CGTM 76
Lance J. Hoffman

Incremental Cost of Encoding Data in a Tape-Based Information Storage and Retrieval System
or
It's Dirt Cheap to Scramble Your Data!

Up to this point, no published results have appeared dealing with the cost of encoding (scrambling) information in an information storage and retrieval system. Opponents of more stringent access controls in computer systems have cited the high cost of adding additional software control functions to the already overburdened operating systems, but to this author's knowledge no hard timing data has yet appeared.

In a quest for some ballpark figures on the costs of encoding data, (and conversely the costs of decoding data), an experiment was run on the IBM 360/67 computer system at the Campus Facility of Stanford University Computation Center. A tape containing 1,050 80-character card images in clear (unscrambled) format was generated and the 80-character records were sequentially read in, scrambled, and the (encoded) card images written out on a new output tape. The appendix contains the listing of the FORTRAN program used to do this job, and also the printout of the timing results.

As can be seen by an examination of the printout, the time used to scramble 1,050 cards using the simple encoding technique shown in the program listing, was 7.5612 seconds. (Actually this is the time used to scramble 1,049 cards, since writing of the first output record required a one-time opening of the data set by the operating system. Since we were trying to get an estimate of the time required to encode a large number of records, this "open time" was disregarded. It would certainly be negligible for any

large-scale data base.) This time, 7.5612 seconds, is the total wall-clock time used from the time the first clear record was read in until the time the last encoded record was written out onto the output tape. All waits for input/output, etc., are included in this time. Again, the time does not represent central processor cycles. It is wall-clock time on a system where this was the only job running in addition to the operating system. (In other words, this job used the entire 600K byte partition on the 360/67.) The experiment was carried out in this manner in order to get a high estimate of the incremental cost involved in scrambling a large number of cards. In a multi-programming system the actual time used in encoding could be overlapped with tasks from other jobs and therefore would not nearly be so costly.

In this worst case we see that 1,049 cards were scrambled in 7.5612 seconds. We can put it another way; the incremental cost of encoding one card image on this system is .007 seconds. Uder the existing rate structure at the Stanford Computation Center, it then costs 0.08 cents to encode each card image. Therefore, encoding one card image (80 bytes of information) for each of the 20,000,000 residents of the State of California would cost only $16,000.

Conclusion

The incremental cost of scrambling information in a large computer data base seems infinitesimal.

Appendix. Printout from Experiment, Including FORTRAN Code and Results

```
* 2.03.52 JOB   732 IEF234A R 0C2
* 2.03.52 JOB   732 IEF234A R 0C3
* 2.03.52 JOB   732 IEF233A M 0C2,2405   ,LJHFAKE
* 2.03.52 JOB   732 IEF233A M 0C3,SCRTCH,LJHFAKE
* 2.04.23 JOB   732 IHC002I STOP      0
* 2.04.24 JOB   732 IEF280I K 0C2,2405   ,LJHFAKE
```

```
//LJHFAKE JOB (N874***,343,2,1,,,,T),'HOFFMAN',MSGLEVEL=1                    JOB
// EXEC FORTHCLG
//FORT  EXEC  PGM=IEKAA00,PARM='MAP'                                         000
//SYSPRINT DD  SYSOUT=A                                                      000
//SYSABEND DD SYSOUT=A                                                       000
//SYSPUNCH DD  UNIT=SYSCP                                                    000
//SYSUT1 DD DSNAME=&X,UNIT=SYSDA,SPACE=(CYL,(10,5))                          000
//SYSUT2 DD DSNAME=&Y,UNIT=SYSDA,SPACE=(CYL,(10,5))                          000
//SYSLIN   DD  DSNAME=&LOADSET,UNIT=2314,DISP=(MOD,PASS),                   *000
//             SPACE=(CYL,(3,1)),DCB=(LRECL=80,BLKSIZE=3200,RECFM=FB)        000
//FORT.SYSIN DD *
IEF236I ALLOC. FOR LJHFAKE  FORT
IEF237I SYSPUNCH ON 00D
IEF237I SYSUT1   ON 230
IEF237I SYSUT2   ON 230
IEF237I SYSLIN   ON 230
IEF237I SYSIN    ON 00C
```

```
   COMPILER OPTIONS - NAME=  MAIN,OPT=00,LINECNT=57,SOURCE,EBCDIC,NOLIST,NODECK,LOAD
002            BLOCK DATA                                                        345
003            IMPLICIT INTEGER(A-Z)                                             346
004            COMMON/CONSTANTS/NUCB,NFORM,MAXUSERS,MAXLLIST,ITALK,              347
              1    FORM1,FORM2,FORM3,                                           348
              2    NEXTALL,SAMEALL,                                             349
              3    FETCHP,STOPEP,UNLEEP,UNLSTP,ELOCKP,SLOCKP,ATTACHP,DETACHP    350
005            COMMON/ADDL1/IRAND,IRPT,PASSWD,USER,CARDA,PWTBL,IPWTBL,UTBL       351
