF CKCOUNT = 0 .AND. (CFLAG = 1 .OR. DFLAG = 1)
    CKCOUNT = 19
    NFLAG = 1
    LOOP
  ENDIF
  IF CKCOUNT < 4
    ? SPACE(40) + '----- NOT ESTIMATED -----'
    ? CLINE
    CLCTR = CLCTR + 2
    IF CKCOUNT = 0
      CKCOUNT = 19
    ELSE
      CKCOUNT = CKCOUNT + 1
    ENDIF
  ELSE
    CKCOUNT = CKCOUNT + 1
  ENDIF
  LOOP
ENDIF
IF ITCTAL # 0
  ? 'INSITU RESERVES'

```



```

? SPACE(21),STR(IPRCV,15,3),STR(IPRCB,15,3),STR(IPCSS,15,3)+SPACE(1)
?? STR(ITOTAL,19,3),SPACE(1)+'TOTAL OF ALL GRADES'
? SPACE(28),'('+STR(IPRCV*100/TOTAL1,6,2)+'%)',SPACE(6)
?? '('+STR(IPRCB*100/TOTAL2,6,2)+'%)',SPACE(6)
?? '('+STR(IPOSS*100/TOTAL3,6,2)+'%)',SPACE(10)
?? '('+STR((ITOTAL)*100/TOTAL7,6,2)+'%)'
CLCTR = CLCTR+3
I = 1
DO WHILE I < CGCTR
  J = STR(I,2)
  IF I < 10
    J = STR(I,1)
  ENDIF
  JJ = CG8J
  IF VAL(JJ) < 10
    JJ = SUBSTR(JJ,2,1)
  ENDIF
  IF (IPROV8JJ + IPROB8JJ + IPCSS8JJ) # 0
    ? SPACE(21),STR(IPROV8JJ,15,3),STR(IPROB8JJ,15,3),STR(IPCSS8JJ,15,3)
    ??SPACE(1)+STR(IPROV8JJ+IPROB8JJ+IPCSS8JJ,19,3),SPACE(1)+CD&J
    ? SPACE(28),'('+STR(IPROV8JJ*100/TOTAL1,6,2)+'%)',SPACE(6)
    ?? '('+STR(IPROB8JJ*100/TOTAL2,6,2)+'%)',SPACE(6)
    ?? '('+STR(IPCSS8JJ*100/TOTAL3,6,2)+'%)',SPACE(10)
    ?? '('+STR((IPROV8JJ+IPROB8JJ+IPCSS8JJ)*100/TOTAL7,6,2)+'%)'
    CLCTR = CLCTR + 2
  ENDIF
  I = I + 1
ENDDO
ELSE
  IF CKOUNT = 0
    ? 'INSITU RESERVES',SPACE(24)+'----- NOT ESTIMATED -----'
    ?
    CLCTR = CLCTR + 2
  ENDIF
ENDIF
IF RTOTAL # 0
  ? 'RECOVERABLE RESERVES'
  ? SPACE(21),STR(RPRCV,15,3),STR(RPRCB,15,3),STR(RPCSS,15,3)+SPACE(1)
  ?? STR(RTOTAL,19,3),SPACE(1)+'TOTAL OF ALL GRADES'
  ? SPACE(28),'('+STR(RPRCV*100/TOTAL4,6,2)+'%)',SPACE(6)
  ?? '('+STR(RPRCB*100/TOTAL5,6,2)+'%)',SPACE(6)
  ?? '('+STR(RPOSS*100/TOTAL6,6,2)+'%)',SPACE(10)
  ?? '('+STR((RTOTAL)*100/TOTAL8,6,2)+'%)'
  CLCTR = CLCTR + 3
  I = 1
  DO WHILE I < CGCTR
    J = STR(I,2)
    IF I < 10
      J = STR(I,1)
    ENDIF
    JJ = CG8J
    IF VAL(JJ) < 10
      JJ = SUBSTR(JJ,2,1)
    ENDIF
    IF (RPROV8JJ + RPROB8JJ + RPCSS8JJ) # 0
      ? SPACE(21),STR(RPROV8JJ,15,3),STR(RPROB8JJ,15,3),STR(RPCSS8JJ,15,3)
      ??SPACE(1)+STR(RPROV8JJ+RPROB8JJ+RPCSS8JJ,19,3),SPACE(1)+CD&J
      ? SPACE(28),'('+STR(RPROV8JJ*100/TOTAL4,6,2)+'%)',SPACE(6)
      ?? '('+STR(RPROB8JJ*100/TOTAL5,6,2)+'%)',SPACE(6)
      ?? '('+STR(RPCSS8JJ*100/TOTAL6,6,2)+'%)',SPACE(10)
      ?? '('+STR((RPROV8JJ+RPROB8JJ+RPCSS8JJ)*100/TOTAL8,6,2)+'%)'
      CLCTR = CLCTR + 2
    ENDIF
    I = I + 1
  ENDWHILE

```



```

      ENDDC
    ENDF
  IF SMTCTAL # 0
    ? "CONDITIONAL (SUEMARGINAL)"
    ? SPACE(21),STR(SMPROV,15,3),STR(SMPROE,15,3),STR(SMPOSS,15,3)
    ?? SPACE(0),STR(SMTCTAL,19,3),SPACE(1)+"TOTAL OF ALL GRADES"
    CLCTR = CLCTR + 2
  ENDF
  IF PMTCTAL # 0
    ? "CONDITIONAL (PARAMARGINAL)"
    ? SPACE(21),STR(PMPROV,15,3),STR(PMPROE,15,3),STR(PMPOSS,15,3)
    ?? SPACE(0),STR(PMTCTAL,19,3),SPACE(1)+"TOTAL OF ALL GRADES"
    CLCTR = CLCTR + 2
  ENDF
  IF PRCS # 0
    ? "PROSPECTIVE RESOURCES",SPACE(47),STR(PRCS,19,3)
    ?? SPACE(2)+"TOTAL OF ALL GRADES"
    CLCTR = CLCTR + 1
  ENDF
  IF PRCG # 0
    ? "PROGNOSTIC RESOURCES",SPACE(47),STR(PRCG,19,3)
    CLCTR = CLCTR + 1
  ENDF
  ? CLINE
  CLCTR = CLCTR + 1
  *** RESERVE SPACE FOR PRINTING NEXT CASE
  IF CKCNT = 0
    IF (CSFLAG = 0 .OR. CS1 = 1)
      CK = CLCTR - 10
    ELSE
      IF CCD1 = 1
        CK = CLCTR - LCD
      ENDF
    ENDF
  ENDF
  IF (CLCTR + CK) > 85
    CLCTR = 0
    FLAG = 0
    ? SPACE(59)+COMPUTER
  ENDF
  CKCNT = CKCNT + 1
ENDDC
IF CLCTR # 0
  ? SPACE(59)+COMPUTER
ENDF
?
SET ALTERNATE OFF
SET PROC 10
CLEAR
6 11,20 SAY "REPORT OF "+CMDTY+" IS CONTAINED IN FILE "+CFILEN+".TXT"
6 13,20 SAY "IT CAN BE PRINTED USING WORDSTAR"
?
?

```



```
*****
** This program is a query subprogram which is used for selecting a
** commodity from the commodities in a particular district. And this
** program is used by all the programs for queries of district level.
*****
```

```
FLAG = 0
SELE 1
USE STRECCRD
SELE 3
USE MINFILE
SET INDEX TO MINFILE
SELE 1
SET INDEX TO STREC2
SEEK STCCDE
@ 2,66 SAY "DISTRICT : " + STCCDE
@ 5,0 CLEAR
@ 5,12 SAY "COMMODITY SELECTION"
@ 6,12 TO 22,65 DOUBLE
IF ECF()
    @ 12,21 SAY "THERE IS NO MINERAL / COMMODITY PRESENT"
    @ 14,33 SAY "IN THIS STATE"
    J = 1
    @ 23,75 SAY ""
    WAIT "
    RETURN                                PRESS ANY KEY TO CONTINUE"
ENDIF
@ 7,17 SAY "CCDE" + SPACE(2) + "COMMODITY NAME"
@ 8,17 SAY "----" + SPACE(2) + "-----"
I = 9
J = 1
DO WHILE STATE = STCCDE .AND. .NOT. ECF()
    MIN = "MI" + MINERAL
    SELE 2
    USE 8MIN
    SET INDEX TO 8MIN
    STATDIST = STCCDE + SECCDE
    SEEK STATDIST
    IF .NOT. ECF()
        SET RELATION TO MINERAL INTO MINFILE
        MDES = C->M_DES
        MCCDE = MINERAL
        IF I > 21
            @ 23,10 SAY SPACE(1)
            WAIT "
            @ 24,0 CLEAR
            IF UPPER(ANS) = 'E'
                EXIT
            ENDIF
            @ 9,15 CLEAR TO 21,64
            I = 9
        ENDIF
        @ 1,17 SAY MCCDE + SPACE(3) + MDES
        I = I+1
        J = J+1
    ENDIF
    SELE 1
    SKIP
ENDDO
IF J = 1
    @ 12,21 SAY "THERE IS NO MINERAL / COMMODITY PRESENT"
    @ 14,33 SAY "IN THIS DISTRICT"
```



```

& 23,75 SAY ""
WAIT "
RETURN
ENDIF
SELE 1
SET INDEX TO STREC1
SET FILTER TO STATE = STCCDE
GC TCP
DO WHILE FLAG = 0
ANS = SPACE(3)
& 24,C CLEAR
& 24,28 SAY "ENTER COMMODITY CODE"
& 24,49 GET ANS
READ
COMM = UPPR(ANS)
SELE 1
SEEK COMM
IF ECF()

```

PRESS ANY KEY TO CONTINUE"

```

    & 24,C CLEAR
    & 23,75 SAY ""
    WAIT "
    ENTERED WRONG CODE, SO PRESS ANY KEY TO ENTER THE CODE AGAIN"
ELSE
    EXIT
ENDIF
ENDDC
SELE 3
USE
SELE 2
USE MINFILE
SET INDEX TO MINFILE

```


PROGRAM NAME :- DISTQLRY.prg

```
*****  
** This program is a query subprogram which controls all the query  
** subprograms for the queries at district level in the state wise queries.  
*****
```

```
CC CASE  
  CASE ANS1 = "C"  
    CC STATQ5  
  CASE ANS1 = "R"  
    CC STATQ3  
  CASE ANS1 = "G"  
    CC STATQ6  
ENDCASE
```


PROGRAM NAME :- CTSLECT.prg

** This program is a query program for selecting a district after selecting
** a state. This is used in all the programs for district level query.

```
FLAG = C
@ 1,69 SAY "STATE : " + STCODE
@ 5,C CLEAR
SELE 2
SET INDEX TO STATFILE
GO TOP
SEEK STCODE
@ 5,7 SAY "DISTRICT SELECTION"
  @ 6,7 TO 22,72 DOUBLE
  @ 7,9 SAY "CODE" + SPACE(2) + "DISTRICT NAME"
  @ 8,9 SAY "----" + SPACE(2) + "-----"
  @ 7,41 SAY "CODE" + SPACE(2) + "DISTRICT NAME"
  @ 8,41 SAY "----" + SPACE(2) + "-----"
  I = 9
  J = 1
  DO WHILE STATE = STCODE .AND. .NOT. EOF()
    SDCES = D_CES
    SDCODE = DIST
    IF I > 21
      @ 23,10 SAY SPACE(1)
      WAIT "          PRESS 'E' TO ENTER CODE OR ANY OTHER KEY TO CONTINUE" TO ANS
      IF UPPER(ANS) = 'E'
        EXIT
      ENDIF
      @ 9,8 CLEAR TO 21,71
      I = 9
    ENDIF
    IF J/2 = INT(J/2)
      @ 1,42 SAY SDCODE + SPACE(3) + SDCES
      I = I+1
    ELSE
      @ 1,10 SAY SDCODE + SPACE(3) + SDCES
    ENDIF
    J = J+1
    SKIP
  ENDDO
DO WHILE FLAG = C
  ANS = SPACE(2)
  @ 24,C CLEAR
  @ 24,28 SAY "ENTER DISTRICT CODE"
  @ 24,49 GET ANS
  READ
  SDCODE = UPPER(ANS)
  SELE 2
  LOCATE FOR DIST = SDCODE
  IF EOF()
    @ 24,0 CLEAR
    @ 23,75 SAY ""
    WAIT "          ENTERED WRONG CODE, SO PRESS ANY KEY TO ENTER THE CODE AGAIN"
  ELSE
    EXIT
  ENDIF
ENDDO
```



```

*****
** This program is for getting the list of various types of errors.
** this is used in the validation program named as valid.
*****

```

```

USE ERRFILE
CLEAR
AA = SPACE(1)
G 10,20 SAY "DO YOU WANT THE LIST OF ERRORS ON PRINTER (Y/N)"
G 10,68 GET AA
READ
IF UPPER(AA) = 'Y'
  CLEAR
  G 10,25 SAY "SET PRINTER ON AND PRESS ANY KEY TO CONTINUE"
  SET CCNS OFF
  WAIT
  SET PRINT ON
  SET CCNS ON
ENDIF
? "S.NO.   CODE   ERROR DESCRIPTION"
? "-----"
I = 1
DO WHILE .NOT. ECF()
  ? SPACE(2)+STR(I,2)+SPACE(3)+ERRORNO+SPACE(3)+ERRDES
  I = I+1
  SKIP
ENDDO
USE
SET PRINT OFF

```



```

*****
** This program is for the selection of a particular grade of a particular
** commodity in the all types of queries.
*****

```

```

FLAG = 0
SELE 2
SET FILTER TO MINERAL = COMM
GC TOP
@ 5,C CLEAR
@ 5,1 SAY "GRADE SELECTION"
@ 6,C TO 19,79 DOUBLE
@ 7,1 SAY "CCDE" + SPACE(2) + "GRADE NAME"
@ 8,1 SAY "----" + SPACE(2) + "-----"
@ 7,41 SAY "CCDE" + SPACE(2) + "GRADE NAME"
@ 8,41 SAY "----" + SPACE(2) + "-----"
I = 9
J = 1
DO WHILE .NOT. EOF()
  IF J/2 # INT(J/2)
    @ 1,2 SAY G_CODE + SPACE(3) + G_DES
  ELSE
    @ 1,42 SAY G_CODE + SPACE(3) + G_DES
    I = I+1
  ENDF
  J = J+1
  SKIP
ENDDO
DO WHILE FLAG = 0
  ANS = SPACE(2)
  @ 24,31 SAY "ENTER GRADE CODE"
  @ 24,49 GET ANS
  READ
  GC TOP
  LOCATE FOR G_CODE = ANS
  IF .NOT. EOF()
    GDES = G_DES
    EXIT
  ENDF
  @ 23,75 SAY ""
  WAIT "      ENTERED WRONG CODE, PLEASE PRESS ANY KEY TO ENTER THE CODE AGAIN"
  @ 24,C CLEAR
ENDDO
SET FILTER TO
GCODE = ANS

```



```

*****
** This program is for the validation of the records of such deposits
** which spreeds over the state or district boundaries.This checks
** weather the no. of records for such types of deposits are appropriate.
*****

```

```

STOR C TO IPRCV,IPRCB,IPOSS,RPROV,RPRCB,RPCSS,CTOTAL
SELE 9
GC TOP
LOCA FOR MINERAL = SMCD .AND. DEPOSIT = DC
I = 11
DO WHILE I < 24
  J = STR(I,2)
  GRADE = IG8J
  IF GRADE # SPACE(2)
    II = 11
    DO WHILE II < SI
      JJ = STR(II,2)
      IF SG8JJ = GRADE
        IF SA8JJ # 'N'
          IPRCV = IPRCV +IPV8J
          IPRCB = IPRCB +IPB8J
          IPCSS = IPCSS +IPS8J
          EXIT
        ENDIF
      ENDIF
      II = II + 1
    ENDDO
  ELSE
    EXIT
  ENDIF
  I = I + 1
ENDDO
I = 11
DO WHILE I < 24
  J = STR(I,2)
  GRADE = RG8J
  IF GRADE # SPACE(2)
    II = 11
    DO WHILE II < SI
      JJ = STR(II,2)
      IF SG8JJ = GRADE
        IF SA8JJ # 'N'
          RPRCV = RPRCV +RPV8J
          RPRCB = RPRCB +RPB8J
          RPCSS = RPCSS +RPS8J
          EXIT
        ENDIF
      ENDIF
      II = II + 1
    ENDDO
  ELSE
    EXIT
  ENDIF
  I = I + 1
ENDDO
CTOTAL = IPROV+IPRCB+IPCSS+RPROV+RPROB+RPCSS+RMPV+RMPB+RMPs+SMPV+SMPB+SMPs+PRCS+PRCG
IF SLCTR + SLCTR1 > 77
  ? SPACE(58), "COMPUTERISED BY NATIONAL INFORMATICS CENTRE, PLANNING COMMISSION"
DO STATHEAD
ENDIF
IF CTOTAL # 0

```



```

SFLAG1 = 'N'
? '***'
SLCTR = SLCTR + 1
IF IPRCV+IPROB+IPOSS # 0
? 'INSITU RESERVES'
? SPACE(21),STR(IPRCV,15,3),STR(IPROB,15,3),STR(IPOSS,15,3)
?? SPACE(1)+STR(IPRCV+IPROB+IPOSS,19,3),SPACE(1)+'TOTAL OF ALL GRADES'
SLCTR = SLCTR + 2
II = 11
DO WHILE II < SI
JJ = STR(II,2)
GRADE = SG8JJ
I = 11
DO WHILE I < 24
J = STR(I,2)
IF IG8J = GRADE
? SPACE(21),STR(IPV8J,15,3),STR(IPB8J,15,3),STR(IPS8J,15,3)
?? SPACE(1)+STR(IPV8J+IPB8J+IPS8J,19,3),SPACE(1)+SC8JJ
SLCTR = SLCTR + 1
EXIT
ENDIF
I = I + 1
ENDDC
II = II + 1
ENDDC
ENDIF
IF RPRCV+RPROB+RPOSS # 0
? 'RECOVERABLE RESERVES'
? SPACE(21),STR(RPRCV,15,3),STR(RPROB,15,3),STR(RPOSS,15,3)
?? SPACE(1)+STR(RPRCV+RPROB+RPOSS,19,3),SPACE(1)+'TOTAL OF ALL GRADES'
SLCTR = SLCTR + 2
II = 11
DO WHILE II < SI
JJ = STR(II,2)
GRADE = SG8JJ
I = 11
DO WHILE I < 24
J = STR(I,2)
IF RG8J = GRADE
? SPACE(21),STR(RPV8J,15,3),STR(RPB8J,15,3),STR(RPS8J,15,3)
?? SPACE(1)+STR(RPV8J+RPB8J+RPS8J,19,3),SPACE(1)+SC8JJ
SLCTR = SLCTR + 1
EXIT
ENDIF
I = I + 1
ENDDC
II = II + 1
ENDDC
ENDIF
IF SMPV+SMPB+SMFS # 0
? 'CONDITIONAL (SUB MARGINAL)'
? SPACE(21),STR(SMPV,15,3),STR(SMPB,15,3),STR(SMFS,15,3)
?? SPACE(1)+STR(SMPV+SMPB+SMFS,19,3),SPACE(1)+'TOTAL OF ALL GRADES'
SLCTR = SLCTR + 2
ENDIF
IF FMPV+FMPB+FMFS # 0
? 'CONDITIONAL (PARAMARGINAL)'
? SPACE(21),STR(FMPV,15,3),STR(FMPB,15,3),STR(FMFS,15,3)
?? SPACE(1)+STR(FMPV+FMPB+FMFS,19,3),SPACE(1)+'TOTAL OF ALL GRADES'
CLCTR = CLCTR + 2
ENDIF
IF PROS # 0
? 'PROSPECTIVE RESOURCES',SPACE(47),STR(PROS,19,3)
?? SPACE(2)+'TOTAL OF ALL GRADES'

```



```

      SLCTR = SLCTR + 1
ENDIF
IF FROS # C
  ? 'PROGNOSTIC RESOURCES ',SPACE(47),STR(FROS,19,3)
  SLCTR = SLCTR + 1
ENDIF
?LINE
? '*** COMMON WITH : '
SLCTR = SLCTR + 2
I = 1
IF SK = 0
  DO WHILE I < 10
    J = STR(I,1)
    STAT = SUBSTR(SD8J,1,2)
    IF STAT = SPACE(2)
      EXIT
    ENDIF
    IF STAT # STCCDE
      SELE 2
      GC TCF
      LCCA FOR STATE = STAT
      ?? TRIM(S_DES)+SPACE(1)
      SELE 9
    ENDIF
    I = I+1
  ENDDO
ELSE
  STAT1 = STCCDE
  DO WHILE I < 10
    J = STR(I,1)
    STAT = SUBSTR(SD8J,1,2)
    DIS = SUBSTR(SD8J,3,2)
    IF STAT = SPACE(2)
      EXIT
    ENDIF
    IF DIS # SDCCDE
      SELE 2
      GC TCF
      LCCA FOR STATE = STAT .AND. DIST = DIS
      IF STAT = STAT1
        ?? TRIM(D_DES)+SPACE(1)
      ELSE
        STAT1 = STAT
        ? SPACE(18) + TRIM(S_DES)+' - '+TRIM(D_DES)+SPACE(1)
        SLCTR = SLCTR+1
      ENDIF
      SELE 9
    ENDIF
    I = I+1
  ENDDO
ENDIF
? LINE
SLCTR = SLCTR + 1
ENDIF
SKIP
IF .NOT. ECF() .AND. MINERAL = SMCE
  DD = DEPOSIT
ELSE
  DD = SPACE(4)
ENDIF

```



```

*****
** This program creates the pre requisite database named 'intrdist'
** which keeps the record of all the deposits which spreads over the
** state or district boundaries.
*****

```

```

SET DELE ON
SELE 1
USE CCATA
SELE 2
USE INTRFILE
ZAP
SELE 1
DO WHILE .NOT. EOF()
  MIN = MINERAL
  X = MIN
  MIN = "MI"+MIN
  SELE 3
  USE &MIN
  SELE 13
  SET FILTER TO ID # '1'
  GO TOP
  IF EOF()
    SELE 1
    SKIP
    LOOP
  ENDOF
  DO WHILE .NOT. EOF()
    STORE 1 TO S
    STORE 11 TO I,R
    B = BLCK
    X1 = VAL(DEPOSIT)
    XX1 = DEPOSIT
    X2 = STATE
    X3 = DIST
    X4 = IGRADE
    X5 = IFRCVED
    X6 = IFPROBABLE
    X7 = IFPOSSIBLE
    X8 = RGRADE
    X9 = RFRVCED
    X10 = RPROBABLE
    X11 = RPOSSIBLE
    X12 = SMPCVED
    X13 = SMPCREABLE
    X14 = SMPOSSIBLE
    X15 = FMPCVED
    X16 = FMPCREABLE
    X17 = FMPOSSIBLE
    X18 = FRCGTY
    X19 = FRCGTY
    SELE 2
    APPEND BLANK
    REPL MINERAL WITH X,DEPOSIT WITH XX1,SC1 WITH X2+X3
    S = S+1
    IF X4 # SPACE(2)
      REPL IG11 WITH X4,IPV11 WITH X5,IPB11 WITH X6,IFS11 WITH X7
      I = I+1
    ENDOF
    IF X8 # SPACE(2)
      REPL RG11 WITH X8,RV11 WITH X9,RPB11 WITH X10,RPS11 WITH X11
      R = R+1
    ENDOF
  ENDOF

```



