UR1982-03

1982/3. FORTRAN programs for the transfer of data to or from a Perkin-Elmer 6/16 minicomputer.

R.G. Richardson

Abstract

These programs transfer data from one logical unit to another by blocking or unblocking data and translating characters to maintain compatibility with the EBCDIC character set. Data transfer is then possible via magnetic tape.

THE PROGRAMS

EB2ASS (Appendix 1)

This program generates an EBCDIC to ASCII translation table by prompting with decimal EBCDIC character values on logical unit 5 and reading the corresponding decimal ASCII character value. The translation table (Appendix 2) is created on logical unit 6 with each record consisting of the EBCDIC decimal value, the ASCII decimal value, and the character.

EBASDT (Appendix 3) and ASEBDT (Appendix 4)

These programs generate FORTRAN DATA statements from the translation table set up by EB2ASS and implement a look-up table for translation from EBCDIC to ASCII (EBASDT) or from ASCII to EBCDIC (ASEBDT). In addition, ASEBDT generates an ASCII to EBCDIC translation table (Appendix 5).

TPTRAN (Appendix 6)

This program transfers data from logical unit 6 to logical unit 3 until a filemark is encountered. Logical unit 6 must be capable of supporting unformatted input. Data may be unblocked with this program.

The input and output recordlengths may differ and truncation or padding with spaces is used for any adjustment. Translation from EBCDIC to ASCII is optional.

Input from logical unit 5 is:

NREC - format I5
- the input recordlength in characters (must be even)

NBLK - format I5
- the input blocksize in characters (must be a multiple of NREC)
- NBLK <10001

NOUT - format I5
- the output recordlength in characters (must be even)
- NOUT <255

Y or N - format A1
- Y if EBCDIC to ASCII translation is required, N otherwise

Y or N - format A1
- Y if output is to a binary device such as disk or tape, N otherwise

The only non-standard subroutine used is IBYTE (A, N) which is an integer function returning the value of the Nth byte in A. N has values 1, 2, 3, - where N = 1 is the most significant byte of the first word of A.

DSKTAP (Appendix 7)

This program transfers data from one or more disk files to logical unit 6 which must be capable of supporting unformatted output. Data may be blocked before output if desired.

The input and output recordlengths may differ, and truncation or padding with spaces is used for any adjustment. Translation from ASCII to EBCDIC is optional.

Input from logical unit 5 is:

NREC - format I5
- the output recordlength in characters (must be even)

NBLK - format I5
- the output blocksize in characters (must be a multiple of NREC)
- NBLK <10001

For each file:

FNAME - format A6
- the file name

DISK - format A2
- the disk drive containing the file

NIN - format I5
- the input recordlength in characters (must be even)
- NIN <255

Y or N - format A1
- Y if ASCII to EBCDIC translation required, N otherwise

Y or N - format A1
- Y if more files to transfer, N otherwise

Non-standard subroutines are:

ACTIV8 (LU, INLU, PROMPT, IATT) - a filename and disk drive number are read from logical unit INLU and that file is activated as logical unit LU write protected if IATT = 1, write enabled if IATT = 0. Prompts are output on logical unit PROMPT.

SBYTE (A, N, K) - this routine sets the Nth byte of A to the value of the least significant byte of K. N has values 1, 2, 3, -- where N = 1 is the most significant byte of the first word of A.

[19 February 1982]

APPENDIX 1

Program EB2ASS

```
$TITL  EB2ASS - EBCDIC TO ASCII TRANSLATION TABLE CREATION
C CREATES FILE EB2ASC.  ALTHOUGH PROBABLY USED ONLY ONCE
C THE TRANSLATE TABLE IS IN A SUITABLE FORM FOR INPUT TO
C THIS PROGRAM FOR FURTHER RUNS.
      WRITE(6,100)
  100 FORMAT(' EBCDIC TO ASCII TRANSLATION TABLE (EBCDIC,ASCII,CHAR)')
      K=32
      L=K*256
      DO 10 I=1,74
      J=I-1
      WRITE(6,101) J,K,L
  101 FORMAT(2I4,1X,A1)
   10 CONTINUE
C DONE 0-73
      DO 11 I=74,169
      WRITE(5,199) I
  199 FORMAT(I4)
      READ(5,200) J
  200 FORMAT(I3)
      IF (J .EQ. 0) J=32
C SPACE NEED NOT BE FED IN FULL
      L=J*256
      WRITE(6,101) I,J,L
   11 CONTINUE
C READ ALL THESE FROM KEYBOARD
      J=32
      L=J*256
      DO 12 I=170,191
      WRITE(6,101) I,J,L
   12 CONTINUE
C A FEW MORE SPACES
      DO 14 I=192,255
      WRITE(5,199) I
      READ(5,200) J
      IF (J .EQ. 0) J=32
      L=J*256
      WRITE(6,101) I,J,L
   14 CONTINUE
C SHOULD ALL BE OUT NOW
      ENDFILE 6
      STOP
      END
```