006            COMMON/UCB/ISTDUCB                                                352
007            COMMON/OWN1/UCB1,LLIST,CS1
008            INTEGER PASSWD(10),USER(10),PWTBL(10,10), UTBL(10,10)             354
009            INTEGER IRPT(4),CARDA(80),IRAND(20)
010            INTEGER UCB1(100,3)                                               356
       C USER CONTROL BLOCKS -- MAXUSERS=100, NUCB=3                             357
       C                                                                         358
011            DATA IRAND/-143295037, 12498331, -99905473, 107015948,            359
              1           -35891432, 13737389, -254817906, 227051690,            360
              2           267059188,-305496183,132598180,-133310762,
              3           -124986609,-143295037,243176905,-240111797,
              4           199832006,-178963561, -219961227,-174003653/
012            DATA USER/10*0/                                                   362
013            DATA PASSWD/10*0/                                                 363
014            INTEGER ISTDUCB(3)/0,0,1/                                         364
       C STANDARD USER CONTROL BLOCK (TEMPLATE)                                  365
015            INTEGER LLIST(4,100)/400*-1/                                      366
       C THE LOCKLIST -- MAXLLIST=100                                            367
016            DATA IPWTBL/4/                                                    368
       C NUMBER OF LEGITIMATE USERS OF SHS SYSTEM                                369
017            DATA IRPT/1,1,1,1/                                                370
       C USER IDENTIFICATIONS TRANSFORMED BY APPROPRIATE ALGORITHM               371
018            DATA UTBL/1384446165, 1803822108, -1730839952,                    372
              1       -1547876608, 6*0,                                          373
              2  1935297975, -2035586489, -1164487326,                         374
              3       -1268502271, 6*0,                                          375
              4  1692507097, -581819502, 1223346686, 7*0,                      376
              5  1692507097, -1377663127, 1618028494, 101222, 6*0,             377
              6  60*0/                                                         378
       C PASSWORDS TRANSFORMED BY APPROPRIATE ALGORITHM                          379
019            DATA PWTBL/1691037135, -1182530491, -951364798,                   380
              1  25995779, 6*0,                                                 381
              2 -162905104, 751179207, -1043920360, -589549044,                382
              3 101222, 5*0,                                                   383
              4 1665843558, 1801889415, -727933704, 7*0,                       384
              5 -1552084301, 2064554537, -2033533773, 814026026,              385
              6   6*0,60*0/                                                    386
       C CONSTANTS USED THROUGHOUT THE SYSTEM                                    387
       C                                                                         388
020            DATA ITALK/2/                                                     389
       C LENGTH OF ARRAY PASSED BY TALK PROGRAM TO ACCESS PROGRAM                390
021            DATA MAXLLIST/100/                                                391
       C MAXIMUM LENGTH OF LIST OF LOCKED DATUMS MAINTAINED BY ACCESS PGM        392
022            DATA UCB1(1,1)/-2/                                                393
       C INITIALIZE TO NO ACTIVE USER CONTROL BLOCKS                             394
023            DATA CS1/1/
       C INITIALIZE TO CRITICAL SECTION OF ACCESS PROC. NOT CURRENTLY IN USE
024            DATA NUCB/3/                                                      395
```

```
C NUCB = NO. OF WORDS IN EACH USER CONTROL BLOCK                    396.
      DATA MAXUSERS/100/                                           397.
C MAXUSERS = MAX. NO. OF USER/TERMINAL COMBINATIONS                 398.
C POSSIBLE AT ANY GIVEN TIME                                        399.
      DATA FORM1/1/                                                400.
      DATA FORM2/2/                                                401.
      DATA FORM3/3/                                                402.
C INTERNAL NAMES OF FORMULARIES                                     403.
      DATA NFORM/3/                                                404.
C NUMBER OF FORMULARIES CURRENTLY IN THE SYSTEM                     405.
      DATA NEXTALL/1000/,SAMEALL/1001/                             406.
C INTERNAL NAMES                                                    407.
C (IN THIS CASE ALSO VIRTUAL ADDRESSES)                             408.
      DATA FETCHP/1/,STOREP/2/,UNLFEP/3/,UNLSTP/4/,                409.
     1  FLOCKP/5/,SLOCKP/6/,ATTACHP/7/,DETACHP/8/                  410.
      END                                                          411.