```

ENDIF
SELE 3
SKIP
DO WHILE VAL(DEPOSIT) = X1 .AND. .NOT. ECF()
  IF STATE = X2 .AND. LIST = X3
    X12 = X12+SMFROVED
    X13 = X13+SMFRCBAELE
    X14 = X14+SMFCSSIELE
    X15 = X15+PMFROVED
    X16 = X16+PMFRCBAELE
    X17 = X17+PMFCSSIELE
    X18 = X18+PRCSQTY
    X19 = X19+PRCGQTY
    IF IGRADE # SPACE(2)
      X4 = IGRADE
      X5 = IPROVED
      X6 = IPROEABLE
      X7 = IPCSSIBLE
      IF BLOCK # B
        SELE 2
        FLG=C
        T = 11
        DO WHILE T < I
          IT=STR(T,2)
          IF X4 = IG&IT
            REPL IPV&IT WITH IPV&IT+X5
            REPL IPB&IT WITH IPB&IT+X6
            REPL IPS&IT WITH IPS&IT+X7
            FLG=1
            EXIT
          ENDIF
          T = T+1
        ENDDO
        IF FLG = 0
          II = STR(I,2)
          REPL IG&II WITH X4,IPV&II WITH X5,IPB&II WITH X6,IPS&II WITH X7
          I = I+1
        ENDIF
      ELSE
        SELE 2
        II = STR(I,2)
        REPL IG&II WITH X4,IPV&II WITH X5,IPB&II WITH X6,IPS&II WITH X7
        I = I+1
      ENDIF
    ENDIF
    SELE 3
  ENDIF
  IF RGRADE # SPACE(2)
    X8 = RGRADE
    X9 = RPROVED
    X10 = RPROEABLE
    X11 = RPOSSIBLE
    IF BLOCK # B
      SELE 2
      FLG=C
      T = 11
      DO WHILE T < R
        IT=STR(T,2)
        IF X8 = RG&IT
          REPL RPV&IT WITH RPV&IT+X9
          REPL RPB&IT WITH RPB&IT+X10
          REPL RPS&IT WITH RPS&IT+X11
          FLG=1
          EXIT
        ENDIF
      ENDDO
    ENDIF
  ENDIF

```



```

        T = T+1
        ENDDC
        IF FLG = 0
            RR = STR(R,2)
            REPL RG8RR WITH X8,RPV8RR WITH X9,RFB8RR WITH X10,RPS8RR WITH X11
            R = R+1
        ENDIF
    ELSE
        SELE 2
        RR = STR(R,2)
        REPL RG8RR WITH X8,RPV8RR WITH X9,RFB8RR WITH X10,RPS8RR WITH X11
        R = R+1
    ENDIF
    SELE 3
ENDIF
ELSE
    SS = STR(S,1)
    X2 = STATE
    X3 = DIST
    SELE 2
    REPL SD8SS WITH X2+X3
    S = S+1
    SELE 3
ENDIF
SKIP
ENDDC
SELE 2
REPL SMPV WITH X12,SMFB WITH X13,SMPS WITH X14,FMPV WITH X15
REPL FMPB WITH X16,FMPS WITH X17,PRCS WITH X18,PRCG WITH X19
SELE 3
ENDDC
SELE 1
SKIP
ENDDC
SELE 2
GC TOP
CC WHILE .NOT. ECF()
    MIN = MINERAL
    SKIP
    DO WHILE MINERAL = MIN
        SKIP - 1
        SD11 = SD1+SD2+SD3+SD4+SD5+SD6+SD7+SD8+SD9
        I = 11
        DO WHILE I < 24
            II = STR(I,2)
            X8II = IG8II
            Y8II = RG8II
            I = I+1
        ENDDC
        SKIP
        IF SD11 = (SD1+SD2+SD3+SD4+SD5+SD6+SD7+SD8+SD9)
            X27 = PMFV
            X28 = PMFB
            X29 = PMPS
            X30 = SMPV
            X31 = SMFB
            X32 = SMPS
            X33 = PRCS
            X34 = PRCG
            I = 11
            DO WHILE I < 24
                II = STR(I,2)
                FLAG = 'N'
                IF IG8II = SPACE(2)

```



```

EXIT
ENDIF
IGRADE = IG8II
X24 = IPV8II
X25 = IPE8II
X26 = IPS8II
J = 11
DO WHILE J < 24
  JJ = STR(J,2)
  IF X8JJ = SPACE(2)
    EXIT
  ENDIF
  IF IGRADE = X8JJ
    FLAG = 'Y'
  ENDIF
  J = J+1
ENDDO
SKIP - 1
IF FLAG = 'Y'
  REPL IPV8JJ WITH IPV8JJ+X24, IPE8JJ WITH IPE8JJ+X25, IPS8JJ WITH IPS8JJ+X26
ELSE
  REPL IG8JJ WITH IGRADE, IPV8JJ WITH X24, IPE8JJ WITH X25, IPS8JJ WITH X26
  X8JJ = IGRADE
ENDIF
SKIP
I = I+1
ENDCC
I = 11
DO WHILE I < 24
  II = STR(I,2)
  FLAG = 'N'
  IF RG8II = SPACE(2)
    EXIT
  ENDIF
  RGRADE = RG8II
  X24 = RPV8II
  X25 = RPE8II
  X26 = RPS8II
  J = 11
  DO WHILE J < 24
    JJ = STR(J,2)
    IF Y8JJ = SPACE(2)
      EXIT
    ENDIF
    IF RGRADE = Y8JJ
      FLAG = 'Y'
    ENDIF
    J = J+1
  ENDDO
  SKIP - 1
  IF FLAG = 'Y'
    REPL RPV8JJ WITH RPV8JJ+X24, RPE8JJ WITH RPE8JJ+X25, RPS8JJ WITH RPS8JJ+X26
  ELSE
    REPL RG8JJ WITH RGRADE, RPV8JJ WITH X24, RPE8JJ WITH X25, RPS8JJ WITH X26
    Y8JJ = RGRADE
  ENDIF
  SKIP
  I = I+1
ENDCC
SKIP - 1
REPL PMPV WITH PMPV+X27, PMPE WITH PMPE+X28, PMPS WITH PMPS+X29
REPL SMPV WITH SMPV+X30, SMPE WITH SMPE+X31, SMPS WITH SMPS+X32

```



```
REPL PRCS WITH FROS+X33,PROG WITH PRCG+X34  
SKIP  
DELE  
ELSE  
SKIP  
ENDIF  
ENDDC  
ENDDC  
SET DELE OFF  
PACK
```



```

*****
** This program is a procedure file which contains different small
** procedure used by the main program named as 'rmi.prg'.
*****

* THIS PROCEDURE ACCEPTS THE PASSWORD AND STARTS THE PROCESSINGS
PROCEDURE PASSWORD
PASS = SPACE(3)
X = 1
DO WHILE X < 4
  CLEAR
  @ 10,21 SAY "To start ENTER the PASSWORD"
  @ 10,52 GET PASS
  CLEAR GETS
  STORE C TO K1,K2,K3
  I = 1
  DO WHILE I < 2
    INK1 = INKEY()
    IF INK1 = 78 .OR. INK1 = 110
      K1 = 1
    ENDIF
    IF INK1 # C
      I = I+1
    ENDIF
  ENDDO
  DO WHILE I < 3
    INK2 = INKEY()
    IF INK2 = 77 .OR. INK2 = 109
      K2 = 1
    ENDIF
    IF INK2 # C
      I = I+1
    ENDIF
  ENDDO
  DO WHILE I < 4
    INK3 = INKEY()
    IF INK3 = 73 .OR. INK3 = 105
      K3 = 1
    ENDIF
    IF INK3 # C
      I = I+1
    ENDIF
  ENDDO
  ? CHR(7)
  IF (K1+K2+K3) = 3
    PW = 'Y'
    EXIT
  ELSE
    SET COLOR TO W*
    @ 13,28 SAY "WRONG PASSWORD"
    SET COLOR TO
    IF X < 3
      @ 16,21 SAY "Press ANY KEY to Try Again"
    ELSE
      @ 16,25 SAY "Sorry ! No more Try"
    ENDIF
    SET CONS OFF
    WAIT
    SET CONS ON
    X = X+1
  ENDIF
ENDIF

```



```

ENDDC
RETURN
*****
* THIS PROCEDURE CREATES THE SCREEN WRITTEN NIC
PROCEDURE NIC
X = REPLICATE(CHR(205),76)
Y = REPLICATE(CHR(10),74)
Z = REPLICATE(CHR(223),70)
V = REPLICATE(CHR(220),70)
W = REPLICATE(CHR(176),68)
L = REPLICATE(CHR(249),3)
M = REPLICATE(CHR(249),6)
N = REPLICATE(CHR(249),2)
T = REPLICATE(CHR(249),9)
S = SPACE(64)
F = CHR(10)
CLEAR
@ 3,2 SAY CHR(201)+X+CHR(187)
@ 4,4 GET Y
@ 5,6 SAY V
@ 6,7 SAY W
I = 4
DO WHILE I < 23
  @ 1,2 SAY CHR(186)
  @ 1,79 SAY CHR(186)
  IF I > 4 .AND. I < 21
    @ 1,4 GET R
    @ 1,77 GET R
  ENDF
  IF I > 5 .AND. I < 21
    @ 1,6 SAY CHR(219)
    @ 1,75 SAY CHR(219)
    @ 1,7 SAY CHR(176)+CHR(176)
    @ 1,73 SAY CHR(176)+CHR(176)
    @ 1+1,9 GET S
  ENDF
  I = I+1
ENDDC
@ 19,7 SAY W
@ 20,6 SAY Z
@ 21,4 GET Y
@ 22,2 SAY CHR(200)+X+CHR(188)
I = 9
J = 22
SET CCLOR TO W*
DO WHILE I < 17
  @ 1,22 SAY N
  @ 1,1,J SAY N
  @ 1,30 SAY N
  @ 1,41 SAY N
  @ 1,52 SAY N
  IF I = 10 .OR. I = 15
    @ 1,59 SAY N
  ENDF
  IF I = 9 .OR. I = 16
    @ 1,39 SAY M
    @ 1,52 SAY T
  ENDF
  I = I+1
  J = J+1
ENDDC
SET CCLOR TO
CLEAR GETS
RETURN

```

* THIS PROCEDURE PRESENTS THE SCREEN WRITTEN 'PRESENTS DATABASE FOR
* DEPARTMENT OF MINES '
PROCEDURE PRESENTS
CLEAR

```
6 6,20 TO 19,60 DOUBLE
DO LEFT WITH 'R',8,37
DO TOP WITH 'E',8,38
DO RIGHT WITH 'S',8,39
DO BOTTOM WITH 'E',8,40
DO LEFT WITH 'P',8,36
DO TOP WITH 'N',8,41
DO RIGHT WITH 'T',8,42
DO BOTTOM WITH 'S',8,43
DO LEFT WITH 'E',11,35
DO BOTTOM WITH 'R',11,36
DO RIGHT WITH 'A',11,37
DO BOTTOM WITH 'L',11,38
DO LEFT WITH 'N',11,34
DO BOTTOM WITH 'C',11,41
DO RIGHT WITH 'A',11,42
DO BOTTOM WITH 'T',11,43
DO LEFT WITH 'I',11,33
DO BOTTOM WITH 'A',11,44
DO RIGHT WITH 'B',11,45
DO BOTTOM WITH 'A',11,46
DO LEFT WITH 'M',11,32
DO BOTTOM WITH 'S',11,47
DO RIGHT WITH 'E',11,48
DO LEFT WITH 'F',14,39
DO BOTTOM WITH 'C',14,40
DO RIGHT WITH 'R',14,41
DO LEFT WITH 'R',17,34
DO BOTTOM WITH 'T',17,35
DO RIGHT WITH 'M',17,36
DO BOTTOM WITH 'E',17,37
DO LEFT WITH 'A',17,33
DO BOTTOM WITH 'N',17,38
DO RIGHT WITH 'T',17,39
DO BOTTOM WITH 'C',17,42
DO LEFT WITH 'P',17,32
DO BOTTOM WITH 'F',17,43
DO RIGHT WITH 'M',17,46
DO BOTTOM WITH 'I',17,47
DO LEFT WITH 'E',17,31
DO BOTTOM WITH 'N',17,48
DO RIGHT WITH 'E',17,49
DO BOTTOM WITH 'S',17,50
DO LEFT WITH 'D',17,30
RETURN
```

* THIS PROCEDURE MOVES THE CHARACTERS FROM UPWARD TO DOWNWARD DIRECTION
PROCEDURE TOP

PARAMETERS X,R,S

Z = LEN(X)

I = 1

DO WHILE I <= Z

SUB = SUBSTR(X,I,1)

K = 1

DO WHILE K <= R

6 K,S SAY SUB

IF K < R

6 K,S SAY SPACE(1)

ENDIF


```

      K = K+1
      IF K = 6
        K = K+1
      ENDIF
    ENDDC
    I = I+1
    S = S+1
  ENDDC
*****

```

```

* THIS PROCEDURE MOVES THE CHARACTERS FROM DOWNWARD TO UPWARD DIRECTION
PROCEDURE BOTTOM

```

```

PARAMETERS X,R,S
Z = LEN(X)
I = 1
DO WHILE I <= Z
  SUB = SUBSTR(X,I,1)
  K = 24
  DO WHILE K >= R
    @ K,S SAY SUB
    IF K > R
      @ K,S SAY SPACE(1)
    ENDIF
    K = K-1
    IF K = 19
      K = K-1
    ENDIF
  ENDDO
  I = I+1
  S = S+1
ENDDC
*****

```

```

* THIS PROCEDURE MOVES THE CHARACTERS FROM LEFT TO RIGHT DIRECTION
PROCEDURE LEFT

```

```

PARAMETERS X,R,S
Z = LEN(X)
I = Z
DO WHILE I >= 1
  SUB = SUBSTR(X,I,1)
  K = 1
  DO WHILE K <= S
    @ R,K SAY SUB
    IF K < S
      @ R,K SAY SPACE(1)
    ENDIF
    K = K+1
    IF K = 20
      K = K+1
    ENDIF
  ENDDO
  I = I-1
  S = S-1
ENDDC
*****

```

```

* THIS PROCEDURE MOVES THE CHARACTERS FROM RIGHT TO LEFT DIRECTION
PROCEDURE RIGHT

```

```

PARAMETERS X,R,S
Z = LEN(X)
I = 1
DO WHILE I <= Z
  SUB = SUBSTR(X,I,1)
  K = 79

```



```

DO WHILE K >= S
  & R,K SAY SLB
  IF K > S
    & R,K SAY SPACE(1)
  ENDF
  K = K-1
  IF K = 60
    K = K-1
  ENDF
ENDDC
I = I+1
S = S+1
ENDDC
*****
* THIS PROCEDURE CREATES THE MAIN MENU
PROCEDURE MENU1
STORE C TO NMI
CODE11 = "1. HELP "
CODE13 = "2. QUERIES "
CODE15 = "3. REPORTS "
CODE17 = "4. PRE-REQUISITIVE "
CODE19 = "5. VALIDATION "
CODE21 = "6. EXIT "
L = CHR(24)
M = CHR(25)
N = CHR(17)+CHR(196)+CHR(217)
CLEAR
& 2,33 SAY "MINERAL DATABASE"
& 3,28 SAY "NATIONAL MINERAL INVENTORY"
& 4,31 SAY "DEPARTMENT OF MINES"
& 8,32 SAY CODE11
& 10,32 SAY CODE13
& 12,32 SAY CODE15
& 14,32 SAY CODE17
& 16,32 SAY CODE19
& 18,32 SAY CODE21
& 6,25 TO 20,55 DCUBLE
& 22,11 SAY "Use "+L+" "+M+" for Curser movement and press "+N+" for selection"
INK = 0
I = 11
DO WHILE I < 22
  II = STR(I,2)
  & I-3,32 GET CODE8II
  INK = INKEY()
  DO CASE
    CASE INK = 5
      & I-3,32 SAY CODE8II
      IF I > 11
        I = I-2
      ELSE
        I = 21
      ENDF
    CASE INK = 24
      & I-3,32 SAY CODE8II
      IF I < 21
        I = I+2
      ELSE
        I = 11
      ENDF
    CASE INK = 13
      EXIT
  ENDCASE
  CLEAR GETS
ENDDC

```



```

CC CASE
CASE I = 11
  CC MIHELP
CASE I = 13
  CC WHILE NMI # 1
    CC QMENU1
  ENDDC
CASE I = 15
  CC WHILE NMI # 2
    CC MENU2
  ENDDC
CASE I = 17
  CC WHILE NMI # 3
    CC MENU3
  ENDDC
CASE I = 19
  CC WHILE NMI # 4
    CC MENU4
  ENDDC
CASE I = 21
  NMI = 6
ENDCASE

```

```

*****

```

```

* THIS PROCEDURE CREATES THE MENU FOR GETTING REPORTS

```

```

PROCEDURE MENU2

```

```

CODE11 = "1. HELP "

```

```

CODE13 = "2. COMMODITY-WISE SUMMARY STATEMENT "

```

```

CODE15 = "3. STATE-WISE SUMMARY STATEMENT "

```

```

CODE17 = "4. EXIT "

```

```

L = CHR(24)

```

```

M = CHR(25)

```

```

N = CHR(17)+CHR(196)+CHR(217)

```

```

CLEAR

```

```

@ 2,33 SAY "MINERAL DATABASE"

```

```

@ 3,28 SAY "NATIONAL MINERAL INVENTORY"

```

```

@ 4,31 SAY "DEPARTMENT OF MINES"

```

```

@ 8,23 SAY CODE11

```

```

@ 10,23 SAY CODE13

```

```

@ 12,23 SAY CODE15

```

```

@ 14,23 SAY CODE17

```

```

@ 6,16 TO 16,64 DCUBLE

```

```

@ 22,11 SAY "Use "+L+" "+M+" for Curser movement and press "+N+" for selection"

```

```

INK = 0

```

```

I = 11

```

```

CC WHILE I < 18

```

```

  II = STR(I,2)

```

```

  @ I-3,23 GET CODE8II

```

```

  INK = INKEY()

```

```

  CC CASE

```

```

    CASE INK = 5

```

```

      @ I-3,23 SAY CODE8II

```

```

      IF I > 11

```

```

        I = I-2

```

```

      ELSE

```

```

        I = 17

```

```

      ENDIF

```

```

    CASE INK = 24

```

```

      @ I-3,23 SAY CODE8II

```

```

      IF I < 17

```

```

        I = I+2

```

```

      ELSE

```

```

        I = 11

```

```

      ENDIF

```

```

    CASE INK = 13

```



```

EXIT
ENDCASE
CLEAR GETS
ENDDC
DO CASE
    CASE I = 11
        DO MIHELP2
        WAIT
        DO MIHELP21
        WAIT
    CASE I = 13
        DO CMDTY
    CASE I = 15
        DO STATWISE
    CASE I = 17
        NMI = 2
ENDCASE
*****
* THIS PROCEDURE CREATES THE MENU FOR PRE REQUISITIVE PURPOSES
PROCEDURE MENU3
CCDE11 = "1. HELP "
CCDE13 = "2. CREATE CDATA.DBF FILE "
CCDE15 = "3. CREATE SCATA.DBF FILE "
CCDE17 = "4. CREATE STDFILE.DBF FILE "
CCDE19 = "5. CREATE INTRFILE.DBF FILE "
CCDE21 = "6. EXIT "
L = CHR(24)
M = CHR(25)
N = CHR(17)+CHR(196)+CHR(217)
CLEAR
@ 2,33 SAY "MINERAL DATABASE"
@ 3,28 SAY "NATIONAL MINERAL INVENTORY"
@ 4,31 SAY "DEPARTMENT OF MINES"
@ 8,27 SAY CCDE11
@ 10,27 SAY CCDE13
@ 12,27 SAY CCDE15
@ 14,27 SAY CCDE17
@ 16,27 SAY CCDE19
@ 18,27 SAY CCDE21
@ 6,20 TO 20,60 DOUBLE
@ 22,11 SAY "Use "+L+" "+M+" for Curser movement and press '+N+' for selection"
INK = 0
I = 11
DO WHILE I < 22
    II = STR(I,2)
    @ 1-3,27 GET CCDE8II
    INK = INKEY()
    DO CASE
        CASE INK = 5
            @ 1-3,27 SAY CCDE8II
            IF I > 11
                I = I-2
            ELSE
                I = 21
            ENDIF
        CASE INK = 24
            @ 1-3,27 SAY CCDE8II
            IF I < 21
                I = I+2
            ELSE
                I = 11
            ENDIF
        CASE INK = 13
            EXIT

```



```

ENDCASE
CLEAR GETS
ENDDC
DO CASE
CASE 1 = 11
    DO MIHELP3
    WAIT
CASE 1 = 13
    CLEAR
    @ 10,10 SAY "CREATING CDATA.DBF FILE, SO PLEASE WAIT FOR A MINUTE"
    DO CDATA
CASE 1 = 15
    CLEAR
    @ 10,10 SAY "CREATING SDATA.DBF FILE, SO PLEASE WAIT FOR A MINUTE"
    DO SDATA
CASE 1 = 17
    CLEAR
    @ 10,10 SAY "CREATING STDFILE.DBF FILE, SO PLEASE WAIT FOR 3-4 MINUTES"
    DO STRECORD
CASE 1 = 19
    CLEAR
    @ 10,10 SAY "CREATING INTRFILE.DBF FILE, SO PLEASE WAIT FOR 7-8 MINUTES"
    DO ISID
CASE 1 = 21
    NMI = 3

```