```

```
  COMPILER OPTIONS - NAME=  MAIN,OPT=00,LINECNT=57,SOURCE,EBCDIC,NOLIST,NODECK,LOAD
002           SUBROUTINE SCRAMBLE(IUCB,CLRBUF,ICLRLEN,ICOMPL,SCRBUF,ISCRLEN)      963
              C THIS SUBROUTINE SCRAMBLES THE UNSCRAMBLED DATUM WHICH              964
              C IS ICLRLEN CHARACTERS LONG STARTING IN CLRBUF(1), AND IS           965
              C STORED FOUR CHARACTERS PER WORD.  IT LEAVES THE                    966
              C SCRAMBLED DATUM IN THE FIRST ISCRLEN BYTES OF THE SCRBUF           967
              C ARRAY (AND RETURNS ISCRLEN TO THE CALLING ROUTINE).                968
              C THIS SUBROUTINE STORES A COMPLETION CODE IN ICOMPL.                969
              C 0 < ISCRLEN < 81 AND 0 < ICLRLEN < 81.                             970
              C                                                                    971
              C COMPLETION CODES STORED IN ICOMPL:                                 972
              C     1    NORMAL EXIT                                               973
              C     2    SCRAMBLE OPERATION NOT PERMITTED BY THIS FORMULARY        974
              C     3    ILLEGAL LENGTH OF  DATUM TO SCRAMBLE                      975
              C                                                                    976
              C ******* CERTIFIED 8 MAY 1969 *****                                 977
              C                                                                    978
              C                                                                    979
003           COMMON/CONSTANTS/NUCB,NFORM,MAXUSERS,MAXLLIST,ITALK,                 980
                1   FORM1,FORM2,FORM3,                                           981
                2   NEXTALL,SAMEALL,                                             982
                3   FETCHP,STOREP,UNLFEP,UNLSTP,FANDLP,SANDLP,ATTACHP,DETACHP    983
004           INTEGER SCRBUF(20),CLRBUF(20),IUCB(NUCB)                           984
005           COMMON/ADDL1/IRAND,IRPT,PASSWD,USER,CARD1,PWTBL,IPWTBL,UTBL        985
006           INTEGER PASSWD(10),USER(10),PWTBL(10,10), UTBL(10,10)              986
007           INTEGER IRPT(4),CARD1(80),IRAND(20)
              C ------------------------------------------                        988
              C -------------- FORMULARY SELECTOR -------------                   989
              C ------  ------  ------  -------  --------  ----                  990
              C                                                                    991
008           III=IUCB(3)                                                        992
009           GO TO (3,1,3), III                                                 993
              C                                                                    994
              C --------  -------  ---------  ---------  -------                 995
              C ------------------------------------------------                 996
010           1 IF ((ICLRLEN .GT. 80) .OR. (ICLRLEN .LT. 1)) GO TO 4             997
012             ISCRLEN=(ICLRLEN-1)/4+1                                          998
013             DO 2 I=1,ISCRLEN                                                 999
014           2 SCRBUF(I)=LGC1XR(CLRBUF(I),IRAND(I))                            1000
015             ISCRLEN=ICLRLEN                                                 1001
016             ICOMPL=1                                                        1002
017             RETURN                                                          1003
018           3 ICOMPL=2                                                        1004
019             RETURN                                                          1005
020           4 ICOMPL=3                                                        1006
021             RETURN                                                          1007
022             END                                                             1008
```

```
COMPILER OPTIONS - NAME=  MAIN,OPT=00,LINECNT=57,SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP
      C
      C THIS PROGRAM SCRAMBLES THE DATA ON DATA SET 8 AND PUTS THE
      C SCRAMBLED VERSION ON DATA SET 9.
      C
            IMPLICIT INTEGER (A-Z)
            INTEGER CARD(20)
            INTEGER SCRB(20)
            INTEGER A(20)
            INTEGER IUCB1(3)
            INTEGER CLOCK1
            REAL ITOLD
            IUCB1(3)=2
      C KLUDGE FOR ATTACHING FML 2
            ICOUNT=0
          2 CONTINUE
            READ(8,900,ERR=2   ,END=802) CARD
            ITOLD=CLOCK1(4)
            GO TO 3
         20 READ(8,900,ERR=20,END=802) CARD
          3 ICOUNT=ICOUNT+1
        900 FORMAT(20A4)
            CALL SCRAMBLE(IUCB1,CARD,80,ICOMP,SCRB,ISCR)
            WRITE(9,900) SCRB
            GO TO 20
        802 CONTINUE
            ITOLD=(CLOCK1(4)-ITOLD)*26/1000000
            WRITE(6,960) ITOLD
        960 FORMAT(' TIME USED WAS ',F8.4,' SECONDS')
            END FILE 9
            WRITE(6,950) ICOUNT
        950 FORMAT(' ',I7,' CARDS WERE SCRAMBLED.')
            REWIND 9
            III=0
        300 CONTINUE
            READ( 9,100,END=200)A
        100 FORMAT(20A4)
            III=III+1
            IF (III .LT. 100) WRITE(6,101)A
        101 FORMAT(' ',20A4)
            GO TO 300
        200 CONTINUE
            WRITE(6,102) III
        102 FORMAT(' END OF SCRAMBLED FILE FOUND, ',I8,' CARDS READ')
            RETURN
            END
```

```
IEF285I   SYSOUT                                                  SYSOUT
IEF295I   VOL SER NOS=          .
IEF285I   X.LJHFAKE                                               DELETED
IEF285I   VOL SER NOS= SYS02 .
IEF285I   Y.LJHFAKE                                               DELETED
IEF285I   VOL SER NOS= SYS02 .
IEF285I   LOADSET.LJHFAKE                                         PASSED
IEF285I   VOL SER NOS= SYS02 .
//LKED  EXEC  PGM=IEWL,PARM='LET,LIST,MAP',COND=(5,LT,FORT)                00
//SYSLIN    DD  DSNAME=*.FORT.SYSLIN,DISP=(OLD,DELETE)                     00
//          DD  DDNAME=SYSIN                                               00
//SYSLIB    DD  DSNAME=SYS1.FORTLIB,DISP=OLD                               00
//          DD  DSNAME=SYS2.SSPLIB,DISP=OLD                                00
//          DD  DSNAME=SYS2.SUBLIB1,DISP=OLD                               00
//SYSUT1    DD  DSNAME=SYS1.UT1,DISP=OLD,DCB=(KEYLEN=0)                    00
//SYSLMOD   DD  DSNAME=&GOSET(MAIN),DISP=(,PASS),UNIT=2314,               *00
//              SPACE=(CYL,(12,1,1))                                       00
//SYSPRINT DD   SYSOUT=A                                                   00
IEF236I ALLOC. FOR LJHFAKE  LKED
IEF237I SYSLIN    ON 230
IEF237I SYSLIB    ON 235
IEF237I           ON 331
IEF237I           ON 331
IEF237I SYSUT1    ON 1C0
IEF237I SYSLMOD   ON 230
```

```
IEF285I   LOADSET.LJHFAKE                                        DELETED
IEF285I   VOL SER NOS= SYS02 .