```
ENDCASE
```

```
*****
```

```
* THIS PROCEDURE CREATES THE MENU FOR VALIDATION PART
```

```
PROCEDURE MENL4
```

```
CCDE11 = "1. HELP "
```

```
CCDE13 = "2. VALIDATION OF A FILE "
```

```
CCDE15 = "3. LIST OF ERROR CODES "
```

```
CCDE17 = "4. EXIT "
```

```
L = CHR(24)
```

```
M = CHR(25)
```

```
N = CHR(17)+CHR(196)+CHR(217)
```

```
CLEAR
```

```
@ 2,33 SAY "MINERAL DATABASE"
```

```
@ 3,28 SAY "NATIONAL MINERAL INVENTORY"
```

```
@ 4,31 SAY "DEPARTMENT OF MINES"
```

```
@ 8,29 SAY CCDE11
```

```
@ 10,29 SAY CCDE13
```

```
@ 12,29 SAY CCDE15
```

```
@ 14,29 SAY CCDE17
```

```
@ 6,22 TO 16,58 DOUBLE
```

```
@ 22,11 SAY "Use "+L+" "+M+" for Curser movement and press '+N+' for selection"
```

```
INK = 0
```

```
I = 11
```

```
DO WHILE I < 18
```

```
    II = STR(I,2)
```

```
    @ 1-3,29 GET CCDE8II
```

```
    INK = INKEY()
```

```
    DO CASE
```

```
        CASE INK = 5
```

```
            @ 1-3,29 SAY CCDE8II
```

```
            IF I > 11
```

```
                I = I-2
```

```
            ELSE
```

```
                I = 17
```

```
            ENDOF
```

```
        CASE INK = 24
```

```
            @ 1-3,29 SAY CCDE8II
```

```
            IF I < 17
```

```
                I = I+2
```



```
        ELSE  
            I = 11  
        ENDIF  
    CASE INK = 13  
        EXIT  
    ENDCASE  
    CLEAR GETS  
ENDDC  
CC CASE  
    CASE I = 11  
        CC MIHELP4  
        WAIT  
    CASE I = 13  
        CC LL  
    CASE I = 15  
        CC ERRLIST  
    CASE I = 17  
        NMI = 4  
ENDCASE
```



```
*****
** This program is a procedure file which contains the various small
** procedure which are used in commodity wise cureies and for
** generating the commodity wise report.
*****
```

```
*****
PROCEDURE MINCHEAD
*****
```

```
CPCTR = CPCTR + 1
IF CPCTR = 1
  ? '.PL 96'
  ? '.MT 0'
  ? '.ME 0'
  ? '.OF'
ELSE
  ? '.PA'
ENDIF
? SPACE(54)+'MINERAL DATABASE'+SPACE(32)+'PAGE :',CMCTY+' - '+STR(CPCTR,4)
? SPACE(49)+'NATIONAL MINERAL INVENTORY'
? SPACE(42)+'MINERAL COMMODITY WISE SUMMARY STATEMENT'
? SPACE(42)+'-----'
? 'NAME OF MINERAL :',CMIN+SPACE(10)+'STATUS DATE :',CS_DATE
?? SPACE(9),'UNIT : '+CUNITDES
? CLINE
? SPACE(45)+'RESERVES AND RESOURCES'
? 'STATE/ DISTRICT'+SPACE(6),REPLICATE('-',67),' GRADE'
? SPACE(31)+'PROVED'+SPACE(8)+'PROBABLE'+SPACE(8)+'POSSIBLE'
?? SPACE(15)+'TOTAL'
? CLINE
CLCTR = 10
```

```
*****
PROCEDURE MIGRADD
*****
```

```
IF IGRADE # SPACE(2)
  DO CASE
    CASE IGRADE = 'C1'
      IPROV1 = IPROV1 + IPROVED
      IPRCB1 = IPRCB1 + IPRCEABLE
      IPCSS1 = IPCSS1 + IPOSSIBLE
    CASE IGRADE = 'C2'
      IPROV2 = IPROV2 + IPROVED
      IPRCB2 = IPRCB2 + IPRCEABLE
      IPCSS2 = IPCSS2 + IPOSSIBLE
    CASE IGRADE = 'C3'
      IPROV3 = IPROV3 + IPROVED
      IPRCB3 = IPRCB3 + IPRCEABLE
      IPCSS3 = IPCSS3 + IPOSSIBLE
    CASE IGRADE = 'C4'
      IPROV4 = IPROV4 + IPROVED
      IPRCB4 = IPRCB4 + IPRCEABLE
      IPCSS4 = IPCSS4 + IPOSSIBLE
    CASE IGRADE = 'C5'
      IPROV5 = IPROV5 + IPROVED
      IPRCB5 = IPRCB5 + IPRCEABLE
      IPCSS5 = IPCSS5 + IPOSSIBLE
    CASE IGRADE = 'C6'
      IPROV6 = IPROV6 + IPROVED
      IPRCB6 = IPRCB6 + IPRCEABLE
```



```

CASE IFCSS6 = IPOSS6 + IFCSSIBLE
CASE IGRADE = 'C7'
IFRCV7 = IPROV7 + IPRCVED
IFRCB7 = IPRCE7 + IPRCEABLE
IFCSS7 = IPOSS7 + IPOSSIBLE
CASE IGRADE = 'C8'
IFRCV8 = IPROV8 + IPRCVED
IFRCB8 = IPRCE8 + IPRCEABLE
IFCSS8 = IPOSS8 + IPOSSIBLE
CASE IGRADE = 'C9'
IFRCV9 = IPROV9 + IPRCVED
IFRCB9 = IPRCE9 + IPRCEABLE
IFCSS9 = IPOSS9 + IPOSSIBLE
CASE IGRADE = '10'
IFRCV10 = IPRCV10 + IPRCVED
IFRCB10 = IPRCE10 + IPRCEABLE
IFCSS10 = IPOSS10 + IFCSSIBLE
CASE IGRADE = '11'
IFRCV11 = IPRCV11 + IPRCVED
IFRCB11 = IPRCE11 + IPRCEABLE
IFCSS11 = IPOSS11 + IFCSSIBLE
CASE IGRADE = '12'
IFRCV12 = IPRCV12 + IPRCVED
IFRCB12 = IPRCE12 + IPRCEABLE
IFCSS12 = IPOSS12 + IFCSSIBLE
CASE IGRADE = '13'
IFRCV13 = IPRCV13 + IPRCVED
IFRCB13 = IPRCE13 + IPRCEABLE
IFCSS13 = IPOSS13 + IFCSSIBLE
CASE IGRADE = '14'
IFRCV14 = IPRCV14 + IPRCVED
IFRCB14 = IPRCE14 + IPRCEABLE
IFCSS14 = IPOSS14 + IFCSSIBLE
CASE IGRADE = '15'
IFRCV15 = IPRCV15 + IPRCVED
IFRCB15 = IPRCE15 + IPRCEABLE
IFCSS15 = IPOSS15 + IFCSSIBLE
CASE IGRADE = '16'
IFRCV16 = IPRCV16 + IPRCVED
IFRCB16 = IPRCE16 + IPRCEABLE
IFCSS16 = IPOSS16 + IFCSSIBLE
CASE IGRADE = '17'
IFRCV17 = IPRCV17 + IPRCVED
IFRCB17 = IPRCE17 + IPRCEABLE
IFCSS17 = IPOSS17 + IFCSSIBLE
CASE IGRADE = '18'
IFRCV18 = IPRCV18 + IPRCVED
IFRCB18 = IPRCE18 + IPRCEABLE
IFCSS18 = IPOSS18 + IFCSSIBLE
CASE IGRADE = '19'
IFRCV19 = IPRCV19 + IPRCVED
IFRCB19 = IPRCE19 + IPRCEABLE
IFCSS19 = IPOSS19 + IFCSSIBLE
CASE IGRADE = '20'
IFRCV20 = IPRCV20 + IPRCVED
IFRCB20 = IPRCE20 + IPRCEABLE
IFCSS20 = IPOSS20 + IFCSSIBLE

```

```

ENDCASE
ENDIF
IF RGRADE # SPACE(2)
DC CASE
CASE RGRADE = 'C1'
RRCV1 = RPROV1 + RPRCVED
RRCB1 = RPRCE1 + RPRCEABLE

```


CASE RFGSS1 = RFGSS1 + RFOSSIBLE
 RFGRADE = 'C2'
 RFGROV2 = RFGROV2 + RFGROVED
 RFGROB2 = RFGROB2 + RFGROBABLE
 RFGSS2 = RFGSS2 + RFOSSIBLE
 CASE RFGRADE = 'C3'
 RFGROV3 = RFGROV3 + RFGROVED
 RFGROB3 = RFGROB3 + RFGROBABLE
 RFGSS3 = RFGSS3 + RFOSSIBLE
 CASE RFGRADE = 'C4'
 RFGROV4 = RFGROV4 + RFGROVED
 RFGROB4 = RFGROB4 + RFGROBABLE
 RFGSS4 = RFGSS4 + RFOSSIBLE
 CASE RFGRADE = 'C5'
 RFGROV5 = RFGROV5 + RFGROVED
 RFGROB5 = RFGROB5 + RFGROBABLE
 RFGSS5 = RFGSS5 + RFOSSIBLE
 CASE RFGRADE = 'C6'
 RFGROV6 = RFGROV6 + RFGROVED
 RFGROB6 = RFGROB6 + RFGROBABLE
 RFGSS6 = RFGSS6 + RFOSSIBLE
 CASE RFGRADE = 'C7'
 RFGROV7 = RFGROV7 + RFGROVED
 RFGROB7 = RFGROB7 + RFGROBABLE
 RFGSS7 = RFGSS7 + RFOSSIBLE
 CASE RFGRADE = 'C8'
 RFGROV8 = RFGROV8 + RFGROVED
 RFGROB8 = RFGROB8 + RFGROBABLE
 RFGSS8 = RFGSS8 + RFOSSIBLE
 CASE RFGRADE = 'C9'
 RFGROV9 = RFGROV9 + RFGROVED
 RFGROB9 = RFGROB9 + RFGROBABLE
 RFGSS9 = RFGSS9 + RFOSSIBLE
 CASE RFGRADE = '10'
 RFGROV10 = RFGROV10 + RFGROVED
 RFGROB10 = RFGROB10 + RFGROBABLE
 RFGSS10 = RFGSS10 + RFGSSIBLE
 CASE RFGRADE = '11'
 RFGROV11 = RFGROV11 + RFGROVED
 RFGROB11 = RFGROB11 + RFGROBABLE
 RFGSS11 = RFGSS11 + RFGSSIBLE
 CASE RFGRADE = '12'
 RFGROV12 = RFGROV12 + RFGROVED
 RFGROB12 = RFGROB12 + RFGROBABLE
 RFGSS12 = RFGSS12 + RFGSSIBLE
 CASE RFGRADE = '13'
 RFGROV13 = RFGROV13 + RFGROVED
 RFGROB13 = RFGROB13 + RFGROBABLE
 RFGSS13 = RFGSS13 + RFGSSIBLE
 CASE RFGRADE = '14'
 RFGROV14 = RFGROV14 + RFGROVED
 RFGROB14 = RFGROB14 + RFGROBABLE
 RFGSS14 = RFGSS14 + RFGSSIBLE
 CASE RFGRADE = '15'
 RFGROV15 = RFGROV15 + RFGROVED
 RFGROB15 = RFGROB15 + RFGROBABLE
 RFGSS15 = RFGSS15 + RFGSSIBLE
 CASE RFGRADE = '16'
 RFGROV16 = RFGROV16 + RFGROVED
 RFGROB16 = RFGROB16 + RFGROBABLE
 RFGSS16 = RFGSS16 + RFGSSIBLE
 CASE RFGRADE = '17'
 RFGROV17 = RFGROV17 + RFGROVED
 RFGROB17 = RFGROB17 + RFGROBABLE


```

        RFCSS17 = RPOSS17 + RFCSSIELE
CASE RGRADE = '18'
        RFRVC18 = RPRCV18 + RPROVED
        RFRCB18 = RPRCB18 + RPROBABLE
        RFCSS18 = RPOSS18 + RFCSSIELE
CASE RGRADE = '19'
        RFRVC19 = RPRCV19 + RPROVED
        RFRCB19 = RPRCB19 + RPROBABLE
        RFCSS19 = RPOSS19 + RFCSSIELE
CASE RGRADE = '20'
        RFRVC20 = RPRCV20 + RPROVED
        RFRCB20 = RPRCB20 + RPROBABLE
        RFCSS20 = RPOSS20 + RFCSSIELE

```

ENDCASE

ENDIF

```

SMPRCV = SMPRCV + SMFROVED
SMPRCE = SMPRCE + SMFROBABLE
SMPOSS = SMPCSS + SMFCSSIELE
FMPRCV = FMPRCV + PMFROVED
FMPRCE = FMPRCE + PMFROBABLE
FMPOSS = FMPCSS + PMFCSSIELE
FROG = PRCG + PRCGQTY
FROS = PRCS + PRCSQTY

```

```

*****
PROCEDURE MIALD
*****

```

```

IPROV = IPROV1+IPROV2+IPROV3+IPROV4+IPROV5+IPROV6+IPROV7+IPROV8
IPROV = IPROV+IPROV9+IPROV10+IPROV11+IPROV12+IPROV13+IPROV14
IPROV = IPROV+IPROV15+IPROV16+IPROV17+IPROV18+IPROV19+IPROV20
IPROB = IPRCE1+IPROB2+IPROB3+IPROB4+IPROB5+IPROB6+IPROB7+IPROB8
IPROB = IPRCE+IPROB9+IPROB10+IPROB11+IPROB12+IPROB13+IPROB14
IPROB = IPRCE+IPROB15+IPROB16+IPROB17+IPROB18+IPROB19+IPROB20
IPCSS = IPCSS1+IPCSS2+IPCSS3+IPCSS4+IPCSS5+IPCSS6+IPCSS7+IPCSS8
IPCSS = IPCSS+IPCSS9+IPCSS10+IPCSS11+IPCSS12+IPCSS13+IPCSS14
IPCSS = IPCSS+IPCSS15+IPCSS16+IPCSS17+IPCSS18+IPCSS19+IPCSS20
RPROV = RPROV1+RPROV2+RPROV3+RPROV4+RPROV5+RPROV6+RPROV7+RPROV8
RPROV = RPROV+RPROV9+RPROV10+RPROV11+RPROV12+RPROV13+RPROV14
RPROV = RPROV+RPROV15+RPROV16+RPROV17+RPROV18+RPROV19+RPROV20
RPROB = RPROB1+RPROB2+RPROB3+RPROB4+RPROB5+RPROB6+RPROB7+RPROB8
RPROB = RPROB+RPROB9+RPROB10+RPROB11+RPROB12+RPROB13+RPROB14
RPROB = RPROB+RPROB15+RPROB16+RPROB17+RPROB18+RPROB19+RPROB20
RPCSS = RPCSS1+RPCSS2+RPCSS3+RPCSS4+RPCSS5+RPCSS6+RPCSS7+RPCSS8
RPCSS = RPCSS+RPCSS9+RPCSS10+RPCSS11+RPCSS12+RPCSS13+RPCSS14
RPCSS = RPCSS+RPCSS15+RPCSS16+RPCSS17+RPCSS18+RPCSS19+RPCSS20

```

I = 1

DO WHILE I < CGCTR

J = STR(I,2)

IF I < 10

J = STR(I,1)

ENDIF

IF UPPER(CADD8J) = 'N'

JJ = CG8J

IF VAL(JJ) < 10

JJ = SUBSTR(JJ,2,1)

ENDIF

IPROV = IPROV - IPROV&JJ

IPROB = IPROB - IPROB&JJ

IPCSS = IPCSS - IPCSS&JJ

RPROV = RPROV - RPROV&JJ

RPROB = RPROB - RPROB&JJ

RPCSS = RPCSS - RPCSS&JJ

ENDIF


```

      I = I + 1
ENDDC
ITOTAL = IPRCV+IPROB+IPOSS
RTOTAL = RPRCV+RPROB+RPOSS
SMTOTAL = SMFCV + SMFROB + SMFOSS
FMTOTAL = FMFCV + FMFROB + FMFOSS
TOTAL = ITOTAL+RTOTAL+SMTOTAL+FMTOTAL+PRCS+PRCG

```

```

*****
PROCEDURE IST
*****

```

```

STOR C TO IPRCV,IPROB,IPCSS,RFCV,RPROB,RFOSS,TOTAL
SELE 6
CC TOP
LOCA FOR MINERAL = CMTY .AND. DEPOSIT = DD
I = 11

```

```

DO WHILE I < 24
  J = STR(I,2)
  GRADE = IG&J
  IF GRADE # SPACE(2)
    II = 1
    DO WHILE II < CGCTR
      JJ = STR(II,2)
      IF II < 10
        JJ = STR(II,1)
      ENDIF
      IF CG&JJ = GRADE
        IF CADD&JJ # 'N'
          IPRCV = IPRCV +IPV&J
          IPROB = IPROB +IPB&J
          IPCSS = IPCSS +IPS&J
          EXIT
        ENDIF
      ENDIF
      II = II + 1
    ENDDO
  ELSE
    EXIT
  ENDIF
  I = I + 1
ENDDC

```

```

I = 11
DO WHILE II < 24
  J = STR(I,2)
  GRADE = RG&J
  IF GRADE # SPACE(2)
    II = 1
    DO WHILE II < CGCTR
      JJ = STR(II,2)
      IF II < 10
        JJ = STR(II,1)
      ENDIF
      IF CG&JJ = GRADE
        IF CADD&JJ # 'N'
          RPROV = RPROV +RPV&J
          RPROB = RPROB +RPB&J
          RFCSS = RFCSS +RPS&J
          EXIT
        ENDIF
      ENDIF
      II = II + 1
    ENDDO
  ELSE

```



```

EXIT
ENDIF
I = I + 1
ENDDC
TOTAL = IPRCV+IPROB+IPCSS+RPRCV+RPROB+RPCSS+IPFV+IPFB+IPFS+SMFV+SMPB+SMPS+PRCS+PRCG
IF TOTAL # 0
  NFLAG = 0
  IF CLCTR = 0
    DO MINHEAD
      IF CS1 = 1
        ? CSTATE
        P1 = LEN(CSTATE)
        ? REPLICATE('-',P1)
        CLCTR = CLCTR + 2
      ELSE
        ? CSTATE
        ? CDIST
        P1 = LEN(CDIST)
        ? REPLICATE('-',P1)
        CLCTR = CLCTR + 3
      ENDIF
    ENDIF
  ENDIF
  ? '***'
  CLCTR = CLCTR + 1
  IF IPRCV+IPROB+IPCSS # 0
    ? 'INSITU RESERVES'
    ? SPACE(21),STR(IPRCV,15,3),STR(IPROB,15,3),STR(IPCSS,15,3)
    ?? SPACE(1)+STR(IPRCV+IPROB+IPCSS,19,3),SPACE(1)+'TOTAL OF ALL GRADES'
    CLCTR = CLCTR + 2
    II = 1
    DO WHILE II < CGCTR
      JJ = STR(II,2)
      IF II < 10
        JJ = STR(II,1)
      ENDIF
      GRADE = CG&JJ
      I = 11
      DO WHILE I < 24
        J = STR(I,2)
        IF IG&J = GRADE
          ? SPACE(21),STR(IPV&J,15,3),STR(IPB&J,15,3),STR(IPS&J,15,3)
          ?? SPACE(1)+STR(IPV&J+IPB&J+IPS&J,19,3),SPACE(1)+CD&JJ
          CLCTR = CLCTR + 1
        ENDIF
        I = I + 1
      ENDWHILE
      II = II + 1
    ENDWHILE
  ENDIF
  IF RPRCV+RPROB+RPCSS # 0
    ? 'RECOVERABLE RESERVES'
    ? SPACE(21),STR(RPRCV,15,3),STR(RPROB,15,3),STR(RPCSS,15,3)
    ?? SPACE(1)+STR(RPRCV+RPROB+RPCSS,19,3),SPACE(1)+'TOTAL OF ALL GRADES'
    CLCTR = CLCTR + 2
    II = 1
    DO WHILE II < CGCTR
      JJ = STR(II,2)
      IF II < 10
        JJ = STR(II,1)
      ENDIF
      GRADE = CG&JJ
      I = 11
      DO WHILE I < 24

```



```

      J = STR(I,2)
      IF RG&J = GRADE
        ? SPACE(21),STR(RFV&J,15,3),STR(RFE&J,15,3),STR(RFS&J,15,3)
        ?? SPACE(1)+STR(RFV&J+RFB&J+RFS&J,19,3),SPACE(1)+CD&J
        CLCTR = CLCTR + 1
        EXIT
      ENDIF
      I = I + 1
    ENDDC
    II = II + 1
  ENDDC
ENDIF
IF SMFV+SMFB+SMPS # 0
  ? 'CONDITIONAL (SUE MARGINAL)'
  ? SPACE(21),STR(SMFV,15,3),STR(SMFB,15,3),STR(SMPS,15,3)
  ?? SPACE(1)+STR(SMFV+SMFB+SMPS,19,3),SPACE(1)+'TOTAL OF ALL GRADES'
  CLCTR = CLCTR + 2
ENDIF
IF PMFV+PMFB+PMPS # 0
  ? 'CONDITIONAL (PARAMARGINAL)'
  ? SPACE(21),STR(PMFV,15,3),STR(PMFB,15,3),STR(PMPS,15,3)
  ?? SPACE(1)+STR(PMFV+PMFB+PMPS,19,3),SPACE(1)+'TOTAL OF ALL GRADES'
  CLCTR = CLCTR + 2
ENDIF
IF PRCS # 0
  ? 'PROSPECTIVE RESOURCES',SPACE(47),STR(PRCS,19,3)
  ?? SPACE(2)+'TOTAL OF ALL GRADES'
  CLCTR = CLCTR + 1
ENDIF
IF PRCS # 0
  ? 'PROGNOSTIC RESOURCES ',SPACE(47),STR(PRCS,19,3)
  CLCTR = CLCTR + 1
ENDIF
? CLINE
? '*** COMMON WITH : '
CLCTR = CLCTR + 2
I = 1
IF CS1 = 1
  DO WHILE I < 10
    J = STR(I,1)
    STAT = SUBSTR(SE&J,1,2)
    IF STAT = SPACE(2)
      EXIT
    ENDIF
    IF STAT # CS
      SELE 3
      GO TOP
      LOCA FOR STATE = STAT
      ?? TRIM(S_DES)+SPACE(1)
      SELE 6
    ENDIF
    I = I+1
  ENDDC
ELSE
  STAT1 = CS
  DO WHILE I < 10
    J = STR(I,1)
    STAT = SUBSTR(SC&J,1,2)
    LIS = SUBSTR(SD&J,3,2)
    IF STAT = SPACE(2)
      EXIT
    ENDIF
    IF LIS # CD
      SELE 3

```



```

      GO TCP
      LOCA FOR STATE = STAT .AND. DIST = DIS
      IF STAT = STAT1
        ?? TRIM(D_DES)+SPACE(1)
      ELSE
        STAT1 = STAT
        ? SPACE(18)+TRIM(S_DES)+' - '+TRIM(D_DES)+SPACE(1)
        CLCTR = CLCTR+1
      ENDIF
      SELE 6
    ENDIF
    I = I+1
  ENDDC
ENDIF
? CLINE
CLCTR = CLCTR + 1
IF CLCTR + CK > 85
  ? SPACE(59)+ COMPUTER
  CLCTR = 0
ENDIF
ENDIF
SKIP
IF .NOT. ECF() .AND. MINERAL = CMCTY
  DE = DEPOSIT
ELSE
  DE = SPACE(4)
ENDIF
SELE 8FN