IEF285I   SYS1.FORTLIB                                           KEPT
IEF285I   VOL SER NOS= SYS00 .
IEF285I   SYS2.SSPLIB                                            KEPT
IEF285I   VOL SER NOS= SYS01 .
IEF285I   SYS2.SUBLIB1                                           KEPT
IEF285I   VOL SER NOS= SYS01 .
IEF285I   SYS1.UTI                                               KEPT
IEF285I   VOL SER NOS= CAMPO8.
IEF285I   GOSET.LJHFAKE                                          PASSED
IEF285I   VOL SER NOS= SYS02 .
IEF285I   SYSOUT                                                 SYSOUT
IEF285I   VOL SER NOS=       .
//GO     EXEC  PGM=*.LKED.SYSLMOD,COND=((5,LT,FORT),(5,LT,LKED))          00001
//FT01F001 DD DSNAME=SYS1.UT1,DISP=OLD,DCB=(RECFM=V,LRECL=796,            X00001
//            BLKSIZE=800)                                               00002
//FT02F001 DD DSNAME=SYS1.UT2,DISP=OLD,DCB=*.FT01F001                    00002
//FT03F001 DD DSNAME=SYS1.UT3,DISP=OLD,DCB=*.FT01F001                    00002
//FT04F001 DD DSNAME=SYS1.UT4,DISP=OLD,DCB=*.FT01F001                    00002
//FT05F001 DD  DDNAME=SYSIN                                              00002
//FT06F001 DD  SYSOUT=A                                                  00002
//FT07F001 DD  UNIT=SYSCP                                                00002
//FT13F001 DD DSNAME=SYS1.UT5,DISP=OLD,DCB=*.FT01F001                    00002
//FT16F001  DD UNIT=2314,VOLUME=SER=SYS03,DISP=(NEW,PASS),               X00002
//            DCB=(RECFM=F,BLKSIZE=600),SPACE=(TRK,(15,5),RLSE)          00002
//GO.FT08F001 DD UNIT=0C2,DISP=(OLD,KEEP),LABEL=(1,BLP),DSNAME=X,        1
//            VOLUME=SER=2405,DCB=(LRECL=80,RECFM=FB,BLKSIZE=3520)
//GO.FT09F001 DD UNIT=0C3,DISP=(NEW,DELETE),DSNAME=X,LABEL=(,NL),        1
//            DCB=(LRECL=80,RECFM=FB,BLKSIZE=3520)
//
IEF236I ALLOC. FOR LJHFAKE  GO
IEF237I PGM=*.DD ON 230
IEF237I FT01F001 ON 1C0
IEF237I FT02F001 ON 230
IEF237I FT03F001 ON 330
IEF237I FT04F001 ON 230
IEF237I FT07F001 ON 00D
IEF237I FT13F001 ON 330
IEF237I FT16F001 ON 330
IEF237I FT08F001 ON 0C2
IEF237I FT09F001 ON 0C3


TIME USED WAS    7.5612 SECONDS
    1C5C CARDS WERE SCRAMBLED.
8  #8C805:¬"I-?<:K@Y J%_0|I+(HH |  "4_ "8GXIN8(OO8JGE75'C+=NI10;9' C5 X#P2TY55
GE(30+E,.#/"7-?<:/(Y J)_0|I+(HH |  "4_ "8GXIN8(OO8JGE75'C+=NI10*#9Z1F5N#P2TY55
GE(30+E,.#/"7-?<./@Y -)_0|I+(HH |  "4_ "8GXIN8(OO8JGE75'C+=NI10*#9 C=DX#P2TY55
GE(30+E,.#/"7-?<:/(Y J)_0|I+(HH |  "4_#"IGXIN8(OO8JGE75'C+=NI10*#9Z1F5N#P2TY55
GE(30+E,.#/"7-?<:/@Y -)_0|I+(HH |  "4_ "8GXIN8(OO8JGE75'C+=NI10;9' C2AX#P2TY55
GE(30+E,.#/"7-?<:/@Y -)_0|I+(HH |  "F_ "IGXIN8(OO8JGE75'C+=NI10;#: C=AX#P2TY55
GE(30+E,H@/"7-?<:/(Y J)_0|I+(HH |  "F_ "IGXIN8(OO8JGE75'C+=NI10*#9 C=DXHVG  55
GE(30+E,H@/"7-?<:/(Y J)_0|I+(HH |  "4_ "IGXIU8(OO8JGE75'C+=NI10$##AA6<U#P2TY55
GE(30+E,H2/"4-?<:/(Y J)_0|I+(HH |  "4_ "IGXIU8(OO8JGE75'C+=NI10;#:Z1F5N#P2TY55
GE(30+E,.#/"7-?<./(Y J)_0|I+(HH |  "F_ "IGXIUI(OO8JGE75'C+=NI10*#9 C=DX#P2TY55
F H57  ZH9/"7-?<:L@Y -)_0|I+(HH |  "F_ "IGXIN8(OO8JGE75'C+=NI10$## C2GU#P2TY55
F .C7GFY.9/"7-?<:K(Y J)_0|I+(HH |  "4_ "8GXIN8(OO.JGE75'C+=N810*@: C3DX#P2TY55
F .07GFY.9/"7-?<:K@Y -)_0|I+(HH |  "F_ "IGXIN8(OO8JGE75'C+=NI10*@: C3<U#P2TY55
F .C7GFY.9/"7-?<:K(Y J)_0|I+(HH |  "4_ "8GXIN8(OO8JGE75'C+=NI10*@: C=DX#P2TY55
F .07GFY.9/"7-?<:K@Y -)_0|I+(HH |  "4_ "8GXIN8(OO8JGE75'C+=NI10*@: C=DXHVGK 55
```

```
F .C7GFY.9/"7-?<:L(Y J)_0|I+(HH | "4_ "8GXIN8(008JGE75'C+=NI10*@: C3DX#P2TY5
F .C7GFY.9/"7-?<:K(Y J)_0|I+(HH | "4_ "8GXIN8(008JGF75'C+=NI10*@: C=DX#P2TY5
F  48+  .9/"7-?<:K@Y -)_0|I+(HH | "F_ "IGXIN8(008JGE75'C+=NI10;#: C15X#P2TY5
F  48|D .9/"7-?<:K(Y J)_0|I+(HH | "F_ "IGXIN8(008JGE75'C+=NI10*@: C=DX#P2TY5
F  49|C .9/"7-?<:K(Y J)_0|I+(HH | "4_ "8GXIN8(008JGE75'C+=NI10*@: C=DX#P2TY5
F  48|D .9/"7-?<:K(Y J)_0|I+(HH | "4_ "8GXIN8(008JGE75'C+=NI10*@: C=DX#P2TY5
F D67 ( .:/"7-?<:/(Y J)_0|I+(HH | "4_#"IGXIU8(008JGE75'C+=NI10*#9Z1F5N#P2TY5
F D67 ( .:/"7-?':/@Y -)_0|I+(HH | "4_#"IGXIU8(0X8JGE75'C+=NI10*@: C3<X<?C0 5
FC(77.FZH9/"7-?<:K@Y J%_0|I+(HH | "4_ "8GXIN8(008JGF75'C+=NI10;#: C=5X#P2TY5
FC(77.FZH9/"7-?<:K@Y J)*0|I+(HH | "4_ "8GXIN8(008JGE75'C+=NI10;#: C65X#P2TY5
FC(77.FZH9/"7-?<:L@Y J%_0|I+(HH | "4_ "8GXIN8(008JGE75'C+=NI10;#: C=5X#P2TY5
FC(:4.(Y.9/"7-?<:L@Y J%_0|I+(HH | "4_ "8GXIN8(008JGE75'C+=NI10;9' C2EX#P2TY5
FC(:4.(Y.9/"7-?<:L@Y J%_0|I+(HH | "4_ "8GXIN8(008JGE75'C+=NI10;9' C2EX#P2TY5
FC(:4.(Y.9/"7-?<:L@Y J%_0|I+(HH | "4_ "8GXIN8(008JGE75'C+=NI10;9' C2EX#P2TY5
FC(:4.(Y.9/"7-?<:L@Y J%_0|I+(HH | "4_ "8GXIN8(008JGE75'C+=NI10;9' C2EX#P2TY5
FC(:4.(Y.9/"7-?<:L@Y J%_0|I+(HH | "4_ "8GXIN8(008JGF75'C+=NI10;9' C2EX#P2TY5
FC(:4.(Y.9/"7-?<:&@Y J%_0|I+(HH | "4_ "8GXIN8(008JGE75'C+=NI10;9' C2EX#P2TY5
FC(:4.(Y.9/"7-?<:L@Y J%_0|I+(HH | "4_ "8GXIN8(008JGE75'C+=NI10;9' C2EX#P2TY5
FC<14.A%.9/"7-?<:&@Y -)_0|I+(HH | "4_ "8GXIN8(008JGE75'C+=NI10;9: C=CU#P2TY5
FC<14.A%.9/"7-?<:&@Y -)_0|I+(HH | "4_ "8GXIN8(008JGE75'C+=NI10;9: C=CU#P2TY5
FC<14.A%.9/"7-?<:K@Y -)_0|I+(HH | "F_ "IGXIN8(008JGE75'C+=NI10$## C=EU#P2TY5
FC<14.A%.9/"7-?<:&@Y -)_0|I+(HH | "4_ "8GXIN8(008JGF75'C+=NI10;9: C=CX#P2TY5
FC<14.A%.9/"7-?<:&@Y -)_0|I+(HH | "4_ "8GXIN8(008JGE75'C+=NI10;9: C=CX#P2TY5
FC<14.A%.9/"7-?<:&@Y -)_0|I+(HH | "4_ "8GXIN8(008JGE75'C+=NI10;9: C=CX#P2TY5
FC<14.A%.9/"7-?<:&@Y -)_0|I+(HH | "4_ "8GXIN8(008JGE75'C+=NI10;9: C=CX#P2TY5
FC<14.A%.9/"7-?<:&@Y -)_0|I+(HH | "4_ "8GXIN8(008JGE75'C+=N810;9: C=CX#P2TY5