*****
PROCEDURE SELECT2
*****
CLEAR
INK = 0
USE CCATA
I = 11
DO WHILE .NOT. ECF()
  II=STR(I,2)
  CCDE&II=MINERAL + SPACE(1)
  I = I+1
  SKIP
ENDDC
DO MINERAL
@ 22,8 SAY "Move ARROW Keys and PgUp & PgDn to Select any one COMMODITY"
I=11
J=1
DO WHILE I < 27
  DO CASE
    CASE J = 1
      II=STR(I,2)
    CASE J = 26
      II=STR(I+16,2)
    CASE J = 51
      IF I = 11
        II=STR(I+32,2)
      ELSE
        II=STR(I+31,2)
      ENDIF
    ENDCASE
    @ I-7,J GET CCDE&II
    CLEAR GETS
    INK = INKEY()
    DO CASE
      CASE INK = 4

```



```

      6 I-7,J SAY CCDE8II
      IF J < 27
        IF I = 12 .AND. J = 26
          I = I+1
        ENDIF
        J=J+25
      ENDIF
CASE INK = 19
      6 I-7,J SAY CCDE8II
      IF J > 1
        J=J-25
      ENDIF
CASE INK = 5
      6 I-7,J SAY CCDE8II
      IF I > 11
        IF I = 13 .AND. J = 51
          I = I-2
        ELSE
          I = I-1
        ENDIF
      ELSE
        I=26
      ENDIF
CASE INK = 24
      6 I-7,J SAY CCDE8II
      IF I < 26
        IF I = 11 .AND. J = 51
          I = I+2
        ELSE
          I = I+1
        ENDIF
      ELSE
        I=11
      ENDIF
CASE INK = 18
      6 I-7,J SAY CCDE8II
      I=11
CASE INK = 3
      6 I-7,J SAY CCDE8II
      I=26
CASE INK = 13
      EXIT
ENDCASE
ENDDC
CMDTY = SUBSTR(CCDE8II,1,3)

```

```

*****
PROCEDURE MINERAL
*****

```

```

* A:MINERAL.FRG
6 1,25 SAY "NATIONAL MINERAL INVENTORY"
6 2,15 SAY "Select one of the following commodity codes : "
6 4,1 SAY "AC2 Antimony          FC3 Flucrite          FC2"
6 4,56 SAY "Phosphate"
6 5,1 SAY "AC3 Asbestos          FC4 Fuller's Earth"
6 5,56 SAY "(Phosphorite/Rck FC4)"
6 6,1 SAY "BC2 Barytes          GC1 Garret          FC4"
6 6,56 SAY "Fctash"
6 7,1 SAY "BC4 Bauxite          GC4 Gold            FC5"
6 7,56 SAY "Pyrite"
6 8,1 SAY "BC5 Bencrite          GC5 Graphite         G01"
6 8,56 SAY "Quartz & Silica Sard"
6 9,1 SAY "BC6 Borax            GC6 Gypsum           G02"

```

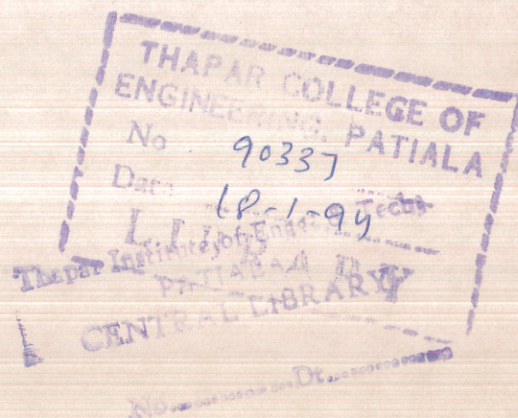

6	9,56	SAY "Quartzite"			
6	10,1	SAY "CC1 Calcite"	IC1	Iron Ore (Haematite)	S02"
6	10,56	SAY "Sillimanite"			
6	11,1	SAY "CC2 Chira Clay"	IC2	Iron Ore (Magnetite)	S04"
6	11,56	SAY "Sulphur (Native)"			
6	12,1	SAY "CC3 Chromite"	K01	Kyanite	T01"
6	12,56	SAY "Talc/Steatite/Soap Stone"			
6	13,1	SAY "CC5 Cobalt"	LC2	Lead-Zinc Ore	T02"
6	13,56	SAY "Tin"			
6	14,1	SAY "CC6 Copper"	LC4	Limestone	T03"
6	14,56	SAY "Titanium"			
6	15,1	SAY "CC7 Corundum"	MC1	Magnesite	T04"
6	15,56	SAY "Tungsten"			
6	16,1	SAY "CC2 Diamond"	MC2	Manganese Ore	V01"
6	16,56	SAY "Vanadium"			
6	17,1	SAY "CC4 Dolomite"	MC5	Molybdenum	V02"
6	17,56	SAY "Vermiculite"			
6	18,1	SAY "FC1 Feldspar"	NC1	Nickel	W01"
6	18,56	SAY "Wollastonite"			
6	19,1	SAY "FC2 Fire Clay"	FC1	Phosphate (Apatite)	Z01"
6	19,56	SAY "Zircon"			

 PROCEDURE SELECT1

```

FLAG = 0
SELE 1
USE CCATA
SELE 2
USE MINFILE
SET INDEX TO MINFILE
SELE 1
SET RELATION TO MINERAL INTO MINFILE
DO MICHEAD WITH "COMMODITY SELECTION"
6 1,64 SAY SPACE(16)
6 6,14 TC 22,65 DCUBLE
6 7,17 SAY "CCDE" + SPACE(2) + "COMMODITY NAME"
6 8,17 SAY "----" + SPACE(2) + "-----"
I = 9
DO WHILE .NOT. EOF()
  MDES = B->M_DES
  MCCDE = MINERAL
  IF I > 21
    6 23,10 SAY SPACE(1)
    WAIT "          PRESS 'E' TO ENTER CODE OR ANY OTHER KEY TO CONTINUE" TO ANS
    IF UPPER(ANS) = 'E'
      EXIT
    ENDIF
    6 9,15 CLEAR TC 21,64
    I = 9
  ENDIF
  6 1,17 SAY MCCDE + SPACE(3) + MDES
  I = I+1
  SKIP
ENDDO
DO WHILE FLAG = 0
  ANS = SPACE(3)
  6 24,0 CLEAR
  6 24,28 SAY "ENTER COMMODITY CCDE"
  6 24,49 GET ANS
  READ
  COMM = UPPER(ANS)

```




```

SELE 2
SEEK CCMM
IF ECF()
  G 24,0 CLEAR
  G 23,75 SAY ""
  WAIT "      ENTERED WRONG CCDE, SO PRESS ANY KEY TO ENTER THE CCDE AGAIN"
ELSE
  EXIT
ENDIF
ENDDC
CMDTY = CCMM

```

```

*****
PROCEDURE MICGHEAD
*****
PARAMETERS SLBHED
CLEAR
? SPACE(32)+'MINERAL DATABASE'+SPACE(16)+"COMMODITY : "+CMDTY
? SPACE(27)+'NATIONAL MINERAL INVENTORY'
P1 = LEN(SLBHED)
P2 = INT(80-P1)/2
? SPACE(P2)+SLBHED
? SPACE(P2)+REPLICATE('-',P1)
RETURN

```



```
*****
** This program is a query program for getting the occurrence of a
** particular commodity in different states in the commodity wise queries.
*****
```

```

SELECT 1
SET FILTER TO
GO TOP
SET INDEX TO TEMP1
SET RELATION TO STATE+DIST INTO STATEFILE
DO WHILE QUERY # 'C' .OR. UPPER(QUERY) # 'E'
    QUERY = SPACE(1)
    DO MICQHEAD WITH 'COMMODITY OCCURENCE'
    @ 6,25,18,53 BOX FRAME
    @ 8,28 SAY "OCCURENCE DETAILS ON : "
    @ 10,33 SAY "1. COMMODITY "
    @ 12,33 SAY "2. STATUS "
    @ 14,33 SAY "3. GRADE "
    @ 16,33 SAY "C. EXIT "
    @ 20,16 SAY "ENTER YOUR CHOICE(C,1,2,3/E,C,S,G/e,c,s,g)"
    @ 20,61 GET QUERY PICTURE "!" VALID(QUERY$ "0123ECSGecsg")
    READ
    DO CASE
        CASE QUERY = 'C' .OR. UPPER(QUERY) = 'E'
            QUERY = SPACE(1)
            RETURN
        CASE QUERY = '1' .OR. UPPER(QUERY) = 'C'
            SET FILTER TO
            GO TOP
            DO MICQ10
        CASE QUERY = '2' .OR. UPPER(QUERY) = 'S'
            DO WHILE .T.
                STORE SPACE(0) TO COND,STATUSA
                STORE SPACE(1) TO SA, SB, SC
                STORE C TO CTR
                @ 19,0 CLEAR
                @ 20,16 SAY " ENTER VALLES FOR STATUS"
                @ 20,46 GET SA PICTURE "!" VALID( SA$ " FLBflb")
                @ 20,53 GET SB PICTURE "!" VALID( SB$ " UPEUpb")
                @ 20,60 GET SC PICTURE "!" VALID( SC$ " CNEcnb")
                @ 21,44 SAY "F/L/B U/P/B C/N/B"
                READ
                SA = UPPER(SA)
                SB = UPPER(SB)
                SC = UPPER(SC)
                IF SA # SPACE(1)
                    COND = 'STATUS_A = SA'
                    CTR = 1
                    DO CASE
                        CASE SA = 'F'
                            SAA = 'FREEHOLD '
                        CASE SA = 'L'
                            SAA = 'LEASEHOLD '
                        CASE SA = 'B'
                            SAA = 'FREEHOLD & LEASEHOLD '
                    ENDCASE
                    STATUSA = STATUSA + SAA
                ENDIF
                IF SB # SPACE(1)
                    IF CTR = 0
                        COND = 'STATUS_B = SB'
                        CTR = 1
                    
```



```

ELSE
  CCND = CCND + '.AND. STATUS_E = SE'
ENDIF
DO CASE
  CASE SB = 'U'
    SBB = 'PUBLIC '
  CASE SB = 'P'
    SBB = 'PRIVATE '
  CASE SB = 'B'
    SBB = 'PUBLIC & PRIVATE '
ENDCASE
STATUSA = STATUSA + SBB
ENDIF
IF SC # SPACE(1)
  IF CTR = 0
    CCND = 'STATUS_C = SC'
    CTR = 1
  ELSE
    CCND = CCND + '.AND. STATUS_C = SC'
  ENDIF
DO CASE
  CASE SC = 'C'
    SCC = 'CAPTIVE'
  CASE SC = 'N'
    SCC = 'NON-CAPTIVE'
  CASE SC = 'B'
    SCC = 'CAPTIVE & NON-CAPTIVE'
ENDCASE
STATUSA = STATUSA + SCC
ENDIF
SET FILTER TO &CCND
GO TOP
IF EOF()
  & 20,C CLEAR
  & 21,27 SAY " NO DEPOSIT FOR THIS STATUS"
  ?
  WAIT "                                Press any key to SELECT other STATUS"
  LOOP
ELSE
  DO MICQ10
  EXIT
ENDIF
ENDDO
CASE QUERY = '3' .OR. UPPER(QUERY) = 'G'
  & 5,C CLEAR
  & 5,1 SAY "GRADE SELECTION : "
  ?
  ? SPACE(1)+'Ccde'+SPACE(2)+'Grade Name',SPACE(21),'Code'+SPACE(2)+'Grade Name'
  ? SPACE(1)+'----'+SPACE(2)+'-----',SPACE(21),'----'+SPACE(2)+'-----'
  I = 1
DO WHILE I < CGCTR
  IF I < 10
    J = STR(I,1)
  ELSE
    J = STR(I,2)
  ENDIF
  IF (I/2) # INT(I/2)
    ? SPACE(1),CG&J,SPACE(1),CD&J
  ELSE
    ?? SPACE(1),CG&J,SPACE(1),CD&J
  ENDIF
  I = I + 1
ENDDO
& 6,C,20,79 BCX FRAME1

```



```

DO WHILE .T.
  GRADE = SPACE(2)
  @ 22,30 SAY "ENTER GRADE CODE" GET GRADE PICTURE "!!"
  READ
  IF GRADE # SPACE(2)
    I = 1
    DO WHILE I < CGCTR
      IF I < 10
        J = STR(I,1)
      ELSE
        J = STR(I,2)
      ENDIF
      IF CG8J = GRADE
        GRADED = CG8J
        EXIT
      ENDIF
      I = I + 1
    ENDDO
    SET FILTER TO (IGRADE = GRADE) .OR. (RGRADE = GRADE)
    GO TOP
    IF ECF()
      @ 23,27 SAY "NO DEPOSIT FOR THIS GRADE"
      YN = ' '
      WAIT "                                Do you want to see for another GRADE" TO YN
      IF UPPER(YN) = 'Y'
        @ 21,C CLEAR
      ELSE
        EXIT
      ENDIF
    ELSE
      DO MICQ10
      EXIT
    ENDIF
  ENDIF
ENDDO
ENDCASE
ENDDC
SET INDEX TO
SET RELATION TO

```



```

*****
** This program is a query subprogram for getting the occurrence of a
** commodity in different states.
*****

```

```

6 5,C CLEAR
6 4,75 SAY ""
? 'COMMODITY :',CMIN,'AS CN :',CS_DATE
DO CASE
  CASE GLERY = '2' .CR. UPPER(QUERY) = 'S'
    ? 'STATUS :',STATUSA
  CASE GLERY = '3' .CR. UPPER(QUERY) = 'G'
    ? 'GRADE :',GRADED
ENDCASE
? CLINE
? SPACE(20)+'IS PRESENT IN THE FOLLOWING STATES : '
6 9,7,22,72 ECX FRAME1
CS = SPACE(2)
CTR = 0
CTR1 = 1
LINE = 10
DO WHILE .NOT. EOF()
  IF STATE # CS
    IF (CTR1/2) # INT(CTR1/2)
      6 LINE,9 SAY STR(CTR1,2)+SPACE(2)+C->S_DES
    ELSE
      6 LINE,40 SAY STR(CTR1,2)+SPACE(2)+C->S_DES
      LINE = LINE + 1
    ENDIF
    CS = STATE
    CTR = CTR + 1
    CTR1 = CTR1 + 1
    IF CTR = 24
      CTR = 0
      6 23,C SAY ""
      WAIT "          Press any key to continue..."
      6 10,8 CLEAR TO 21,71
      LINE = 10
    ENDIF
  ENDIF
  SKIP
ENDDO
IF CTR # C
  6 23,C SAY ""
  WAIT "          Press any key to continue..."
ENDIF
RETURN

```



```

*****
** This program is a query program for getting the reserve position
** of a commodity in the commodity wise query. This is the main program.
*****

```

```

SELE 1
SET INDEX TO
DO WHILE QUERY # 'C' .OR. UPPER(QUERY) # 'E'
  QUERY = SPACE(1)
  DO MICGHEAD WITH 'RESERVE POSITION'
  @ 6,26,18,53 BOX FRAME
  @ 8,29 SAY "RESERVE POSITION CN : "
  @ 10,34 SAY "1. COMMODITY "
  @ 12,34 SAY "2. STATUS      "
  @ 14,34 SAY "3. GRADE       "
  @ 16,34 SAY "C. EXIT        "
  @ 20,16 SAY "ENTER YOUR CHOICE(0,1,2,3/E,C,S,G/e,c,s,g)"
  @ 20,61 GET QUERY PICTURE "!" VALID(QUERY$ "C123ECSGecsg")
  READ
  DO CASE
    CASE QUERY = 'C' .OR. UPPER(QUERY) = 'E'
      QUERY = SPACE(1)
      RETURN
    CASE QUERY = '1' .OR. UPPER(QUERY) = 'C'
      SET FILTER TO
      GO TOP
      DO MICG20
    CASE QUERY = '2' .OR. UPPER(QUERY) = 'S'
      DO WHILE .T.
        STORE SPACE(0) TO CCND,STATUSA
        STORE SPACE(1) TO SA, SE, SC
        STORE C TO CTR
        @ 19,0 CLEAR
        @ 20,16 SAY " ENTER VALUES FOR STATUS"
        @ 20,46 GET SA PICTURE "!" VALID(SA$ " FLBflb")
        @ 20,53 GET SB PICTURE "!" VALID(SE$ " LPBLpb")
        @ 20,60 GET SC PICTURE "!" VALID(SC$ " CNBcrb")
        @ 21,44 SAY "F/L/B  U/F/B  C/N/E"
        READ
        SA = UPPER(SA)
        SB = UPPER(SB)
        SC = UPPER(SC)
        IF SA # SPACE(1)
          CCND = 'STATUS_A = SA'
          CTR = 1
          DO CASE
            CASE SA = 'F'
              SAA = 'FREEHOLD '
            CASE SA = 'L'
              SAA = 'LEASEHOLD '
            CASE SA = 'B'
              SAA = 'FREEHOLD & LEASEHOLD '
          ENDCASE
          STATUSA = STATUSA + SAA
        ENDIF
        IF SB # SPACE(1)
          IF CTR = 0
            CCND = 'STATUS_B = SE'
            CTR = 1
          ELSE
            CCND = CCND + '.AND. STATUS_B = SE'
          ENDIF
        ENDIF
      ENDWHILE
    CASE QUERY = '3' .OR. UPPER(QUERY) = 'G'
      GO TOP
      DO MICG20
  ENDCASE
ENDWHILE

```



```

DO CASE
  CASE SB = 'U'
    SBB = 'PUBLIC '
  CASE SB = 'P'
    SBB = 'PRIVATE '
  CASE SB = 'B'
    SBB = 'PUBLIC & PRIVATE '
ENDCASE
STATUSA = STATUSA + SBB
ENDIF
IF SC # SPACE(1)
  IF CTR = 0
    CCND = 'STATUS_C = SC'
    CTR = 1
  ELSE
    CCND = CCND + '.AND. STATUS_C = SC'
  ENDIF
ENDCASE
CASE SC = 'C'
  SCC = 'CAPTIVE'
CASE SC = 'N'
  SCC = 'NON-CAPTIVE'
CASE SC = 'B'
  SCC = 'CAPTIVE & NON-CAPTIVE'
ENDCASE
STATUSA = STATUSA + SCC
ENDIF
SET FILTER TO &CCND
GO TOP
IF EOF()
  & 20,0 CLEAR
  & 21,31 SAY " NO DEPOSIT FOR THIS STATUS"
  ?
  WAIT "                                Press any other key to SELECT other STATUS"
  LOOP
ELSE
  DO MICQ20
  EXIT
ENDIF
ENDDO
CASE QUERY = '3' .OR. UPPER(QUERY) = 'G'
  & 5,0 CLEAR
  & 5,1 SAY "GRADE SELECTION :"
  ?
  ? SPACE(1)+'Code'+SPACE(2)+'Grade Name',SPACE(21),'Code'+SPACE(2)+'Grade Name'
  ? SPACE(1)+'-----'+SPACE(2)+'-----',SPACE(21),'-----'+SPACE(2)+'-----'
  I = 1
DO WHILE I < CGCTR
  IF I < 10
    J = STR(I,1)
  ELSE
    J = STR(I,2)
  ENDIF
  IF (I/2) # INT(I/2)
    ? SPACE(1),CG&J,SPACE(1),CD&J
  ELSE
    ?? SPACE(1),CG&J,SPACE(1),CD&J
  ENDIF
  I = I + 1
ENDDO
& 6,0,20,79 BOX FRAME1
DO WHILE .T.
  GRADE = SPACE(2)
  & 22,30 SAY "ENTER GRADE CODE" GET GRADE PICTURE "!!"

```



```

READ
IF GRADE # SPACE(2)
  I = 1
  DO WHILE I < CGCTR
    IF I < 10
      J = STR(I,1)
    ELSE
      J = STR(I,2)
    ENDIF
    IF CG&J = GRADE
      GRADED = CG&J
      EXIT
    ENDIF
    I = I + 1
  ENDDO
  SET FILTER TO (IGRADE = GRADE) .OR. (RGRADE = GRADE)
  GO TOP
  IF ECF()
    @ 23,27 SAY "NO DEPOSIT FOR THIS GRADE"
    WAIT "                                Press any key to SELECT other GRADE"
    @ 21,C CLEAR
  ELSE
    DO MICG2C
    EXIT
  ENDIF
ENDIF
ENDDO
ENDCASE
ENDDC

```



```

*****
** This program is a query program for getting the reserve position
** of a commodity in the commodity wise query.
*****

```

```

STORE 0 TO IFROV,IFROB,IFCSS,RFCRV,RFCRB,RFCSS,SMFROV,SMFROB,SMPOSS
STORE 0 TO PMFROV,PMFROB,PMPCSS,FROS,PROG
DO WHILE .NOT. EOF()

```

```

    DO CASE

```

```

        CASE QUERY = '3' .OR. UPPER(QUERY) = 'G'

```

```

            IF IGRADE = GRADE

```

```

                I = 1

```

```

                DO WHILE I < CGCTR

```

```

                    J = STR(I,2)

```

```

                    IF I < 10

```

```

                        J = STR(I,1)

```

```

                    ENDIF

```

```

                    IF CG&J = GRADE

```

```

                        IF UPPER(CADD&J) # 'N'

```

```

                            IFROV = IFROV + IFROVEC

```

```

                            IFROB = IFROB + IFROBAELE

```

```

                            IFCSS = IFCSS + IFCSSIELE

```

```

                            EXIT

```

```

                        ENDIF

```

```

                    ENDIF

```

```

                    I = I + 1

```

```

                ENDDO

```

```

            ENDIF

```

```

            IF RGRADE = GRADE

```

```

                I = 1

```

```

                DO WHILE I < CGCTR

```

```

                    J = STR(I,2)

```

```

                    IF I < 10

```

```

                        J = STR(I,1)

```

```

                    ENDIF

```

```

                    IF CG&J = GRADE

```

```

                        IF UPPER(CADD&J) # 'N'

```

```

                            RFCRV = RFCRV + RFCRVEC

```

```

                            RFCRB = RFCRB + RFCRBAELE

```

```

                            RFCSS = RFCSS + RFCSSIELE

```

```

                            EXIT

```

```

                        ENDIF

```

```

                    ENDIF

```

```

                    I = I + 1

```

```

                ENDDO

```

```

            ENDIF

```

```

        OTHERWISE

```

```

            IF IGRADE # SPACE(2)

```

```

                I = 1

```

```

                DO WHILE I < CGCTR

```

```

                    J = STR(I,2)

```

```

                    IF I < 10

```

```

                        J = STR(I,1)

```

```

                    ENDIF

```

```

                    IF CG&J = IGRADE

```

```

                        IF UPPER(CADD&J) # 'N'

```

```

                            IFROV = IFROV + IFROVEC

```

```

                            IFROB = IFROB + IFROBAELE

```

```

                            IFCSS = IFCSS + IFCSSIELE

```

```

                            EXIT

```

```

                        ENDIF

```

```

                    ENDIF

```



```

        I = I + 1
    ENDDC
ENDIF
IF RGRADE # SPACE(2)
    I = 1
    DO WHILE I < CGCTR
        J = STR(I,2)
        IF I < 10
            J = STR(I,1)
        ENDIF
        IF CG&J = RGRADE
            IF UPPER(CADE&J) # 'N'
                RPROV = RPROV + RPROVED
                RPROB = RPROB + RPROBABLE
                RPCSS = RPCSS + RPCSSIBLE
            EXIT
            ENDIF
        ENDIF
        I = I + 1
    ENDDC
ENDIF
IF PMGRADE # SPACE(25)
    PMPROV = PMPROV + PMPROVED
    PMPROB = PMPROB + PMPROBABLE
    PMPCSS = PMPCSS + PMPOSSIBLE
ENDIF
IF SMGRADE # SPACE(25)
    SMPROV = SMPROV + SMPROVED
    SMPROB = SMPROB + SMPROBABLE
    SMPCSS = SMPCSS + SMPOSSIBLE
ENDIF
PRCG = PRCG + PROCGTY
PRCS = PRCS + PROSGTY