FC<14.A%.9/"7-?<:&@Y -)_0|I+(HH | "4_ "8GXIN8(008JGE75'C+=NI10;9: C=CX#P2TY5
FC<14.A%.9/"7-?<:&@Y -)_0|I+(HH | "4_ "8GXIN8(008JGE75'C+=NI10;9: C=CX#P2TY5
FC<79(CYH9/"7-?<:K@Y -)_0|I+(HH | "4_ "8GXIN8(008JGE75'C+=NI10¬@9 C=DX#P2TY5
FC<79(CYH9/"7-?<:K@Y -)_0|I+(HH | "4_ "8GXIN8(008JGE75'C+=NI10¬@9 C=DX#P2TY5
FC<79(CYH9/"7-?<:K@Y -)_0|I+(HH | "4_ "8GXIN8(008JGE75'C+=NI10¬@9 C=DX#P2TY5
FC<79(CYH9/"7-?<:K@Y -)_0|I+(HH | "4_ "8GXIN8(008JGE75'C+=NI10¬@9 C=DX#P2TY5
FC<79(CYH9/"7-?<:K@Y -)_0|I+(HH | "4_ "8GXIN8(008JGE75'C+=NI10¬@9 C=DX#P2TY5
FC<79(CYH9/"7-?<:K@Y -)_0|I+(HH | "4_ "8GXIN8(008JGE75'C+=NI10¬@9 C=DX#P2TY5
FC<79(CYH9/"7-?<:K@Y -)_0|I+(HH | "4_ "8GXIN8(008JGE75'C+=NI10¬@9 C=DX#P2TY5
FC<79(CYH9/"7-?<:K@Y -)_0|I+(HH | "4_ "8GXIN8(008JGE75'C+=NI10¬@9 C=DX#P2TY5
FC<79(CYH9/"7-?<:K@Y -)_0|I+(HH | "4_ "8GXIN8(008JGE75'C+=NI10¬@9 C=DX#P2TY5
FC<68.C .9/"7-?<:K@Y -)_0|I+(HH | "F_ "IGXIN8(008JGE75'C+=NI10;#:AA65U#P2TY5
FC<52(D,H9/"7-?<./(Y J)_0|I+(HH | "4_#"IGXIN8(008JGE75'C+=NI10;9'AA"<U#P2TY5
FC<55( _.9/"7-?<:K@Y -)_0|I+(HH | "4* "IG0IN8(008JGE75'C+=NI10*@: I4DX#P2TY5
FC<43(E,.9/"7-?<:K(Y J)_0|I+(HH | "F_ "IGXIN8(008JGE75'C+=NI10;9' C2FU#P2TY5
FC<#7CE_H9/"7-?<:L@Y J)_0|I+(HH | "E_ "IGXIN8(008JGE75'C+=NI10*@: C=<U#P2TY5
FC|27IG H9/"7-?<:L@Y -)_0|I+(HH | "F_ "IGXIN8(008JGE75'C+=NI10;#:AA65U#P2TY5
FC|04FDZH9/"7-?<:K@Y -)_0|I+(HH | "F_ "IGXIN8(00 JGE75<C+=NI10¬@9 C=DX#P2TY5
FC|73IDZH9/"7-?<:K@Y J)*0|I+(HH | "4* "IG0IN8(008JGE75'C+=NI10;#: C=5X#P2TY5
FC|40<<,H9/"7-?<:K@Y -)_0|I+(HH | "F_ "IGXIN8(00.JGE75'C+=N810;#: C=AX#P2TY5
FC|40<<,H9S"F-?<:/(Y J)_0|I+(HH | "4_ "IGXIU8(008JGE75'C+=NI10?..Z1F5N#P2TY5
FC+21F( .:/"4-?<./(Y J)_0|I+(HH | "4_#"IGXIU8(X08JGE75'C+=NI10*#9Z1F5N#P2TY5
FC+21FC,H9/"7-?<./@Y -)_0|I+(HH | "F_ "IGXIN8@008JGE75'C+=NI10;#: C=5X#P2TY5
FC+25IE%.9/"7-;<:/(Y J)_0|I+(HH | "4_#"IGXIU8(X08JGE75'C+=NI10*#9Z1F5N#P2TY5
FC+71IGY.9/"7-?<./(Y J)_0|I+(HH | "F_ "IGXIN8(008JGE75'C+=NI10;9' C2AU#P2TY5
FC+58.E_H9/"7-?<:&@Y -)_0|I+(HH | "F_ "IGXIN8(008JGE75'C+=NI10$## C=DU#P2TY5
FC+47G(,.9/"7-?':/@Y J)*0|I+(HH | "4_#"IGXIU8(X08JGE75'C+=NI10;9' C3CX#P2TY5
FC+:4G(YH9/"7-?<./@Y -)_0|I+(HH | "4_ "8GXIN8(008JGE75'C+=NI10*#9 C=DX#P2TY5
FC+:4G(YH9/"7-?<:&@Y -)_0|I+(HH | "4_ "8GXIN8(008JGE75'C+=NI10*#9 C=DX#P2TY5
FC+:4G(YH:/"7-?<./(Y J)_0|I+(HH | "4¬ "IGXIN8(008JGE75'C+=NI10*#9 C=DX#P2TY5
FC+:6G<_.9/"7-?<:K@Y -)_0|I+(HH | "F_ "IGXIN8(008JGE75'C+=NI10;#:AA65X#P2TY5
FC+:7FF?.:/"7-?<./@Y -)_0|I+(HH | "F_ "IGXIU (00<JGE75'C+=N810;#:&F1 X#P2TY5
FCI20< >H:/"7-?<:&@Y J%_0|I+(HH | "4_ "8GXIN8(008JGE75'C+=NI10$## C=CX#P2TY5
FCI23GF>.9/"7-?<:K(R -)_0|I+(HH | "4_ "8GXIN8(008JGE75'C+=NI10¬@9 C=DX#P2TY5
```

```
IEF285I   GOSET.LJHEAKE                                          PASSED
IEF285I   VOL SER NOS= SYS02 .
IEF285I   SYS1.UT1                                               KEPT
IEF285I   VOL SER NOS= CAMP08.
IEF285I   SYS1.UT2                                               KEPT
IEF285I   VOL SER NOS= SYS02 .
IEF285I   SYS1.UT3                                               KEPT
IEF285I   VOL SER NOS= SYS03 .
IEF285I   SYS1.UT4                                               KEPT
IEF285I   VOL SER NOS= SYS02 .
IEF285I   SYSOUT                                                 SYSOUT
IEF285I   VOL SER NOS=       .
IEF285I   SYS1.UT5                                               KEPT
IEF285I   VOL SER NOS= SYS03 .
IEF285I   AAAAAAAA.AAAAAAAA.AAAAAAAA.AAAAAAAA.00000012 PASSED
IEF285I   VOL SER NOS= SYS03 .
IEF285I   X                                                      KEPT
IEF285I   VOL SER NOS= 2405  .
IEF285I   X                                                      DELETED
IEF285I   VOL SER NOS= LGL001.
IEF290I K OC2,2405  ,LJHEAKE
IEF285I   GOSET.LJHEAKE                                          DELETED
IEF285I   VOL SER NOS= SYS02 .
IEF285I   AAAAAAAA.AAAAAAAA.AAAAAAAA.AAAAAAAA.00000012 DELETED
IEF285I   VOL SER NOS= SYS03 .

H A S P  JOB STATISTICS --     171 CARDS READ --       590 LINES PRINTED --
         312 I/O CALLS              434 SVC CALLS          0.20 MINUTES CPU TM
```

```
FCI23GF>.9/"7-?<:K@Y -)_0|I+(HH |  "F_ "IGXIN8(008JGE75'C+=NI10¬@9 C=DX#P2TY5
FCI23GF>.9/"7-?<:K@Y -)_0|I+(HH |  "4_ "8GXIN8(008JGE75'C+=NI10¬@9 C=DX#P2TY5
FCI26<AS.9/"7-?<:&(Y J)_0|I+(HH |  "4_ "IGXIU8@008JGE75'C+=NI10;9' C=DU#P2TY5
FCI26<AS.9/"7-?<:&(Y J)_0|I+(HH |  "4_ "IGXIU8@008JGE75'C+=NI10;9' C=DU#P2TY5
FCI[8|F,.:/"7-?<./@Y -)_0|I+(HH |  "F_ "8GXIN8(008JGE75'C+=NI10;#: C=AX#P2TY5
FCI18|F,.:/"7-?<./@Y J)*0|I+(HH |  "4_ "8GXIN8(008JGE75'C+=NI10;#:&F1 X#P2TY5
FCI0!1<I.:/"7-?<:&@Y J)*0|I+(HH |  "F_ "IGXIN8(008JGE75'C+=NI10*@# C=DU#P2TY5
FCI0!1<I.:/"7-?<:&@Y J)*0|I+(HH |  "F_ "IGXIN8(008JGE75'C+=NI10*@# C=AU#P2TY5
FCI71GG%H9/"7-?<:&@Y J)*0|I+(HH |  "F_ "IGXIN8(008JGE75'C+=NI10;9' C=DU#P2TY5
FCI75(A,H:/"7-?<./(Y J)_0|I+(HH |  "4* "IGXIN8(008JGE75'C+=NI10;9' C2DU#P2TY5
FCI59(GZ.:/"7-?<:&(Y J)_0|I+(HH |  "4* "I6XIN8(008JGE75'C+=NI10;9' C6FU#P2TY5
FCI59(GZ.:/"7-?<./(Y J)_0|I+(HH |  "F_ "IGXIN8(008JGE75'C+=NI10;9' C6<U#P2TY5
FCI:3IG,H9/"7-?<:K@Y -)_0|I+(HH |  "4_ "8GXIN8(008JGE75'C+=NI10¬@9 C=DX#P2TY5
FCI:3IG,H9/"7-?<:K@Y -)_0|I+(HH |  "4_ "8GXIN8(008JGE75'C+=NI10¬@9 C=DX#P2TY5
FCI:3IG,H9/"7-?<:K@Y -)_0|I+(HH |  "4_ "8GXIN8(008JGE75'C+=NI10¬@9 C=DX#P2TY5
FCI:3IG,H9/"7-?<:K@Y -)_0|I+(HH |  "4_ "8GXIN8(008JGE75'C+=NI10¬@9 C=DX#P2TY5
FCI:9 B%.9/"7-?<:K@Y J%_0|I+(HH |  "4_ "8GXIN8(008JGE75'C+=NI10;9' C5 X#P2TY5
FCI:9 B%.9/"7-?<:K@Y J%_0|I+(HH |  "4_ "8GXIN8(008JGE75'C+=NI10;9' C5 X#P2TY5
FCI:9 B%.9/"7-?<:K@Y J%_0|I+(HH |  "4_ "8GXIN8(008JGE75'C+=NI10;9' C5 X#P2TY5
FCI:9 B%.9/"7-?<:L@Y J%_0|I+(HH |  "4_ "8GXIN8(008JGE75'C+=NI10;9' C5 X#P2TY5