ENDCASE
SKIP
ENDDC
TOTAL = IFROV+IPROB+IFCSS+RPROV+RPROB+RPCSS+SMPROV+SMPROB+SMPCSS
TOTAL = TOTAL+PMPROV+PMPROB+PMPCSS+PRCG+PRCS
6 5,C CLEAR
6 4,75 SAY ""
? 'COMMUNITY :',CMIN,SPACE(3),'UNIT :',CLNITDES
? 'AS ON      :',CS_DATE
DC CASE
    CASE QUERY = '2' .OR. UPPER(QUERY) = 'S'
        ? 'STATUS      :',STATLSA
    CASE QUERY = '3' .OR. UPPER(QUERY) = 'G'
        ? 'GRADE       :',GRADED
ENDCASE
? CLINE
? SPACE(24)+'PROVED'+SPACE(7)+'PROBABLE'+SPACE(7)+'POSSIBLE'+SPACE(14)+'TOTAL'
? CLINE
IF TOTAL # 0
    IF (IFROV+IPROB+IFCSS) # 0
        ? 'INSITU RESERVES'
        ? SPACE(15),STR(IFROV,14,2),STR(IPROB,14,2),STR(IFCSS,14,2),STR(IFROV+IPROB+IFCSS,18,2)
    ENDIF
    IF (RPROV+RPROB+RPCSS) # 0
        ? 'RECOVERABLE RESERVES'
        ? SPACE(15),STR(RPROV,14,2),STR(RPROB,14,2),STR(RPCSS,14,2),STR(RPROV+RPROB+RPCSS,18,2)
    ENDIF
    IF QUERY # '3' .OR. UPPER(QUERY) # 'G'
        IF (SMPROV+SMPROB+SMPCSS) # 0
            ? 'CONDITIONAL (SUBMARGINAL) RESOURCES'
            ? SPACE(15),STR(SMPROV,14,2),STR(SMPROB,14,2),STR(SMPCSS,14,2),STR(SMPROV+SMPROB+SMPCSS,18,2)
        ENDIF
    ENDIF
ENDIF

```



```

ENDIF
IF (FMFRCV+FMFRCB+FMFCSS) # 0
  ? 'CONDITIONAL (PARAMARGINAL) RESOURCES'
  ? SPACE(15),STR(FMFRCV,14,2),STR(FMFRCB,14,2),STR(FMFCSS,14,2),STR(FMFRCV+FMFRCB+FMFCSS,1
ENDIF
IF FROS # 0
  ? 'PROSPECTIVE RESOURCES',SPACE(38),STR(FROS,18,2)
ENDIF
IF FRCG # 0
  ? 'PROGNOSTIC RESOURCES ',SPACE(38),STR(FRCG,18,2)
ENDIF
ENDIF
ELSE
  ? SPACE(25)+'-----NOT ESTIMATED-----'
ENDIF
? CLINE
?
WAIT
RETURN

```



```

*****
** This program is a query subprogram for getting the detailed
** information of a deposit in the commodity wise queries.
*****

```

```

DC WHILE QUERY # 'C' .OR. UPPER(QUERY) # 'E'
  QUERY = SPACE(1)
  DC MICGHEAD WITH 'DEPOSIT DETAILS'
  @ 6,12,16,64 BOX FRAME
  @ 8,15 SAY "1. TOTAL NO. OF DEPOSITS ESTIMATED/ UNESTIMATED"
  @ 10,15 SAY "2. STATUS WISE COUNT OF DEPOSITS"
  @ 12,15 SAY "3. RESERVE POSITION OF A PARTICULAR DEPOSIT"
  @ 14,15 SAY "C. EXIT"
  @ 18,16 SAY "ENTER YOUR CHOICE(C,1,2,3,E,T,S,R,e,t,s,r)"
  @ 18,61 GET QUERY PICTURE "!" VALID(QUERY$ "0123ETSReter")
  READ
  DC CASE
    CASE QUERY = 'C' .OR. UPPER(QUERY) = 'E'
      QUERY = SPACE(1)
      RETURN
    CASE QUERY = '1' .OR. UPPER(QUERY) = 'T'
      SELE 1
      SET INDEX TO
      SET FILTER TO
      GC TOP
      DC MICQ31
    CASE QUERY = '2' .OR. UPPER(QUERY) = 'S'
      STATUSA = SPACE(0)
      SELE 1
      SET INDEX TO
      STORE SPACE(0) TO COND, STATUSA
      STORE SPACE(1) TO SA, SB, SC
      STORE C TO CTR
      @ 17,0 CLEAR
      @ 18,16 SAY " ENTER VALUES FOR STATUS"
      @ 18,46 GET SA PICTURE "!" VALID( SA$ " FLEflb")
      @ 18,53 GET SB PICTURE "!" VALID( SB$ " UFEupb")
      @ 18,60 GET SC PICTURE "!" VALID( SC$ " CNEcnb")
      @ 19,44 SAY "F/L/E  U/F/B  C/N/B"
      READ
      SA = UPPER(SA)
      SB = UPPER(SB)
      SC = UPPER(SC)
      IF SA # SPACE(1)
        COND = 'STATUS_A = SA'
        CTR = 1
        DC CASE
          CASE SA = 'F'
            SAA = 'FREEHOLD '
          CASE SA = 'L'
            SAA = 'LEASEHOLD '
          CASE SA = 'B'
            SAA = 'FREEHOLD & LEASEHOLD '
        ENDCASE
        STATUSA = STATUSA + SAA
      ENDIF
      IF SB # SPACE(1)
        IF CTR = 0
          COND = 'STATUS_E = SB'
          CTR = 1
        ELSE
          COND = COND + '.AND. STATUS_E = SB'

```



```

ENDIF
DO CASE
  CASE SB = 'U'
    SBE = 'PUBLIC '
  CASE SB = 'P'
    SBE = 'PRIVATE '
  CASE SB = 'B'
    SBE = 'PUBLIC & PRIVATE '
ENDCASE
STATUSA = STATUSA + SBE
ENDIF
IF SC # SPACE(1)
  IF CTR = 0
    CCND = 'STATUS_C = SC'
    CTR = 1
  ELSE
    CCND = CCND + '.AND. STATUS_C = SC'
  ENDIF
DO CASE
  CASE SC = 'C'
    SCC = 'CAPTIVE'
  CASE SC = 'N'
    SCC = 'NON-CAPTIVE'
  CASE SC = 'B'
    SCC = 'CAPTIVE & NON-CAPTIVE'
ENDCASE
STATUSA = STATUSA + SCC
ENDIF
SET FILTER TO &CCND
GC TOP
IF ECF()
  & 21,31 SAY " NO DEPOSIT FOR THIS STATUS"
  ?
  WAIT
  LOCF
ELSE
  DO MICG31
ENDIF
CASE QUERY = '3' .OR. UPPER(QUERY) = 'R'
  SELE 1
  SET INDEX TO
  SET FILTER TO
  GC TOP
  DO MICG33
ENDCASE
ENDDC

```

 ** This program is a query subprogram for getting various estimated
 ** and not estimated deposits in the commodity wise queries.

```

STORE 0 TO ES, UN, LC
FLAG = .F.
AA = "IGRADE # SPACE(2) .OR. RGRADE # SPACE(2) .OR. SMGRADE # SPACE(25)"
EB = "PMGRADE # SPACE(25) .OR. PROCSGRADE # SPACE(25) .OR. PROCGTY # C"
?SPACE(15), "PROCESSING IN PROGRSS, PLEASE STAND BY..."
DO WHILE .NOT. EOF()
  DO = DEPOSIT
  DO WHILE VAL(DEPOSIT) = VAL(DO)
    IF 8AA .OR. 8BB
      FLAG = .T.
    ENDIF
    SKIP
  ENDDO
  IF FLAG
    ES = ES + 1
    FLAG = .F.
  ELSE
    UN = UN + 1
    SKIP -1
    DELE
    SKIP
  ENDIF
ENDDO
@ 5,C CLEAR
@ 5,C SAY "COMMODITY : "+CMIN
IF QUERY = "2"
  @ 6,C SAY "STATUS : "+STATLSA
ENDIF
? CLINE
?
?
? SPACE(27)+"NO. OF DEPOSITS ESTIMATED      =",STR(ES,4)
? SPACE(27)+"NO. OF DEPOSITS NOT ESTIMATED =",STR(UN,4)
? SPACE(27)+"TOTAL NO. OF DEPOSITS        =",STR(ES+UN,4)
?
WAIT "                                     Press any key to continue..."
IF UN > 0
  YN = "N"
  @ 16,25 SAY "DO YOU WANT LIST OF UNESTIMATED DEPOSITS" GET YN VALID(YN$ "YNyn")
  READ
  IF UPPER(YN) = "Y"
    GO TOP
    LC = 0
    @ 7,C CLEAR
    @ 7,15 SAY "LIST OF UNESTIMATED DEPOSITS"
    @ 8,10,22,65 BOX FRAME1
    @ 9,15 SAY "DEPOSIT NO.    DEPOSIT NAME"
    DO WHILE .NOT. EOF()
      IF LC = 0
        @ 10,11 CLEAR TO 21,64
        LINE = 10
      ENDIF
      IF DELETED()
        @ LINE,19 SAY DEPOSIT+SPACE(7)+DEFNO
        LC = LC + 1
        LINE = LINE + 1
      IF LC = 12

```



```
      @ 22,C SAY ""
      WAIT "
      LC = C
      ENDIF
    ENDIF
  SKIP
  ENDEC
ENDIF
IF LC # C
  @ 22,C SAY ""
  WAIT "
  Press any key to continue..."
ENDIF
RECALL ALL
RETURN
```

** This program is a query sub program for getting the detailed
 ** information of a particular deposit in the commodity wise queries.

*** CMDTY : CODE OF COMMODITY BEING PROCESSED; DFLAG : IS 1 IF ID > 1 ***

*** CD : CURRENT DISTRICT CODE; CDIST : CURRENT DISTRICT NAME ***

*** CS : CURRENT STATE CODE; CSTATE : CURRENT STATE NAME ***

YN = ' '

DO WHILE QUERY # 'O' .OR. UPPER(QUERY) # 'E'

STORE SPACE(4) TO DEP

STORE SPACE(2) TO CS,CD,BLCK,CSTATE,CDIST

STORE SPACE(1) TO DEP1

STORE C TO BLK,DFLAG

QUERY = SPACE(1)

CLEAR

DO MICGHEAD WITH 'DETAILS OF DEPOSIT'

@ 6,18,16,55 BOX FRAME

@ 8,24 SAY "DEPOSIT ACCESS BY : "

@ 10,21 SAY "1. FIRST LETTER OF DEPOSIT NAME "

@ 12,21 SAY "2. STATE NAME "

@ 14,21 SAY "C. EXIT "

@ 18,16 SAY "ENTER YOUR CHOICE(C,1,2/E,F,S/e,1,s)" GET QUERY PICTURE "!" VALID(QUERY\$ "C12EFSeF

READ

DO CASE

CASE QUERY = 'C' .OR. UPPER(QUERY) = 'E'

QUERY = SPACE(1)

RETURN

CASE QUERY = '1' .OR. UPPER(QUERY) = 'F'

SELE 1

SET INDEX TO TEMP

SET FILTER TO

GO TOP

@20,10 SAY "ENTER THE FIRST LETTER OF THE NAME OF THE DEPOSIT" GET DEP1

READ

DEP1 = UPPER(DEP1)

SEEK DEP1

IF .NOT. EOF()

@ 5,0 CLEAR

@ 5,10 SAY "DEPOSIT SELECTION :"

@ 6,10,22,70 BOX FRAME1

CTR = 0

LINE = 8

DD = "CCCC"

@ 7,15 SAY "DEPOSIT NO DEPOSIT NAME"

DO WHILE SUBSTR(DEPNO,1,1) = DEP1

IF VAL(DEPOSIT) # VAL(DD)

IF CTR = 14

@ 23,0 SAY ""

WAIT ' Press "E" to ENTER DEPOSIT NO., any other key to continue'

IF UPPER(YN) = 'E'

EXIT

ENDIF

@ 23,0 CLEAR

CTR = 0

LINE = 8

@ 8,11 CLEAR TO 21,69

ENDIF

@ LINE,18 SAY DEPOSIT+SPACE(8)+DEPNO

DD = DEPOSIT

CTR = CTR + 1

LINE = LINE + 1


```

        ENDIF
        SKIP
    ENDDC
    @ 23,C CLEAR
    DD = SPACE(4)
    @ 23,16 SAY "ENTER DEPOSIT NUMBEER AND PRESS "+CHR(17)+CHR(196)+CHR(217)+" KEY" GET C
    READ
    IF DD # SPACE(4)
        SET FILTER TO DEPOSIT = DD
        COPY TO TEMP1
        SELE 6
        USE TEMP1
    ELSE
        LOOP
    ENDIF
ELSE
    ?
    ? SPACE(15), 'NO DEPOSIT NAME STARTS WITH',DEP1
    WAIT
    DD = 'CCCC'
    LOOP
ENDIF
CASE QUERY = '2' .CR. UPPER(QUERY) = 'S'
SELE 1
SET FILTER TO
GC TOP
SET INDEX TO TEMP1
SET RELATION TO STATE+DIST INTO STATFILE
CS = SPACE(2)
@ 5,C CLEAR
@ 6,7,22,72 BOX FRAME1
@ 5,7 SAY "STATE SELECTION : "
@ 7,9 SAY "CODE"+SPACE(2)+"STATE NAME"+SPACE(15)+"CODE"+SPACE(2)+"STATE NAME"
CTR = 0
CTR1 = 1
LINE = 9
DO WHILE .NOT. EOF()
    IF STATE # CS
        IF (CTR1/2) # INT(CTR1/2)
            @ LINE,10 SAY STATE+SPACE(3)+C->S_DES
        ELSE
            @ LINE,41 SAY STATE+SPACE(3)+C->S_DES
            LINE = LINE + 1
        ENDIF
        CS = STATE
        CTR = CTR + 1
        CTR1 = CTR1 + 1
        IF CTR = 24
            CTR = 0
            @ 23,C SAY ""
            WAIT "
            IF UPPER(YN) = 'E'
                EXIT
            ENDIF
            @ 8,8 CLEAR TO 21,71
            LINE = 10
        ENDIF
    ENDIF
SKIP
ENDDC
CS = SPACE(2)
@ 23,C CLEAR
@ 23,29 SAY "ENTER STATE CODE" GET CS
READ

```

Press E to ENTER CODE, any other key to continue" TO


```

SEEK CS
CD = SPACE(2)
@ 5,0 CLEAR
@ 2,64 SAY "STATE      : "+CS
@ 6,7,22,72 BCX FRAME1
@ 5,7 SAY "DISTRICT SELECTION : "
@ 7,9 SAY "CODE"+SPACE(2)+"DISTRICT NAME"+SPACE(12)+"CODE"+SPACE(2)+"DISTRICT NAME"
CTR = C
CTR1 = 1
LINE = 9
DO WHILE STATE = CS
  IF DIST # CD
    IF (CTR1/2) # INT(CTR1/2)
      @ LINE,10 SAY DIST+SPACE(3)+C->D_DES
    ELSE
      @ LINE,41 SAY DIST+SPACE(3)+C->D_DES
      LINE = LINE + 1
    ENDIF
    CD = DIST
    CTR = CTR + 1
    CTR1 = CTR1 + 1
    IF CTR = 24
      CTR = C
      @ 23,C SAY ""
      WAIT "                      Press E to ENTER CODE, any other key to continue" TO YN
      IF UPPER(YN) = 'E'
        EXIT
      ENDIF
      @ 10,8 CLEAR TO 21,71
      LINE = 10
    ENDIF
  ENDIF
  SKIP
ENDDO
CD = SPACE(2)
@ 23,C CLEAR
@ 23,27 SAY "ENTER DISTRICT CODE" GET CD
READ
SEEK CS+CD
CTR = C
IF .NOT. EOF()
  @ 5,0 CLEAR
  @ 3,64 SAY "DISTRICT : "+CD
  @ 5,10 SAY "DEPCST SELECTION : "
  @ 6,10,22,70 BCX FRAME1
  CTR = C
  LINE = 8
  DD = 'CCCC'
  @ 7,15 SAY "DEPCST NO      DEPCST NAME"
  DO WHILE STATE = CS .AND. DIST = CD
    IF VAL(DEPOSIT) # VAL(CD)
      IF CTR = 14
        @ 23,C SAY ""
        WAIT "                      Press "E" to ENTER DEPCST NO., any other key to continue"
        IF UPPER(YN) = 'E'
          EXIT
        ENDIF
        @ 23,C CLEAR
        CTR = C
        LINE = 8
        @ 8,11 CLEAR TO 21,69
      ENDIF
      @ LINE,18 SAY DEPCST+SPACE(8)+DEFNO
      CD = DEPCST

```



```

        CTR = CTR + 1
        LINE = LINE + 1
    ENDIF
    SKIP
ENDDC
@ 23,C CLEAR
DD = SPACE(4)
@ 23,18 SAY "ENTER DEPOSIT NUMBER AND PRESS "+CHR(17)+CHR(196)+CHR(217)+" KEY" GET D
READ
IF DD # SPACE(4)
    SET FILTER TO DEPOSIT = DD
    COPY TO TEMP1
    SELE 6
    USE TEMP1
ELSE
    LOOP
ENDIF
ELSE
    ?
    ? SPACE(15), 'NO DEPOSIT IN THIS DISTRICT'
    WAIT
    DD = 'CCCC'
    LOOP
ENDIF
ENDCASE
*** INITIALIZE VARIABLES FOR STORING VARIOUS TOTALS
I = 1
DO WHILE I < 21
    J = STR(I,2)
    IF I < 10
        K = STR(I,1)
    ENDIF
    STORE C TO IPRCV&J,IPRCB&J,IPCSS&J,RPRCV&J,RPRCB&J,RPOSS&J
    I = I + 1
ENDDO
STORE C TO IPRCV,IPRCB,IPOSS,RPRCV,RPRCB,RPCSS,SMPRCV,SMPRCB,SMPOSS
STORE C TO FMPRCV,FMPRCB,PMPCSS,PROG,PROS,ITCTAL,RICTAL,SMICTAL
STORE C TO FMTCTAL,TOTAL
*** PROCESSING STARTS
DO WHILE .NOT. EOF()
    *** MAKE A NOTE OF INTER STATE OR INTER DISTRICT DEPOSITS
    IF ID # '1'
        DFLAG = 1
    ENDIF
    IF ELOCK # BLOC
        ELK = BLK + 1
    ENDIF
    *** ADD GRADE WISE RESERVES
    DO MIGRADD
    SKIP
ENDDO
*** DO FINAL TALLING
DO MIADD
*** PRINTING STARTS
*** PRINT TOP HEADING
*DO MICHEAD WITH 'DEPOSIT DETAILS'
GC TOP
IF CS = SPACE(2)
    CS = STATE
    CD = DIST
ENDIF
SELE 3
SEEK CS+CD
CSTATE = S_DES

```



```

CCIST = C_DES
SELE 6
CLEAR
? 'MINERAL : '+CMIN+SPACE(5)+'UNIT : '+CUNITDES
? 'DEFCST : '+DEFNC+SPACE(9)+'AS CN : '+CS_DATE
? 'STATE : '+CSTATE+SPACE(5)+'DISTRICT : '+CCIST
? CLINE
? SPACE(8)+'PROVED'+SPACE(7)+'PROBABLE'+SPACE(7)+'POSSIBLE'+SPACE(2)+'GRADE'
? CLINE
? 'INSITU RESERVES'
IF ITCTAL # 0
  ? STR(IPROV,14,2),STR(IPROE,14,2),STR(IPCSS,14,2)+SPACE(2)+'TOTAL OF ALL GRADES'
  I = 1
  DO WHILE I < CGCTR
    J = STR(I,2)
    IF I < 10
      J = STR(I,1)
    ENDIF
    JJ = CG8J
    IF VAL(JJ) < 10
      JJ = SUBSTR(JJ,2,1)
    ENDIF
    IF (IPROV&JJ + IPROE&JJ + IPCSS&JJ) # 0
      ? STR(IPROV&JJ,14,2),STR(IPROE&JJ,14,2),STR(IPCSS&JJ,14,2)+SPACE(2)+CD&J
    ENDIF
    I = I + 1
  ENDDO
ELSE
  ? SPACE(20)+'-----NOT ESTIMATED-----'
ENDIF
? CLINE
WAIT
& 7,0 CLEAR
& 7,0 SAY 'RECOVERABLE RESERVES'
IF RTCTAL # 0
  ? STR(RPROV,14,2),STR(RPROE,14,2),STR(RPCSS,14,2)+SPACE(2)+'TOTAL OF ALL GRADES'
  I = 1
  DO WHILE I < CGCTR
    J = STR(I,2)
    IF I < 10
      J = STR(I,1)
    ENDIF
    JJ = CG8J
    IF VAL(JJ) < 10
      JJ = SUBSTR(JJ,2,1)
    ENDIF
    IF (RPROV&JJ + RPROE&JJ + RPCSS&JJ) # 0
      ? STR(RPROV&JJ,14,2),STR(RPROE&JJ,14,2),STR(RPCSS&JJ,14,2)+SPACE(2)+CD&J
    ENDIF
    I = I + 1
  ENDDO
ELSE
  ? SPACE(20)+'-----NOT ESTIMATED-----'
ENDIF
? CLINE
WAIT
& 7,0 CLEAR
IF SMCTAL + PMTCTAL + PROS + FROG # 0
  IF SMCTAL # 0
    ? 'CONDITIONAL (SUBMARGINAL)'
    ? STR(SMPROV,14,2),STR(SMPROE,14,2),STR(SMPCSS,14,2)+SPACE(2)+'TOTAL OF ALL GRADES'
  ENDIF
  IF PMTCTAL # 0
    ? 'CONDITIONAL (PARAMARGINAL)'
  ENDIF

```