FCI:9 B%.9/"7-?<:K@Y J%_0|I+(HH |  "4_ "8GXIN8(008JGE75'C+=NI10;9' C5 X#P2TY5
FCI:9 B%.9/"7-?<:K@Y J%_0|I+(HH |  "4_ "8GXIN8(008JGE75'C+=NI10;9' C5 X#P2TY5
FCI:9 B%.9/"7-?<:K@Y J%_0|I+(HH |  "4_ "8GXIN8(008JGE75'C+=NI10;9' C5 X#P2TY5
END OF SCRAMBLED FILE FOUND,      1050 CARDS READ
```

732	HOFFMAN	END	02:05	09/13	END	HOFFMAN	S
732	HOFFMAN	END	02:05	09/13	END	HOFFMAN	S
732	HOFFMAN	END	02:05	09/13	END	HOFFMAN	S
732	HOFFMAN	END	02:05	09/13	END	HOFFMAN	S
732	HOFFMAN	END	02:05	09/13	END	HOFFMAN	S
732	HOFFMAN	END	02:05	09/13	END	HOFFMAN	S
732	HOFFMAN	END	02:05	09/13	END	HOFFMAN	S
732	HOFFMAN	END	02:05	09/13	END	HOFFMAN	S
732	HOFFMAN	END	02:05	09/13	END	HOFFMAN	S
732	HOFFMAN	END	02:05	09/13	END	HOFFMAN	S
732	HOFFMAN	END	02:05	09/13	END	HOFFMAN	S
732	HOFFMAN	END	02:05	09/13	END	HOFFMAN	S
732	HOFFMAN	END	02:05	09/13	END	HOFFMAN	S
732	HOFFMAN	END	02:05	09/13	END	HOFFMAN	S
732	HOFFMAN	END	02:05	09/13	END	HOFFMAN	S
732	HOFFMAN	END	02:05	09/13	END	HOFFMAN	S
732	HOFFMAN	END	02:05	09/13	END	HOFFMAN	S
732	HOFFMAN	END	02:05	09/13	END	HOFFMAN	S
732	HOFFMAN	END	02:05	09/13	END	HOFFMAN	S
732	HOFFMAN	END	02:05	09/13	END	HOFFMAN	S
732	HOFFMAN	END	02:05	09/13	END	HOFFMAN	S
732	HOFFMAN	END	02:05	09/13	END	HOFFMAN	S
732	HOFFMAN	END	02:05	09/13	END	HOFFMAN	S
732	HOFFMAN	END	02:05	09/13	END	HOFFMAN	S
732	HOFFMAN	END	02:05	09/13	END	HOFFMAN	S
732	HOFFMAN	END	02:05	09/13	END	HOFFMAN	S
732	HOFFMAN	END	02:05	09/13	END	HOFFMAN	S
732	HOFFMAN	END	02:05	09/13	END	HOFFMAN	S
732	HOFFMAN	END	02:05	09/13	END	HOFFMAN	S
732	HOFFMAN	END	02:05	09/13	END	HOFFMAN	S
732	HOFFMAN	END	02:05	09/13	END	HOFFMAN	S
732	HOFFMAN	END	02:05	09/13	END	HOFFMAN	S
732	HOFFMAN	END	02:05	09/13	END	HOFFMAN	S
732	HOFFMAN	END	02:05	09/13	END	HOFFMAN	S
732	HOFFMAN	END	02:05	09/13	END	HOFFMAN	S
732	HOFFMAN	END	02:05	09/13	END	HOFFMAN	S
732	HOFFMAN	END	02:05	09/13	END	HOFFMAN	S
732	HOFFMAN	END	02:05	09/13	END	HOFFMAN	S
732	HOFFMAN	END	02:05	09/13	END	HOFFMAN	S
732	HOFFMAN	END	02:05	09/13	END	HOFFMAN	S
732	HOFFMAN	END	02:05	09/13	END	HOFFMAN	S
732	HOFFMAN	END	02:05	09/13	END	HOFFMAN	S
732	HOFFMAN	END	02:05	09/13	END	HOFFMAN	S
732	HOFFMAN	END	02:05	09/13	END	HOFFMAN	S
732	HOFFMAN	END	02:05	09/13	END	HOFFMAN	S
732	HOFFMAN	END	02:05	09/13	END	HOFFMAN	S
732	HOFFMAN	END	02:05	09/13	END	HOFFMAN	S
732	HOFFMAN	END	02:05	09/13	END	HOFFMAN	S
732	HOFFMAN	END	02:05	09/13	END	HOFFMAN	S
732	HOFFMAN	END	02:05	09/13	END	HOFFMAN	S
732	HOFFMAN	END	02:05	09/13	END	HOFFMAN	S
732	HOFFMAN	END	02:05	09/13	END	HOFFMAN	S
732	HOFFMAN	END	02:05	09/13	END	HOFFMAN	S
732	HOFFMAN	END	02:05	09/13	END	HOFFMAN	S
732	HOFFMAN	END	02:05	09/13	END	HOFFMAN	S
732	HOFFMAN	END	02:05	09/13	END	HOFFMAN	S
732	HOFFMAN	END	02:05	09/13	END	HOFFMAN	S
732	HOFFMAN	END	02:05	09/13	END	HOFFMAN	S
732	HOFFMAN	END	02:05	09/13	END	HOFFMAN	S
732	HOFFMAN	END	02:05	09/13	END	HOFFMAN	S
732	HOFFMAN	END	02:05	09/13	END	HOFFMAN	S