```

      ? STR(FMPCOV,14,2),STR(FMPRCE,14,2),STR(FMPCSS,14,2)+SPACE(2)+'TOTAL OF '
ENDIF
IF FROS # C
      ? 'PROSPECTIVE RESOURCES',SPACE(3),STR(FROS,18,2)+SPACE(2)+'TOTAL OF ALL GRAL
ENDIF
IF FROG # C
      ? 'PROGNOSTIC RESOURCES ',SPACE(3),STR(FROG,18,2)
ENDIF
? CLINE
WAIT
ENDIF
IF DFLAG = 1
GO TOP
? 'THIS DEPOSIT ALSO SPREADS OVER FOLLOWING STATES AND DISTRICTS :"'
?
?
DO WHILE .NOT. EOF()
  IF STATE # CS .OR. DIST # CD
    CSS = STATE
    CDD = DIST
    SELE 3
    SEEK CSS+CDD
    IF STATE # CS
      ? TRIM(S_DES)+' : ',D_DES+SPACE(2)
    ELSE
      ?? TRIM(D_DES)+SPACE(2)
    ENDIF
    SELE 1
  ENDIF
  SKIF
ENDDO
?
ENDIF
? 'BLOCK NO ', 'BLOCK NAME'+SPACE(25), 'STATUS-A', 'STATUS-B', 'STATUS-C'
IF BLK > 1
GO TOP
BLCK = 'COC'
DO WHILE .NOT. EOF()
  IF ELCK # BLCK
    ? SPACE(3)+BLOCK,SPACE(4)+BLCKNO,SPACE(3)+STATUS_A,SPACE(7)+STATUS_B,SPACE(7)+STATUS_C
    BLCK = BLOCK
  ENDIF
  SKIF
ENDDO
ELSE
GO TOP
? SPACE(3)+BLOCK,SPACE(3)+BLCKNO,SPACE(3)+STATUS_A,SPACE(7)+STATUS_B,SPACE(7)+STATUS_C
ENDIF
WAIT
ENDDO

```



```

*****
** This program is a query main program which controls the main menu
** of the commodity wise queries.
*****

```

```

FRAME = CHR(201)+CHR(205)+CHR(187)+CHR(186)+CHR(188)+CHR(205)+CHR(200)+CHR(186)+CHR(176)
FRAME1 = CHR(201)+CHR(205)+CHR(187)+CHR(186)+CHR(188)+CHR(205)+CHR(200)+CHR(186)
CS_DATE = '01/01/1985'
CLINE = REPLICATE('-',79)
SET PROC TO MICPROC
DO WHILE .T.
  CMDTY = SPACE(3)
  *** SELECT COMMODITY
  CLEAR
  DO SELECT1
  CFILEN = "MI"+CMDTY
  SELE 1
  USE &CFILEN
  INDEX ON SUBSTR(DEFNO,1,1) TO TEMP
  INDEX ON STATE+DIST TO TEMP1
  SELE 2
  USE MINFILE
  SELE 3
  USE STATFILE
  SET INDEX TO STDTFIL
  *** EXTRACT GRADES
  SELE 2
  SET FILTER TO MINERAL = CMDTY
  GO TOP
  CMIN = M_DES
  I = 0
  DO WHILE .NOT. EOF()
    I = I+1
    J = STR(I,2)
    IF I < 10
      J = STR(I,1)
    ENCLF
    CG&J = G_CCDE
    CD&J = G_DES
    CACC&J = ACDS
    SKIP
  ENDDO
  CGCTR = I+1
  SELE 1
  DO CASE
    CASE UNIT = ' '
      CUNITDES = SPACE(5)
    CASE UNIT = '1'
      CUNITDES = 'CARATS'
    CASE UNIT = '2'
      CUNITDES = 'KILOGRAMS'
    CASE UNIT = '3'
      CUNITDES = 'TONNES'
    CASE UNIT = '4'
      CUNITDES = "CCC'TONNES"
    CASE UNIT = '5'
      CUNITDES = "MILLION TONNES"
  ENDCASE
  STORE SPACE(1) TO QUERY
  DO WHILE QUERY # 'C' .OR. UPPER(QUERY) # 'E'
    QUERY = SPACE(1)
  CLEAR

```



```

DO MICGHEAD WITH 'COMMODITY WISE QUERY'
@ 6,25,16,52 BCX FRAME
@ 8,28 SAY "1. COMMODITY OCCURANCE"
@ 10,28 SAY "2. RESERVE POSITION      "
@ 12,28 SAY "3. DEFOSIT DETAILS      "
@ 14,28 SAY "C. EXIT                  "
@ 18,15 SAY "ENTER YOUR CHOICE(C,1,2,3/E,C,R,D/e,c,r,d)"
@ 18,60 GET QUERY PICTURE "!" VALID(QUERY$ "0123ECRDecrc")
READ
DO CASE
  CASE QUERY = 'C' .OR. UPFER(QUERY) = 'E'
    EXIT
  CASE QUERY = '1' .OR. LPFER(QUERY) = 'C'
    DO MICG1
  CASE QUERY = '2' .OR. UPFER(QUERY) = 'R'
    SELE 1
    SET FILTER TO
    GO TOP
    DO MICG2
  CASE QUERY = '3' .OR. LPFER(QUERY) = 'D'
    DO MICG3
ENDCASE
ENDDO
YN = 'Y'
@ 23,20 SAY "DO YCL WANT GLERY ON ANOTHER COMMODITY" GET YN PICTURE "!" VALID(YN$ "YNyn")
READ
IF LPFER(YN) = 'N'
  RETURN
ENDIF
ENDDC

```



```

*****
** This is the main controlling program of the system. This requires
** the procedure file named 'mainproc' active all the time.
*****

```

```

SET PROCEDURE TO MAINPROC
PUBLIC PW,NMI,SI,SJ,SK,FLAG,SECODE,STCODE,SMINDEX
PUBLIC SDES,STDES,SLDES,SFLAG1,SSP,SET,CCMM
STCODE = SPACE(2)
STORE SPACE(1) TO STDT,CCND,ANS1
PW = 'N'
STORE 0 TO NMI,NMI1
DC PASSWCRD
IF PW = 'N'
    RETURN
ENDIF
DC NIC
I = 1
DC WHILE I < 800
    I = I+1
ENDDC
DC PRESENTS
I = 1
DC WHILE I < 500
    I = I+1
ENDDC
DC WHILE NMI # 6
    DC MENL1
    SET PRCC TO MAINPROC
ENDDC
SET PROC TO
CLEAR

```



```

*****
** This program is a procedure file which contains the various small
** procedures which are use while generating the state wise report.
*****

```

PROCEDURE STATHEAD

```

* THIS PROCEDURE PRINTS THE HEADINGS FOR STATEWISE SUMMARY STATEMENT.
* CALLED BY PROGRAM STATWISE
? SPACE(54) + "MINERAL DATABASE" + SPACE(36) + "PAGE :","STCODE" - '+STR(SSP,3)
? SPACE(49) + "NATIONAL MINERAL INVENTORY"
? SPACE(47) + "STATE WISE - SUMMARY STATEMENT" + SPACE(28) + "AS ON :","SET
? "NAME OF STATE :","SDES
? "DISTRICT      :","SDES
? LINE
? SPACE(46) + "RESERVES AND RESOURCES"
? "MINERAL & COMMODITY" + SPACE(4) + "-----"
+ SPACE(2) + "GRADE"
? SPACE(31) + "PROVED      PROBABLE      POSSIBLE      TOTAL"
? LINE
? SMINDES,SPACE(9),"UNIT :","SDES
? LINE
SSP = SSP + 1
*****

```

PROCEDURE UNIT

```

CC CASE
CASE UNIT = ' '
SDES = " "
CASE UNIT = '1'
SDES = "CARATS"
CASE UNIT = '2'
SDES = "KILOGRAMS"
CASE UNIT = '3'
SDES = "TONNES"
CASE UNIT = '4'
SDES = "1000 TONNES"
CASE UNIT = '5'
SDES = "MILLION TONNES"
ENDCASE

```

PROCEDURE FILTER

```

CC CASE
CASE SKCUNT = 1
SET FILTER TO STATUS_A = 'F'
STATUS = 'FREEHOLD'
CASE SKCUNT = 2
SET FILTER TO STATUS_A = 'B'
STATUS = 'FREEHOLD & LEASEHOLD'
CASE SKCUNT = 3
SET FILTER TO STATUS_A = 'L'
STATUS = 'LEASEHOLD'
CASE SKCUNT = 4
SET FILTER TO STATUS_E = 'L'
STATUS = 'PUBLIC'
CASE SKCUNT = 5
SET FILTER TO STATUS_E = 'L' .AND. STATUS_C = 'C'
STATUS = 'PUBLIC CAPTIVE'
CASE SKCUNT = 6
SET FILTER TO STATUS_E = 'L' .AND. STATUS_C = 'N'
STATUS = 'PUBLIC NON-CAPTIVE'
CASE SKCUNT = 7
SET FILTER TO STATUS_E = 'L' .AND. STATUS_C = 'E'

```



```

STATUS = 'PUBLIC CAPTIVE & NON-CAPTIVE'
CASE SKCUNT = 8
SET FILTER TO STATUS_E = 'B'
STATUS = 'PUBLIC & PRIVATE'
CASE SKCUNT = 9
SET FILTER TO STATUS_E = 'B' .AND. STATUS_C = 'C'
STATUS = 'PUBLIC & PRIVATE CAPTIVE'
CASE SKCUNT = 10
SET FILTER TO STATUS_E = 'E' .AND. STATUS_C = 'N'
STATUS = 'PUBLIC & PRIVATE NON-CAPTIVE'
CASE SKCUNT = 11
SET FILTER TO STATUS_E = 'B' .AND. STATUS_C = 'E'
STATUS = 'PUBLIC & PRIVATE CAPTIVE & NON-CAPTIVE'
CASE SKCUNT = 12
SET FILTER TO STATUS_E = 'P'
STATUS = 'PRIVATE'
CASE SKCUNT = 13
SET FILTER TO STATUS_B = 'P' .AND. STATUS_C = 'C'
STATUS = 'PRIVATE CAPTIVE'
CASE SKCUNT = 14
SET FILTER TO STATUS_E = 'P' .AND. STATUS_C = 'N'
STATUS = 'PRIVATE NON-CAPTIVE'
CASE SKCUNT = 15
SET FILTER TO STATUS_E = 'P' .AND. STATUS_C = 'E'
STATUS = 'PRIVATE CAPTIVE & NON-CAPTIVE'
CASE SKCUNT = 16
SET FILTER TO STATUS_C = 'C'
STATUS = 'CAPTIVE'
CASE SKCUNT = 17
SET FILTER TO STATUS_C = 'B'
STATUS = 'CAPTIVE & NON-CAPTIVE'
CASE SKCUNT = 18
SET FILTER TO STATUS_C = 'N'
STATUS = 'NON-CAPTIVE'
ENDCASE
*****
PROCEDURE IGRADACC
IF IGRADE # SPACE(2)
DC CASE

```

```

CASE IGRADE = 'C1'
IFRCV1 = IPROV1 + IPRCVD
IFRCB1 = IPROE1 + IPRCEABLE
IFCSS1 = IPCSS1 + IPOSSIBLE
CASE IGRADE = 'C2'
IFRCV2 = IPROV2 + IPRCVD
IFRCB2 = IPROE2 + IPRCEABLE
IFCSS2 = IPCSS2 + IPOSSIBLE
CASE IGRADE = 'C3'
IFRCV3 = IPROV3 + IPRCVD
IFRCB3 = IPROE3 + IPRCEABLE
IFCSS3 = IPCSS3 + IPOSSIBLE
CASE IGRADE = 'C4'
IFRCV4 = IPROV4 + IPRCVD
IFRCB4 = IPROE4 + IPRCEABLE
IFCSS4 = IPCSS4 + IPOSSIBLE
CASE IGRADE = 'C5'
IFRCV5 = IPROV5 + IPRCVD
IFRCB5 = IPROE5 + IPRCEABLE
IFCSS5 = IPCSS5 + IPOSSIBLE
CASE IGRADE = 'C6'
IFRCV6 = IPROV6 + IPRCVD
IFRCB6 = IPROE6 + IPRCEABLE
IFCSS6 = IPCSS6 + IPOSSIBLE
CASE IGRADE = 'C7'

```



```

IFRCV7 = IFRCV7 + IFRCVED
IFRCB7 = IPRCE7 + IPRCEABLE
IFCSS7 = IFCSS7 + IFOSSIBLE
CASE IGRADE = 'C8'
IFRCV8 = IFRCV8 + IFRCVED
IFRCB8 = IPRCE8 + IPRCEABLE
IFCSS8 = IFCSS8 + IFOSSIBLE
CASE IGRADE = 'C9'
IFRCV9 = IFRCV9 + IFRCVED
IFRCB9 = IPRCE9 + IPRCEABLE
IFCSS9 = IFCSS9 + IFOSSIBLE
CASE IGRADE = '10'
IFRCV10 = IFRCV10 + IFRCVED
IFRCB10 = IPRCE10 + IPRCEABLE
IFCSS10 = IFCSS10 + IFOSSIBLE
CASE IGRADE = '11'
IFRCV11 = IFRCV11 + IFRCVED
IFRCB11 = IPRCE11 + IPRCEABLE
IFCSS11 = IFCSS11 + IFOSSIBLE
CASE IGRADE = '12'
IFRCV12 = IFRCV12 + IFRCVED
IFRCB12 = IPRCE12 + IPRCEABLE
IFCSS12 = IFCSS12 + IFOSSIBLE
CASE IGRADE = '13'
IFRCV13 = IFRCV13 + IFRCVED
IFRCB13 = IPRCE13 + IPRCEABLE
IFCSS13 = IFCSS13 + IFOSSIBLE
CASE IGRADE = '14'
IFRCV14 = IFRCV14 + IFRCVED
IFRCB14 = IPRCE14 + IPRCEABLE
IFCSS14 = IFCSS14 + IFOSSIBLE
CASE IGRADE = '15'
IFRCV15 = IFRCV15 + IFRCVED
IFRCB15 = IPRCE15 + IPRCEABLE
IFCSS15 = IFCSS15 + IFOSSIBLE
CASE IGRADE = '16'
IFRCV16 = IFRCV16 + IFRCVED
IFRCB16 = IPRCE16 + IPRCEABLE
IFCSS16 = IFCSS16 + IFOSSIBLE
CASE IGRADE = '17'
IFRCV17 = IFRCV17 + IFRCVED
IFRCB17 = IPRCE17 + IPRCEABLE
IFCSS17 = IFCSS17 + IFOSSIBLE
CASE IGRADE = '18'
IFRCV18 = IFRCV18 + IFRCVED
IFRCB18 = IPRCE18 + IPRCEABLE
IFCSS18 = IFCSS18 + IFOSSIBLE
CASE IGRADE = '19'
IFRCV19 = IFRCV19 + IFRCVED
IFRCB19 = IPRCE19 + IPRCEABLE
IFCSS19 = IFCSS19 + IFOSSIBLE
CASE IGRADE = '20'
IFRCV20 = IFRCV20 + IFRCVED
IFRCB20 = IPRCE20 + IPRCEABLE
IFCSS20 = IFCSS20 + IFOSSIBLE

```

ENDCASE

ENDIF

PROCEDURE RGRADADD

IF RGRADE # SPACE(2)

DO CASE

CASE RGRADE = 'C1'

RPROV1 = RPROV1 + RPROVED

RPRCB1 = RPROE1 + RPRCEABLE

RFCSS1 = RFOSS1 + RFOSSIBLE
 CASE RGRADE = 'C2'
 RFRV2 = RPROV2 + RPROVED
 RRCB2 = RRCB2 + RRCBABLE
 RFCSS2 = RFOSS2 + RFOSSIBLE
 CASE RGRADE = 'C3'
 RFRV3 = RPROV3 + RPROVED
 RRCB3 = RRCB3 + RRCBABLE
 RFCSS3 = RFOSS3 + RFOSSIBLE
 CASE RGRADE = 'C4'
 RFRV4 = RPROV4 + RPROVED
 RRCB4 = RRCB4 + RRCBABLE
 RFCSS4 = RFOSS4 + RFOSSIBLE
 CASE RGRADE = 'C5'
 RFRV5 = RPROV5 + RPROVED
 RRCB5 = RRCB5 + RRCBABLE
 RFCSS5 = RFOSS5 + RFOSSIBLE
 CASE RGRADE = 'C6'
 RFRV6 = RPROV6 + RPROVED
 RRCB6 = RRCB6 + RRCBABLE
 RFCSS6 = RFOSS6 + RFOSSIBLE
 CASE RGRADE = 'C7'
 RFRV7 = RPROV7 + RPROVED
 RRCB7 = RRCB7 + RRCBABLE
 RFCSS7 = RFOSS7 + RFOSSIBLE
 CASE RGRADE = 'C8'
 RFRV8 = RPROV8 + RPROVED
 RRCB8 = RRCB8 + RRCBABLE
 RFCSS8 = RFOSS8 + RFOSSIBLE
 CASE RGRADE = 'C9'
 RFRV9 = RPROV9 + RPROVED
 RRCB9 = RRCB9 + RRCBABLE
 RFCSS9 = RFOSS9 + RFOSSIBLE
 CASE RGRADE = '10'
 RFRV10 = RPROV10 + RPROVED
 RRCB10 = RRCB10 + RRCBABLE
 RFCSS10 = RFOSS10 + RFOSSIBLE
 CASE RGRADE = '11'
 RFRV11 = RPROV11 + RPROVED
 RRCB11 = RRCB11 + RRCBABLE
 RFCSS11 = RFOSS11 + RFOSSIBLE
 CASE RGRADE = '12'
 RFRV12 = RPROV12 + RPROVED
 RRCB12 = RRCB12 + RRCBABLE
 RFCSS12 = RFOSS12 + RFOSSIBLE
 CASE RGRADE = '13'
 RFRV13 = RPROV13 + RPROVED
 RRCB13 = RRCB13 + RRCBABLE
 RFCSS13 = RFOSS13 + RFOSSIBLE
 CASE RGRADE = '14'
 RFRV14 = RPROV14 + RPROVED
 RRCB14 = RRCB14 + RRCBABLE
 RFCSS14 = RFOSS14 + RFOSSIBLE
 CASE RGRADE = '15'
 RFRV15 = RPROV15 + RPROVED
 RRCB15 = RRCB15 + RRCBABLE
 RFCSS15 = RFOSS15 + RFOSSIBLE
 CASE RGRADE = '16'
 RFRV16 = RPROV16 + RPROVED
 RRCB16 = RRCB16 + RRCBABLE
 RFCSS16 = RFOSS16 + RFOSSIBLE
 CASE RGRADE = '17'
 RFRV17 = RPROV17 + RPROVED
 RRCB17 = RRCB17 + RRCBABLE


```

      RFCSS17 = RPOSS17 + RFCSSIELE
CASE RGRADE = '18'
      RFRVC18 = RPRCV18 + RFROVED
      RFRCB18 = RPRCB18 + RFROBAELE
      RFCSS18 = RPOSS18 + RFCSSIELE
CASE RGRADE = '19'
      RFRVC19 = RPRCV19 + RFROVED
      RFRCB19 = RPRCB19 + RFROBAELE
      RFCSS19 = RPOSS19 + RFCSSIELE
CASE RGRADE = '20'
      RFRVC20 = RPRCV20 + RFROVED
      RFRCB20 = RPRCB20 + RFROBAELE
      RFCSS20 = RPOSS20 + RFCSSIELE

```

ENDCASE

ENDIF

PROCEDURE ALLADD

```

IFROV = IFROV1+IFROV2+IFROV3+IFROV4+IFROV5+IFROV6+IFROV7+IFROV8+IFROV9+IFROV10
IFROV = IFROV+IFROV11+IFROV12+IFROV13+IFROV14+IFROV15+IFROV16+IFROV17+IFROV18+IFROV19+IFROV20
IFROB = IFROB1+IFROB2+IFROB3+IFROB4+IFROB5+IFROB6+IFROB7+IFROB8+IFROB9+IFROB10
IFROB = IFROB+IFROB11+IFROB12+IFROB13+IFROB14+IFROB15+IFROB16+IFROB17+IFROB18+IFROB19+IFROB20
IFCSS = IFCSS1+IFCSS2+IFCSS3+IPOSS4+IFCSS5+IFCSS6+IFCSS7+IPOSS8+IPOSS9+IFCSS10
IFCSS = IFCSS+IFCSS11+IFCSS12+IFCSS13+IFCSS14+IFCSS15+IFCSS16+IFCSS17+IFCSS18+IFCSS19+IFCSS20
RFRVC = RFRVC1+RFRVC2+RFRVC3+RFRVC4+RFRVC5+RFRVC6+RFRVC7+RFRVC8+RFRVC9+RFRVC10
RFRVC = RFRVC+RFRVC11+RFRVC12+RFRVC13+RFRVC14+RFRVC15+RFRVC16+RFRVC17+RFRVC18+RFRVC19+RFRVC20
RFRCB = RFRCB1+RFRCB2+RFRCB3+RFRCB4+RFRCB5+RFRCB6+RFRCB7+RFRCB8+RFRCB9+RFRCB10
RFRCB = RFRCB+RFRCB11+RFRCB12+RFRCB13+RFRCB14+RFRCB15+RFRCB16+RFRCB17+RFRCB18+RFRCB19+RFRCB20
RPOSS = RFCSS1+RFCSS2+RPCSS3+RPOSS4+RFCSS5+RFCSS6+RPCSS7+RPOSS8+RPOSS9+RFCSS10
RPOSS = RFCSS+RPCSS11+RPCSS12+RPCSS13+RPCSS14+RPCSS15+RPCSS16+RPCSS17+RPCSS18+RPCSS19+RPCSS20

```

PROCEDURE STGHEAD

PARAMETERS SLEHEC

CLEAR

? SPACE(32)+'MINERAL DATABASE'+SPACE(21)

? SPACE(27)+'NATIONAL MINERAL INVENTORY'

F1 = LEN(SLBHED)

F2 = INT(80-F1)/2

? SPACE(F2)+SLBHED

? SPACE(F2)+REPLICATE(CHR(205),F1)

RETURN

```

*****
** This program is a query program which controls the main menu for
** the queries of all types.
*****

```

```

SET PROC TO PROCEDURE
STORE 0 TO NMI1
CLEAR
VV = REPLICATE(CHR(176),27)
DO STGHEAD WITH "QUERY MAIN MENU"
@ 1,69 SAY SPACE(10)
@ 7,25 TO 15,53 DOUBLE
I = 8
DO WHILE I < 15
    @ 11,26 SAY VV
    I = I+1
ENDDO
@ 9,29 SAY "1. MINERAL WISE QUREY"
@ 11,29 SAY "2. STATE WISE QUREY  "
@ 13,29 SAY "C. EXIT              "
@ 17,23 SAY "ENTER YOUR CHOICE (C,1,2/E,N,S/e,n,s)"
ANS = SPACE(1)
DO WHILE ANS = SPACE(1)
    @ 17,61 GET ANS
    READ
    DO CASE
        CASE UPPR(ANS) = 'M' .CR. ANS = '1'
            DO MICQUERY
        CASE UPPR(ANS) = 'S' .CR. ANS = '2'
            DO WHILE NMI1 # 2
                DO GMENL2
            ENDDO
        CASE UPPR(ANS) = 'E' .CR. ANS = 'C'
            NMI = 1
            EXIT
        OTHERWISE
            ANS = SPACE(1)
    ENDCASE
ENDDO
SET PROC TO MAINPROC

```



```

*****
** This program is a query program which controls the query main menu for
** the state wise queries.
*****

```

```

CLEAR
STORE 0 TO NMI1
VV = REPLICATE(CHR(176),39)
DO STGHEAD WITH "STATE WISE QUERY"
@ 7,20 TO 17,60 DCUBLE
I = 8
DO WHILE I < 17
  @ I,21 SAY VV
  I = I+1
ENDDO
@ 9,24 SAY "1. COMMODITY OCCURANCE"
@ 11,24 SAY "2. RESERVE POSITION OF A COMMODITY"
@ 13,24 SAY "3. RESERVE POSITION OF A GRADE"
@ 15,24 SAY "C. EXIT"
@ 20,18 SAY "ENTER YOUR CHOICE (C,1,2,3/E,C,R,G/e,c,r,g)"
STORE SPACE(1) TO ANS,ANS1
DO WHILE ANS = SPACE(1)
  @ 20,62 GET ANS
  READ
  DO CASE
    CASE UPPER(ANS) = 'C' .CR. ANS = '1'
      DO WHILE NMI1 # 3
        ANS1 = "C"
        DO GMENU3
      ENDDO
    CASE UPPER(ANS) = 'R' .CR. ANS = '2'
      DO WHILE NMI1 # 3
        ANS1 = "R"
        DO GMENU3
      ENDDO
    CASE UPPER(ANS) = 'G' .CR. ANS = '3'
      DO WHILE NMI1 # 3
        ANS1 = "G"
        DO GMENU3
      ENDDO
    CASE UPPER(ANS) = 'E' .CR. ANS = 'C'
      NMI1 = 2
      EXIT
    OTHERWISE
      ANS = SPACE(1)
  ENDCASE
ENDDO

```



```

*****
** This program is a query program which controls the query sub menu
** for the statwise queries.
*****

```

```

CLEAR
STORE C TO NM1
VV = REPLICATE(CHR(176),23)
DO STGHEAD WITH "STATE WISE QUERY"
@ 1,69 SAY SPACE(10)
@ 7,27 TO 15,51 DOUBLE
I = 8
DO WHILE I < 15
  @ 11,28 SAY VV
  I = I+1
ENDDO
@ 9,32 SAY "1. IN A STATE      "
@ 11,32 SAY "2. IN A DISTRICT"
@ 13,32 SAY "C. EXIT          "
@ 17,22 SAY "ENTER YOUR CHOICE (C,1,2/E,S,D/e,s,d)"
ANS = SPACE(1)
DO WHILE ANS = SPACE(1)
  @ 17,60 GET ANS
  READ
  DO CASE
    CASE UPPER(ANS) = 'S' .CR. ANS = '1'
      DO STATQUERY
    CASE UPPER(ANS) = 'D' .CR. ANS = '2'
      DO DISTQUERY
    CASE UPPER(ANS) = 'E' .CR. ANS = 'C'
      NM1 = 3
      EXIT
    OTHERWISE
      ANS = SPACE(1)
  ENDCASE
ENDDO

```



```

*****
** This program is a query subprogram for getting the reserve position
** of a commodity at state or district level in the state wise queries.
*****

```

```

MIN = 'MI'+CCMM
SELE 1
USE 8MIN
SET INDEX TO 8MIN
SEEK STD1
STOR SPACE(1) TO GCD1,GCD2,GCD3,GCD4
SELE 2
GO TOF
SEEK COMM
MCES = M_CES
I = 1
DO WHILE MINERAL = CCMM .AND. .NOT. ECF()
  IF ADDS = 'Y'
    II = STR(I,1)
    GCD&II = G_CCDE
    I = I+1
  ENDIF
  SKIP
ENDDO
SELE 1
DO UNIT
STOR C TO IPRCV,IPRCB,IPOSS,RFCRV,RPRCB,RFOSS,PMFCRV,FMPCRB
STOR C TO FMFCSS,SMFCRV,SMPCRB,SMFOSS,PRCS,PROG,CTOTAL
DO WHILE 8CONC .AND. .NOT. ECF()
  IF 8SINTR = '1'
    IF IGRADE # GCD1 .AND. IGRADE # GCD2 .AND. IGRADE # GCD3;
      .AND. IGRADE # GCD4
      IPRCV = IPRCV + IMPROVED
      IPRCB = IPRCB + IMPRCEABLE
      IPOSS = IPOSS + IMPOSSIBLE
    ENDIF
    IF RGRADE # GCD1 .AND. RGRADE # GCD2 .AND. RGRADE # GCD3;
      .AND. RGRADE # GCD4
      RPRCV = RPRCV + RPROVED
      RPRCB = RPRCB + RPRCEABLE
      RFOSS = RFOSS + RPOSSIBLE
    ENDIF
    PMFCRV = PMFCRV + FMPCRV
    PMFCRB = PMFCRB + FMPCRB
    PMFCSS = PMFCSS + FMPCSS
    SMFCRV = SMFCRV + SMPCRV
    SMFCRB = SMFCRB + SMPCRB
    SMFCSS = SMFCSS + SMPCSS
    PRCS = PRCS + PROSGTY
    PROG = PROG + PROGGTY
  ENDIF
  SKIP
ENDDO
CTOTAL = IPRCV+IPROB+IPOSS+RFCRV+RPRCB+RFCSS+FMFCRV+FMPCRB
CTOTAL = CTOTAL+FMPCSS+SMFCRV+SMFCRB+SMPCSS+PRCS+PROG
& 5,C CLEAR
& 5,C SAY "STATE : " + STDES
& 6,C SAY "COMMODITY : " + MCES
& 7,C TO 7,78 DOLELE
& 3,61 SAY "AS ON : 01/01/1985"
IF CTOTAL = 0
  @ 12,10 SAY "THIS COMMODITY HAS NOT BEEN ESTIMATED IN THIS STATE BY NCW"

```



```

ELSE
6 6,58 SAY "UNIT : " + SLDES
6 8,C SAY SPACE(24)+"PROVED          PRCEABLE          POSSIBLE"+SPACE(14)+"TOTAL"
6 9,C TO 9,78 DOLELE
IF (IFROV+IPRCB+IPOSS) # C
? "INSITU RESERVES"
? SPACE(15), STR(IFROV,14,2), STR(IPRCB,14,2), STR(IPOSS,14,2)
?? SPACE(1) + STR(IPRCV+IPRCB+IPOSS,18,2)
ENDIF
IF (RFROV+RPRCB+RPOSS) # C
? "RECOVERABLE RESERVES"
? SPACE(15), STR(RFROV,14,2), STR(RPRCB,14,2), STR(RPOSS,14,2)
?? SPACE(1) + STR(RPRCV+RPRCB+RPOSS,18,2)
ENDIF
IF (FMPCV+PMFROB+PMFOSS) # C
? "CONDITIONAL (SLEMARGINAL) RESOURCES"
? SPACE(15), STR(FMPCV,14,2), STR(PMFROB,14,2), STR(PMFOSS,14,2)
?? SPACE(1) + STR(FMPCV+PMFROB+PMFOSS,18,2)
ENDIF
IF (SMPCV+SMFROB+SMFOSS) # C
? "CONDITIONAL (PARAMARGINAL) RESOURCES"
? SPACE(15), STR(SMPCV,14,2), STR(SMFROB,14,2), STR(SMFOSS,14,2)
?? SPACE(1) + STR(SMPCV+SMFROB+SMFOSS,18,2)
ENDIF
IF PROS # C
? "PROSPECTIVE RESOURCES"
? SPACE(60), STR(PROC,18,2)
ENDIF
IF PROG # C
? "PROGNOSTIC RESOURCES"
? SPACE(60), STR(PROG,18,2)
ENDIF
? REPLICATE(CHR(205),79)
ENDIF
6 23,75 SAY ""
WAIT "          PRESS ANY KEY TO CONTINUE"

```



```

*****
** This program is a query subprogram for getting the reserve position
** of a grade at state or district level in the state wise queries.
*****

6 1,C SAY "CCMMOITY : " + CCMM
6 5,C CLEAR
6 5,C SAY "STATE : " + STDES
6 6,C SAY "GRADE : " + GDES
MIN = 'MI'+CCMM
SELE 1
LSE 8MIN
SET INDEX TO 8MIN
SEEK STD1
IF ECF()
    6 12,1C SAY "THIS GRADE IS NOT PRESENT IN THIS STATE / DISTRICT"
    RETURN
ENDIF
CC UNIT
STOR C TO IPRCV,IPRCB,IPCSS,RPROV,RPRCB,RPOSS,CTOTAL
CC WHILE 8CCND .AND. .NOT. ECF()
    IF 8SINTR = '1'
        IF IGRADE = GCODE
            IPRCV = IPRCV + IPRCVD
            IPRCB = IPRCB + IPRCBEABLE
            IPOSS = IPOSS + IPOSSIBLE
        ENDIF
        IF RGRADE = GCODE
            RPRCV = RPRCV + RPRCVD
            RPRCB = RPRCB + RPRCBEABLE
            RPOSS = RPOSS + RPOSSIBLE
        ENDIF
    ENDIF
    SKIP
ENDIF
CTOTAL = IPRCV+IPRCB+IPOSS+RPROV+RPRCB+RPOSS
6 7,C TO 7,78 DOBLE
6 3,61 SAY "AS ON : 01/01/1985"
IF CTOTAL = 0
    6 12,7 SAY "THIS GRADE HAS NOT BEEN ESTIMATED IN THIS STATE/DISTRICT BY NOW"
ELSE
    6 6,58 SAY "UNIT : " + SLDES
    6 8,C SAY SPACE(24)+"PROVED          PRCEABLE          FOSSIBLE"+SPACE(14)+"TOTAL"
    6 9,C TO 9,78 DOBLE
    IF (IFROV+IPRCB+IPOSS) # 0
        ? "INSITU RESERVES"
        ? SPACE(15), STR(IFROV,14,2), STR(IPRCB,14,2), STR(IPCSS,14,2)
        ?? SPACE(1) + STR(IPRCV+IPRCB+IPOSS,18,2)
    ENDIF
    IF (RPROV+RPRCB+RPOSS) # 0
        ? "RECCVERABLE RESERVES"
        ? SPACE(15), STR(RPROV,14,2), STR(RPRCB,14,2), STR(RPCSS,14,2)
        ?? SPACE(1) + STR(RPRCV+RPRCB+RPOSS,18,2)
    ENDIF
    ? REFLICATE(CHR(205),79)
ENDIF
6 23,75 SAY ""
WAIT "
6 2,66 SAY SPACE(13)
        FRESS ANY KEY TO CONTINUE"

```



```
*****
** This program creates the pre requisite database named 'scata'
** which keeps only the codes of all the states.
*****
```

```
USE SCATA
ZAP
APPEND FROM STATFILE
GO TOP
DO WHILE .NOT. EOF()
    SCD = STATE
    SKIP
    IF .NOT. EOF()
        DO WHILE STATE = SCD
            DELETE
            SKIP
        ENDDO
    ENDIF
ENDDO
PACK
USE STATFILE
INDEX ON STATE TO STATFILE
USE
```



```

*****
** This program is a query subprogram for selection of a commodity
** from the list of commodities in a state in the state wise queries.
*****

```

```

FLAG = 0
SELE 1
USE STRECCRD
SET INDEX TO STREC2
SELE 2
USE MINFILE
SET INDEX TO MINFILE
SELE 1
SET RELATION TO MINERAL INTO MINFILE
GO TOP
SEEK STCCDE
@ 5,12 SAY "COMMODITY SELECTION :"
@ 6,12 TO 22,65 DCUBLE
IF ECF()
  @ 12,21 SAY "THERE IS NO MINERAL / COMMODITY PRESENT"
  @ 14,33 SAY "IN THIS STATE"
  J = 1
  @ 23,75 SAY ""
  WAIT "                                PRESS ANY KEY TO CONTINUE"
  RETURN
ENDIF
@ 7,17 SAY "CODE" + SPACE(2) + "COMMODITY NAME"
@ 8,17 SAY "----" + SPACE(2) + "-----"
I = 9
DO WHILE STATE = STCCDE .AND. .NOT. ECF()
  MDES = B->M_DES
  MCCDE = MINERAL
  IF I > 21
    @ 23,10 SAY SPACE(1)
    WAIT "                                PRESS 'E' TO ENTER CCDE OR ANY OTHER KEY TO CONTINUE" TO ANS
    @ 24,0 CLEAR
    IF UPPER(ANS) = 'E'
      EXIT
    ENDIF
    @ 9,15 CLEAR TO 21,64
    I = 9
  ENDIF
  @ 1,17 SAY MCCDE + SPACE(3) + MDES
  I = I+1
  SKIP
ENDDO
SELE 1
SET INDEX TO STREC1
SET FILTER TO STATE = STCCDE
GO TOP
DO WHILE FLAG = 0
  ANS = SPACE(3)
  @ 24,0 CLEAR
  @ 24,28 SAY "ENTER COMMODITY CCDE"
  @ 24,49 GET ANS
  READ
  COMM = UPPER(ANS)
  SELE 1
  SEEK COMM
  IF ECF()
    @ 24,0 CLEAR
    @ 23,75 SAY ""

```


WAIT '
ELSE
EXIT
ENDIF
ENDDC

ENTERED WRONG CCDE, SO PRESS ANY KEY TO ENTER THE CODE AGAIN"

 ** This program is a query program for getting the occurrence of
 ** commodities in a state in state wise query.

```

DC STSLECT1
@ 1,69 SAY "STATE : " + STCODE
@ 5,C CLEAR
SELE 1
LSE STRECORD
SET INDEX TO STREC2
SELE 2
LSE MINFILE
SET INDEX TO MINFILE
SELE 1
SET RELATION TO MINERAL INTO MINFILE
GO TOP
SEEK STCODE
@ 5,C SAY "STATE : " + STDES + SPACE(28) + "AS ON : 01/01/1985"
@ 6,C TO 6,75 DOUBLE
IF EOF()
  @ 12,10 SAY "THERE IS NO MINERAL / COMMODITY PRESENT IN THIS STATE"
ELSE
  @ 7,C SAY "THE FOLLOWING COMMODITIES ARE PRESENT IN THIS STATE : "
  @ 8,12 TO 22,65 DOUBLE
  I = 9
  J = 1
  DO WHILE STATE = STCODE
    MDES = E->M_DES
    IF I > 21
      @ 23,75 SAY ""
      WAIT "                                PRESS ANY KEY TO CONTINUE"
      @ 24,C CLEAR
      @ 9,13 CLEAR TO 21,64
      I = 9
    ENCIF
    @ 1,16 SAY STR(J,2) + SPACE(2) + MDES
    I = I+1
    J = J+1
    SKIP
  ENDDO
ENDIF
@ 23,75 SAY ""
WAIT "                                PRESS ANY KEY TO CONTINUE"
CLOSE DATA
  
```



```
*****  
** This program is a query program for getting the reserve position  
** of a commodity in at state level in the state wise queries.  
*****
```

```
STORE 0 TO FLAG,J  
SINTR = 'IS'  
CC STSLECT1  
G 1,69 SAY "STATE : " + STCCDE  
G 5,C CLEAR  
CC STATCCM  
IF J # 1  
    STCT = STCCDE  
    CCND = "STATE = STCCDE"  
    CC RESFOS  
ENDIF
```



```
*****
** This program is a query program for getting the reserve position
** of a commodity in a district in the state wise query.
*****
```

```
STORE 0 TO J,FLAG
SINTR = 'ID'
DC STSLECT1
DC DTSLECT
6 5,C CLEAR
DC DISTCCM
IF J # 1
    STCT = STCCDE+SDCCDE
    CCND = "STATE = STCCDE .AND. DIST = SECCDE"
    DC RESFCS
ENDIF
```



```
*****
** This program is a query subprogram for getting the reserve position
** of a grade at state level in the state wise queries.
*****
```

```
STORE C TO FLAG,J
SINTR = 'IS'
DC STSLECT1
6 5,C CLEAR
DC STATCCM
IF J # 1
  STORE SPACE(2) TO GCODE,GDES
  DC GRESLECT
  STDT = STCCDE
  CCND = "STATE = STCCDE"
  DC RESPCSGW
ENDIF
```

 ** This program is a query program for getting the occurrence of
 ** commodities in a district in the state wise queries.

```

DO STSLECT1
DO DTSLECT
SELE 1
USE STRECORD
SET INDEX TO STREC2
SELE 3
USE MINFILE
SET INDEX TO MINFILE
SELE 1
SEEK STCCDE
6 5,C CLEAR
6 5,C SAY "DISTRICT : " + SDES + SPACE(28) + "AS ON : 01/01/1985"
6 6,C TO 6,79 DOUBLE
6 7,C SAY "IN THIS DISTRICT THE FOLLOWING COMMODITIES ARE PRESENT : "
6 8,12 TO 22,65 DOUBLE
I = 9
J = 1
DO WHILE STATE = STCCDE .AND. .NOT. ECF()
  MIN = 'MI'+MINERAL
  SELE 2
  USE 8MIN
  SET INDEX TO 8MIN
  STATDIST = STCCDE + SECCDE
  SEEK STATDIST
  IF .NOT. ECF()
    SET RELATION TO MINERAL INTO MINFILE
    MDES = C->M_DES
    MCCDE = MINERAL
    IF I > 21
      6 23,75 SAY ""
      WAIT "                                PRESS ANY KEY TO CONTINUE"
      6 24,C CLEAR
      6 9,13 CLEAR TO 21,64
      I = 9
    ENDIF
    6 1,16 SAY STR(J,2) + SPACE(2) + MDES
    I = I+1
    J = J+1
  ENDIF
  SELE 1
  SKIP
ENDDO
IF J = 1
  6 12,21 SAY "THERE IS NO MINERAL / COMMODITY PRESENT"
  6 14,33 SAY "IN THIS DISTRICT"
ENDIF
6 23,75 SAY ""
WAIT "                                PRESS ANY KEY TO CONTINUE"
CLOSE DATA

```



```
*****
** This program is a query program for getting the reserve position
** of a grade in the state wise query. This is used in the query main
** program named 'ameru1'.
*****
```

```
STORE 0 TO FLAG,J
SINTR = 'ID'
DC STSLECT1
& 5,C CLEAR
DC DTSLECT
DC DISTCCM
IF J # 1
  STORE SPACE(2) TO GCODE,GDES
  DC GRESLECT
  STLT = STCCDE+SDCCDE
  COND = "STATE = STCCDE .AND. DIST = SDCCDE"
  DC RESFCSEW
ENDIF
```



```
*****
** This program is a query program for getting the state wise queries
** at state level. This is used in the query main program named 'qmeru1'.
*****
```

```
CC CASE
CASE ANS1 = "C"
    CC STATQ1
CASE ANS1 = "R"
    CC STATQ2
CASE ANS1 = "G"
    CC STATQ4
ENDCASE
```



```

*****
** This is the main program for generating the state wise report.
** this uses the subprogram named 'abcd' and the various procedures
** from the procedure file named 'procedur'.
*****

```

```

CLOSE DATA
SET PROCEDURE TO PROCEDUR
STORE 0 TO SI, SK, FLAG
SSP = 1
SDT = '01/01/1985'
SDDDES = 'ALL DISTRICTS'
STDES = SPACE(8)
LINE = REPLICATE('-',123)
CLEAR
DO STSLECT
* STATE CCDE = STCCDE *
TXTFILE = "S"+STCCDE
SET ALTERNATE TO &TXTFILE
SET ALTERNATE ON
SELE 1
USE MINFILE1
SELE 2
USE STINDEX
SELE 4
USE STRECCRD
SET INDEX TO STREC2
SELE 5
USE TEMP
ZAP
SELE 6
USE INTRFILE
DO ABCD
SK = 1
  SELE 2
  GC TOP
  SET FILTER TO STATE = STCCDE
  GC TOP
  COPY TO TEMP3
  SET FILTER TO
  SELE 8
  USE TEMP3
  DO WHILE .NOT. EOF()
    SDDCDE = DIST
    SDDDES = D_DES
    DO ABCD
    SELE 8
    SKIP
  ENDDO
SET ALTERNATE OFF
SET ALTERNATE TO
SET PROC TO MAINPROC

```



```

*****
** This program creates the pre requisite database named 'stctfile'
** which keeps the correspondence between the states and the commodities.
*****

* THIS PROGRAM KEEPS THE RECORD OF COMMODITIES IN EACH STATE
SELE 1
USE CDATA
SELE 2
USE STRECORD
ZAP
SELE 1
DEPCTR = 1
DO WHILE .NOT. EOF()
  MIN = MINERAL
  MIN1 = 'MI'+MIN
  STCRE C TO TOTAL1,TOTAL2,TCTAL3,TCTAL4,TOTAL5,TOTAL6,TCTAL7,TCTAL8
  SELE 3
  USE &MIN1
  INDEX ON STATE + DIST TO &MIN1
  SET INDEX TO &MIN1
  GO TOP
  DO WHILE .NOT. EOF()
    TOTAL1 = TCTAL1 + IMPROVED
    TOTAL2 = TCTAL2 + IMPROVEABLE
    TOTAL3 = TCTAL3 + IMPOSSIBLE
    TOTAL4 = TCTAL5 + IMPROVED
    TOTAL5 = TCTAL6 + IMPROVEABLE
    TOTAL6 = TCTAL7 + IMPOSSIBLE
    SCD = STATE
    DEF = DEPOSIT
    SELE 2
    APPEND BLANK
    REPL STATE WITH SCD
    REPL MINERAL WITH MIN
  SELE 3
  SKIP
  IF .NOT. EOF()
    IF DEPOSIT # DEF
      DEPCTR = DEPCTR + 1
      DEF = DEPOSIT
    ENDIF
    IF STATE = SCD
      DO WHILE STATE = SCD
        IF DEPOSIT # DEF
          DEPCTR = DEPCTR + 1
          DEF = DEPOSIT
        ENDIF
        TOTAL1 = TCTAL1 + IMPROVED
        TOTAL2 = TCTAL2 + IMPROVEABLE
        TOTAL3 = TCTAL3 + IMPOSSIBLE
        TOTAL4 = TCTAL5 + IMPROVED
        TOTAL5 = TCTAL6 + IMPROVEABLE
        TOTAL6 = TCTAL7 + IMPOSSIBLE
        SKIP
      ENDDO
    ENDIF
  ENDIF
ENDIF
TOTAL7 = TCTAL1 + TCTAL2 + TCTAL3
TOTAL8 = TCTAL5 + TCTAL6 + TCTAL7
SAVE TO &MIN ALL LIKE TOTAL*

```


SAVE TO DEFCTR ALL LIKE DEFCTR
SELE 1
SKIP

ENDCC
SELE 2
INDEX ON MINERAL TO STREC1
INDEX ON STATE TO STREC2


```

*****
** This program is program for selection of a state from the screen
** with statecodes & their names with the help of curser movement
*****

```

```

CLEAR
INK = 0
SELE 1
USE SCATA
SELE 2
USE STATFILE
SET INDEX TO STATFILE
SELE 1
SET RELATION TO STATE INTO STATFILE
SKIP
I = 11
DO WHILE .NOT. EOF()
    II=STR(I,2)
    CCDE&II=STATE
    DES&II=B->S_DES
    I = I+1
    SKIP
ENDDO
DO STATE.PRG
I=11
J=12
G=16
DO WHILE I < 27
    IF J=12
        II=STR(I,2)
    ELSE
        II=STR(I+16,2)
    ENDF
    @ I-6,J GET CCDE&II
    @ I-6,G GET DES&II
    CLEAR GETS
    INK = INKEY()
    DO CASE
        CASE INK = 4
            @ I-6,J SAY CCDE&II
            @ I-6,G SAY DES&II
            IF J < 13
                J=J+31
                G=G+31
            ENDF
        CASE INK = 19
            @ I-6,J SAY CCDE&II
            @ I-6,G SAY DES&II
            IF J > 13
                J=J-31
                G=G-31
            ENDF
        CASE INK = 5
            @ I-6,J SAY CCDE&II
            @ I-6,G SAY DES&II
            IF I > 11
                I=I-1
            ELSE
                I=26
            ENDF
        CASE INK = 24
            @ I-6,J SAY CCDE&II

```



```
  @ 1-6,G SAY DES&II
  IF I < 26
    I=I+1
  ELSE
    I=11
  ENCIF
CASE INK = 18
  @ 1-6,J SAY CCDE&II
  @ 1-6,G SAY DES&II
  I=11
CASE INK = 3
  @ 1-6,J SAY CCDE&II
  @ 1-6,G SAY DES&II
  I=26
CASE INK = 13
  EXIT
ENDCASE
ENDDC
STCODE = CCDE&II
STDES = DES&II
```



```

*****
** This program is for the selection of a commodity from the list of
** commodity codes and their names at various levels.
*****

```

```

SET TALK OFF
SET SAFETY OFF
SET PROC TO PROCEDURE
FLAG = 0
CLEAR
SELE 1
USE CCATA
SELE 2
USE MINFILE
SET INDEX TO MINFILE
SELE 1
SET RELATION TO MINERAL INTO MINFILE
DO STGHEAD WITH "DEPARTMENT OF MINES"
@ 1,70 SAY SPACE(7)
@ 5,14 SAY "COMMODITY SELECTION"
@ 6,14 TO 22,65 DOUBLE
@ 7,17 SAY "CODE" + SPACE(2) + "COMMODITY NAME"
@ 8,17 SAY "----" + SPACE(2) + "-----"
I = 9
DO WHILE .NOT. EOF()
    MDES = E->M_DES
    MCCDE = MINERAL
    IF I > 21
        @ 23,10 SAY SPACE(1)
        WAIT "          PRESS 'E' TO ENTER CODE OR ANY OTHER KEY TO CONTINUE" TO ANS
        IF UPPER(ANS) = "E"
            EXIT
        ENDIF
        @ 9,15 CLEAR TO 21,64
        I = 9
    ENDIF
    @ 1,17 SAY MCCDE + SPACE(3) + MDES
    I = I+1
    SKIP
ENDDO
ENDIF
DO WHILE FLAG = 0
    ANS = SPACE(3)
    @ 24,C CLEAR
    @ 24,28 SAY "ENTER COMMODITY CODE"
    @ 24,49 GET ANS
    READ
    COMM = UPPER(ANS)
    SELE 2
    SEEK COMM
    IF EOF()
        @ 24,C CLEAR
        @ 23,75 SAY ""
        WAIT "          ENTERED WRONG CODE, SO PRESS ANY KEY TO ENTER THE CODE AGAIN"
    ELSE
        EXIT
    ENDIF
ENDDO

```



```

*****
** This program is for the validation of the data. This uses different
** procedures from the procedure file named 'valproc'.
*****

```

```

CLOSE DATA
SET PROC TO VALPRCC
DO CCMSLECT
INPUT1 = "MI" + CCMM
CLEAR
@ 10,10 SAY "VALIDATION OF " + CCMM + " FILE IS GOING ON, SO WAIT FOR LISTING"
SELECT 1
USE MINFILE1
SELECT 2
USE STATFILE
SELECT 3
USE 8INPUT1
SELECT 4
USE ERRORS
ZAP
APPEND BLANK
STORE SPACE(2) TO G1,G2,G3,G4,G5,G6,G7,G8,G9,
,G10,G11,G12,G13,G14,G15,G16,G17
PUBLIC LCTR,FLAGR,FLAGS,SUNIT,TEMP
STORE C TO LCTR,FLAGR,FLAGS,KNT,FLG
SELECT 1
DO TOP
SET FILTER TO MINERAL=COMM
DO TOP
MIN = M_DES
I=1
DO WHILE .NOT. ECF()
  IF I<10
    II = STR(I,1)
  ELSE
    II = STR(I,2)
  ENDOF
  G8II=G_CODE
  I = I+1
  SKIP
ENDDO
SET FILTER TO
SELECT 3
DO TOP
DO UNITREFL
DO WHILE .NOT. ECF()
  FLAGB = 'N'
  Y='C1'
  IF MINERAL # COMM
    DO ERROR
  ENDOF
  IF VAL(DEPOSIT) <= C
    Y='C2'
    DO ERROR
  ENDOF
  IF VAL(IS) <= C
    Y='C3'
    DO ERROR
  ENDOF
  IF VAL(ID) <= C
    Y='C4'
    DO ERROR

```



```

ENDIF
IF VAL(IS) > VAL(IC)
  Y='C5'
  CC ERROR
ENDIF
IF VAL(STATE) < 10 .AND. STATE # SPACE(2)
  X = 'STATE'
  Y = STATE
  CC CORRECT WITH X,Y
ENDIF
IF VAL(DIST) < 10 .AND. DIST # SPACE(2)
  X = 'DIST'
  Y = DIST
  CC CORRECT WITH X,Y
ENDIF
SCC=STATE
DCC=DIST
SELECT 2
  CC TCP
  SET FILTER TO STATE=SCC
  CC TCP
  IF ECF()
    Y='06'
    CC ERROR
    Y='07'
    CC ERROR
  ELSE
    LOCATE FOR DIST=DCC
    IF ECF()
      Y='07'
      CC ERROR
    ENDIF
    SELECT 2
    SET FILTER TO
    SELECT 3
  ENDIF
IF STATUS_A # "F" .AND. STATUS_A # "L" .AND. STATUS_A # "B"
  Y='C8'
  CC ERROR
ENDIF
IF STATUS_B # "P" .AND. STATUS_E # "U" .AND. STATUS_B # "B" ;
  .AND. STATUS_B # " "
  Y='C9'
  CC ERROR
ENDIF
IF STATUS_C # "C" .AND. STATUS_C # "N" .AND. STATUS_C # "B" ;
  .AND. STATUS_C # " "
  Y='10'
  CC ERROR
ENDIF
STORE C TO FLAG1, FLAG2, FLG
IF IGRADE # SPACE(2) .OR. RGRADE # SPACE(2)
  IF IGRADE # SPACE(2)
    IF VAL(IGRADE) < 10
      X = 'IGRADE'
      Y = IGRADE
      CC CORRECT WITH X,Y
    ENDIF
  ENDIF
  IF RGRADE # SPACE(2)
    IF VAL(RGRADE) < 10
      X = 'RGRADE'
      Y = RGRADE
      CC CORRECT WITH X,Y
    ENDIF
  ENDIF

```



```

    ENDIF
ENDIF
I=1
DO WHILE I < 18
    IF I < 10
        II = STR(I,1)
    ELSE
        II = STR(I,2)
    ENDIF
    IF G&II # SPACE(2)
        IF G&II=IGRADE
            FLAG1=1
        ENDIF
        IF G&II=RGRADE
            FLAG2=1
        ENDIF
        IF FLAG1=1 .AND. FLAG2=1
            EXIT
        ELSE
            I = I+1
        ENDIF
    ELSE
        EXIT
    ENDIF
ENDDO
ENDIF
IF IGRADE # SPACE(2)
    IF FLAG1=0
        Y='11'
        DO ERROR
    ENDIF
ENDIF
IF RGRADE # SPACE(2)
    IF FLAG2=0
        Y='12'
        DO ERROR
    ENDIF
ENDIF
ENDIF
Y = '13'
XX = IMPROVED
DO VALCHECK WITH XX
Y = '14'
XX= IMPROBABLE
DO VALCHECK WITH XX
Y = '15'
XX=IMPOSSIBLE
DO VALCHECK WITH XX
Y = '16'
XX= RPROVED
DO VALCHECK WITH XX
Y = '17'
XX=RPROCEABLE
DO VALCHECK WITH XX
Y = '18'
XX = RPOSSIBLE
DO VALCHECK WITH XX
Y = '19'
XX=FMPROVED
DO VALCHECK WITH XX
Y = '20'
XX=FMPROBABLE
DO VALCHECK WITH XX
Y = '21'
XX=FMPOSSIBLE

```



```

DO VALCHECK WITH XX
Y = '22'
XX=SMPCVED
DO VALCHECK WITH XX
Y = '23'
XX=SMPCBABLE
DO VALCHECK WITH XX
Y = '24'
XX=SMPCSSIBLE
DO VALCHECK WITH XX
Y = '25'
XX=FRCSGTY
DO VALCHECK WITH XX
Y = '26'
XX=FRCSGTY
DO VALCHECK WITH XX
IF VAL(UNIT) < 1 .OR. VAL(UNIT) > 5
  Y='28'
  CC ERROR
ENDIF
IF FLAG = 'Y'
  CELE
  FLG = 1
  SELE 4
  APPEND BLANK
  SELE 3
ENDIF
IF VAL(ID) > 1
  CC STORVAL
  CC ISTDIST
ENDIF
RECNO = RECNO()
SKIP
ENDDC
SET CELE ON
CLEAR
SET ALTERNATE TO 8COMM
SET ALTERNATE ON
*SET PRINT ON
? ".FL 72"
? ".MT 0"
? ".ME 0"
? ".PC 0"
? ".CF"
? SPACE(58)+"FILE NAME - ",COMM
? SPACE(58)+"*****"
?
? SPACE(54)+"MINERAL - ",MIN
? SPACE(54)+"*****"
?
? "THE LIST OF CORRECT RECCRS"
? "-----"
?
LCTR = LCTR+9
CC TOP
IF .NOT. ECF()
  CC HEADLIST
ELSE
  ? "ALL THE RECCRS ARE INCORRECT. SEE THE LIST ON THE NEXT PAGE."
  LCTR = LCTR+1
ENDIF
CC WHILE .NOT. ECF()
  IF LCTR = C
    ? ".PA"

```



```

DO HEADLIST
ENDIF
? STR(RECNC(),5)+SPACE(3)+MINERAL,SPACE(3),DEPCST,SPACE(2),IS+SPACE(2)+ID
?? SPACE(2)+STATE,SPACE(2),DIST,SPACE(1),BLOCK,SPACE(2),DEFNO,BLOCKNO,STATUS_A
? SPACE(7),STATUS_B,SPACE(6),STATUS_C,SPACE(6),IGRADE,SPACE(5)
?? IPRCVD,IPRCBABLE,IPOSSIBLE,RGRADE,SPACE(3),RPRCVD,RPRCBABLE
? SPACE(7),RPOSSIBLE,FMGRADE,FMPROVED,FMPRCBABLE,FMPOSSIBLE,SMGRADE
? SPACE(7),SMPROVED,SMPCBABLE,SMPCSSIBLE,FROSGRADE,PROSGTY,PROGGTY,UNIT
? "-----"
?? "-----"
LCTR = LCTR+5
IF LCTR+5 > 68
    LCTR = C
ENDIF
SKIP
ENDDC
SET DELE CFF
GC TOP
SET FILTER TO DELETE()
GC TOP
IF EOF()
    ?
    ?
    ? "THERE IS NO ERROR IN THE DATA & ALL THE RECORDS ARE CORRECT"
    ? "-----"
ELSE
    DO WHILE LCTR < 72
        ?
        LCTR = LCTR+1
    ENDC
    LCTR = C
    ? ".PA"
    *EJECT
    ? "THE LIST OF INCORRECT RECORDS"
    ? "-----"
    ?
    LCTR = LCTR+3
    DO HEADLIST
    SELE 4
    GO TOP
    SELE 3
    DO WHILE .NOT. EOF()
        IF LCTR = C
            ? ".PA"
            DO HEADLIST
        ENDIF
        ? STR(RECNC(),5)+SPACE(3)+MINERAL,SPACE(3),DEPCST,SPACE(2),IS+SPACE(2)+ID
        ?? SPACE(2)+STATE,SPACE(2),DIST,SPACE(1),BLOCK,SPACE(2),DEFNO,BLOCKNO,STATUS_A
        ? SPACE(7),STATUS_E,SPACE(6),STATUS_C,SPACE(6),IGRADE,SPACE(5)
        ?? IPRCVD,IPRCBABLE,IPCSSIBLE,RGRADE,SPACE(3),RPRCVD,RPRCBABLE
        ? SPACE(7),RPOSSIBLE,PMGRADE,FMPROVED,FMPRCBABLE,PMPOSSIBLE,SMGRADE
        ? SPACE(7),SMPROVED,SMPCBABLE,SMPCSSIBLE,FROSGRADE,PROSGTY,PROGGTY,UNIT
        ? "-----"
        ?? "-----"
        ? "ERROR CODES : "
        SELE 4
        KOUNT = 1
        I = 1
        DO WHILE I < 29
            II=STR(I,2)
            IF I < 10
                II="C"+STR(I,1)
            ENDIF
            IF C&II # SPACE(3)

```



```

        IF KCUNT = 1
            ?? C&II
        ELSE
            ?? " "+C&II
        ENDIF
        KCUNT = KCUNT+1
    ENDIF
    I = I+1
ENDDO
SKIP
SELE 3
? "-----"
?? "-----"
LCTR = LCTR+7
IF LCTR+7 > 68
    LCTR = 0
    *EJECT
ENDIF
KNT = KNT+1
SKIP
ENDDO
ENDIF
RECALL ALL
IF FLAGS = 1
    ? "** NOTE |- THE UNIT CODE IS NOT CONSISTENT THROUGH OUT THE FILE"
    ? "-----"
ENDIF
?
?
? "SUMMARY OF RECCRDS"
? "-----"
?
? "NC. OF TOTAL RECCRES      = ",STR(RECNC,4)
? "NC. OF CORRECT RECCRDS   = ",STR(RECNC - KNT,4)
? "NC. OF INCORRECT RECCRES = ",STR(KNT,4)
? "-----"
*SET PRINT OFF
SET ALTERNATE OFF
SET ALTERNATE TO
CLEAR
& 10,10 SAY "THE OUTPUT HAS BEEN STORED IN "+CCMM+".TXT FILE "
& 12,10 SAY "IT CAN BE SEEN OR PRINTED USING WORDSTAR"
SET PROC TO MAINPROC

```



```

*****
** This program is a procedure file which contains the various small
** procedures use in the validation program named 'valic.prg'.
*****

```

```

PROCEDURE UNITREFL
CC WHILE .NOT. ECF()
  IF UNIT # SPACE(1)
    IF FLAGR = 0
      SUNIT = UNIT
      FLAGR = 1
    ELSE
      IF SUNIT # UNIT
        FLAGR = 1
        EXIT
      ENDIF
    ENDIF
  ENDIF
  SKIP
ENDCC
IF FLAGR = 1 .AND. FLAGR = 0
  REFL ALL UNIT WITH SUNIT
ENDIF
GC TOP

```

```

*****
PROCEDURE CORRECT
PARAMETERS X,Y
TEMP = SUBSTR(Y,1,1)
IF TEMP # '0' .AND. TEMP # SPACE(1)
  REFL &X WITH 'C'+TEMP
ELSE
  IF TEMP = SPACE(1)
    TEMP = SUBSTR(Y,2,1)
    REFL &X WITH 'C'+TEMP
  ENDIF
ENDIF

```

```

*****
PROCEDURE VALCHECK
PARAMETERS X
IF X < 0
  FLAGB = 'Y'
  SELECT 4
    REFL C&Y WITH 'E&Y'
  SELECT 3
ENDIF

```

```

*****
PROCEDURE STORVAL
PUBLIC DEF,IST,IDI,D,S,B,SA,SE,SC
DEF=DEPOSIT
IST=IS
IDI=IC
C=DISI
S=STATE
E=BLOCK
SA=STATUS_A
SB=STATUS_B
SC=STATUS_C

```

```

*****

```



```

PROCEDURE REPLACE
REPL IGRADE WITH " "
REPL IPROVED WITH C
REPL IPRCEABLE WITH C
REPL IPOSSIBLE WITH C
REPL RGRADE WITH " "
REPL RPROVED WITH C
REPL RPRCEABLE WITH C
REPL RPOSSIBLE WITH C
REPL FMGRADE WITH SPACE(25)
REPL FMPROVED WITH C
REPL FMPRCEABLE WITH C
REPL FMPOSSIBLE WITH C
REPL SMGRADE WITH SPACE(25)
REPL SMPROVED WITH C
REPL SMPRCEABLE WITH C
REPL SMPPOSSIBLE WITH C
REPL PROSGTY WITH C
REPL PROGGTY WITH C

```

```

PROCEDURE ERROR
  FLAGB = 'Y'
  SELECT 4
    REFL C&Y WITH "E&Y"
  SELECT 3

```

```

PROCEDURE HEADLIST
* THIS PROCEDURE PRINTS THE HEADINGS FOR LISTING OF CORRECT & INCORRECT RECORDS
? "=====
?? "=====
? "R.NO.  ","MINERAL","DEPOSIT","IS","ID","STATE","DIST","BLOCK","DEFNO",SPACE(30)
?? "BLCKNO",SPACE(28),"STATUS_A"
? SPACE(7),"STATUS_B","STATUS_C","IGRADE",SPACE(8),"IFROVED",SPACE(7),"IFROBABLE"
?? SPACE(7),"IPOSSIBLE","RGRADE",SPACE(8),"RPROVED",SPACE(6),"RPRCEABLE"
? SPACE(14),"RPOSSIBLE","FMGRADE",SPACE(25),"FMPROVED",SPACE(5),"FMPROBABLE"
?? SPACE(6),"FMPOSSIBLE","SMGRADE"
? SPACE(15),"SMPROVED",SPACE(5),"SMPRCEABLE",SPACE(5),"SMPOSSIBLE"
?? SPACE(1)+ "PROSGRADE",SPACE(24),"PROSGTY",SPACE(8),"PROGGTY","UNIT"
? "=====
?? "=====
LCCTR = LCCTR+6

```

```

PROCEDURE ISTLIST
KCUNT = 1
FLAG = 0
FG = C
RNO = RECNO()
SKIP
K = 1
IF VAL(DEPOSIT)=VAL(DEP) .AND. IS=IST .AND. ID=IDI
  IF VAL(STATE) # VAL(S) .OR. VAL(DIST) # VAL(D)
    I = 1
    DO WHILE I < VAL(IDI)
      IF VAL(DEPOSIT)=VAL(DEP) .AND. IS=IST .AND. ID=IDI
        DO REPLACE
          FLAGB='N'
          KCUNT=KCUNT+1
          IF VAL(STATE) < 10 .AND. STATE # SPACE(2)
            X='STATE'
            Y=STATE
            DO CORRECT WITH X,Y

```

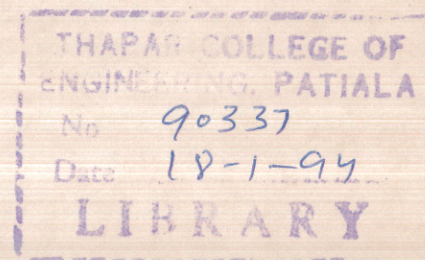


```

ENDIF
IF VAL(DIST) < 10 .AND. DIST # SPACE(2)
  X='DIST'
  Y=DIST
  DC CORRECT WITH X,Y
ENDIF
SCD=STATE
DCC=DIST
SELECT 2
GC TCF
SET FILTER TO STATE=SCD
GC TCF
IF ECF()
  Y='C6'
  DC ERROR
  Y='C7'
  DC ERROR
ELSE
  LOCATE FOR DIST=DCC
  IF EOF()
    Y='C7'
    DC ERROR
  ENDIF
  SELECT 2
  SET FILTER TO
  SELECT 3
ENDIF
IF FLAGB = 'Y'
  DELETE
  K = K+1
  SELE 4
  APPEND BLANK
  SELE 3
ENDIF
SKIP
I = I+1
ELSE
  EXIT
ENDIF
ENDLC
ELSE
  FLAG = 1
ENDIF
ENDIF
SKIP - 1
RN = RECNO()
IF FLAG = 0 .AND. KOUNT # VAL(IDI)
  GC RNC
  DELETE
  SELE 4
  IF RECNO() = 1
    FG = 1
  ENDIF
  SKIP - K
  IF FLG = 0
    IF FG = 1
      INSE BLANK BEFORE
    ELSE
      INSE BLANK
    ENDIF
  ENDIF
  Y = '27'
  DC ERROR
  SELE 4

```

No.....Dr.....
 CENTRAL LIBRARY
 PATIALA.
 Thapar Institute of Engineering & Technology



SKIP K
SELE 3
GC RN
ENDIF