5/20

APPENDIX 2

EBCDIC to ASCII translation table

EBCDIC TO ASCII TRANS. TABLE

0	32	59	32		119	32		179	32		221	32			
1	32	60	32		120	32		180	32		222	32			
2	32	61	32		121	96	`	181	32		223	32			
3	32	62	32		122	58	:	182	32		224	92	\		
4	32	63	32		123	35	#	183	32		225	32			
5	32	64	32		124	64	@	184	32		226	83	S		
6	32	65	32		125	39	'	185	32		227	84	T		
7	32	66	32		126	61	=	186	32		228	85	U		
8	32	67	32		127	34	"	187	32		229	86	V		
9	32	68	32		128	32		188	32		230	87	W		
10	32	69	32		129	97	a	189	32		231	88	X		
11	32	70	32		130	98	b	190	32		232	89	Y		
12	32	71	32		131	99	c	191	32		233	90	Z		
13	32	72	32		132	100	d	192	123	{	234	32			
14	32	73	32		133	101	e	193	65	A	235	32			
15	32	74	91	[	134	102	f	194	66	B	236	32			
16	32	75	46	.	135	103	g	195	67	C	237	32			
17	32	76	60	<	136	104	h	196	68	D	238	32			
18	32	77	40	(	137	105	i	197	69	E	239	32			
19	32	78	43	+	138	32		198	70	F	240	48	0		
20	32	79	33	!	139	32		199	71	G	241	49	1		
21	32	80	38	&	140	32		200	72	H	242	50	2		
22	32	81	32		141	32		201	73	I	243	51	3		
23	32	82	32		142	32		202	32		244	52	4		
24	32	83	32		143	32		203	32		245	53	5		
25	32	84	32		144	32		204	32		246	54	6		
26	32	85	32		145	106	j	205	32		247	55	7		
27	32	86	32		146	107	k	206	32		248	56	8		
28	32	87	32		147	108	l	207	32		249	57	9		
29	32	88	32		148	109	m	208	125	}	250	32			
30	32	89	32		149	110	n	209	74	J	251	32			
31	32	90	93	]	150	111	o	210	75	K	252	32			
32	32	91	36	$	151	112	p	211	76	L	253	32			
33	32	92	42	*	152	113	q	212	77	M	254	32			
34	32	93	41	)	153	114	r	213	78	N	255	32			
35	32	94	59	;	154	32		214	79	O					
36	32	95	94	↑	155	32		215	80	P					
37	32	96	45	-	156	32		216	81	Q					
38	32	97	47	/	157	32		217	82	R					
39	32	98	32		158	32		218	32						
40	32	99	32		159	32		219	32						
41	32	100	32		160	32		220	32						
42	32	101	32		161	126	~								
43	32	102	32		162	115	s								
44	32	103	32		163	116	t								
45	32	104	32		164	117	u								
46	32	105	32		165	118	v								
47	32	106	124	¦	166	119	w								
48	32	107	44	,	167	120	x								
49	32	108	37	%	168	121	y								
50	32	109	95	_	169	122	z								
51	32	110	62	>	170	32									
52	32	111	63	?	171	32									
53	32	112	32		172	32									
54	32	113	32		173	32									
55	32	114	32		174	32									
56	32	115	32		175	32									
57	32	116	32		176	32									
58	32	117	32		177	32									
		118	32		178	32									

7/20

APPENDIX 3

Program EBASDT

8/20

```
$TITL  EBASDT - CREATE EBCDIC TO ASCII TRANSLATE DATA STATEMENT
C MAY REQUIRE MODIFICATION FOR USE WITH SOME COMPILERS TO
C PRODUCE THE APPROPRIATE FORMAT
      INTEGER*2 ITAB(256),I
C 256 AS HAVE VALUES 0 TO 255
      REWIND 3
      READ(3,100) I
  100 FORMAT(A1)
C SKIP THE FIRST LINE
      READ(3,101) ITAB
  101 FORMAT(4X,I4)
C READ THE ASCII DECIMAL VALUES
      WRITE(6,200)
  200 FORMAT(6X,'DATA ITRANS/')
      WRITE(6,201) (ITAB(I),I=1,255)
  201 FORMAT(16(5X,'.  ',16(I3,',')/))
      WRITE(6,202) ITAB(256)
  202 FORMAT(5X,'.  ',I3,'/')
      ENDFILE 6
      STOP
      END
```

APPENDIX 4

Program ASEBDT

```
$TITL   ASEBDT - CREATE ASCII TO EBCDIC TRANSLATE TABLE
C MAY REQUIRE MODIFICATION WHEN TRANSFERRED TO OTHER MACHINES
      INTEGER*2 EBTAB(256),ITRANS(128)
C 256 AS HAVE 0 TO 255, 128 AS HAVE 0 TO 127
C ASSUME 7 BIT ASCII FOR TRANSLATE - IF DOUBTS ARE HELD LOGICAL AND
C TO ENSURE 7 BITS
      REWIND 3
      READ(3,100) I
  100 FORMAT(A1)
C SKIP THE FIRST LINE IN THE FILE OF TRANSLATE EQUIVALENTS SET UP
C FOR EBCDIC TO ASCII TRANSLATE
      READ(3,101) EBTAB
  101 FORMAT(4X,I4)
C READ THE ASCII DECIMAL VALUES
      DO 10 I=1,33
      ITRANS(I)=64
   10 CONTINUE
C FIRST 33 CHARACTERS ALL TRANSLATE TO EBCDIC SPACE
      DO 12 I=33,127
      DO 14 J=64,256
      IF (EBTAB(J) .EQ. I) GOTO 16
   14 CONTINUE
      J=65
   16 ITRANS(I+1)=J-1
C FILL WITH EITHER BLANK OR THE CHARACTER
   12 CONTINUE
      WRITE(4,200)
  200 FORMAT(6X,'DATA ITRANS/')
      WRITE(4,201) (ITRANS(I),I=1,127)
  201 FORMAT(8(5X,'  ',16(I3,',')/))
      WRITE(4,202) ITRANS(128)
  202 FORMAT(5X,'  ',I3,'/')
      ENDFILE 4
      WRITE(6,300)
  300 FORMAT(' ASCII TO EBCDIC TRANSLATE TABLE (ASC,EBC,CHAR)')
      DO 20 I=1,128
      J=I-1
      K=ITRANS(I)
C TO EBCDIC
      K=EBTAB(K+1)*256
C BACK TO ASCII FOR CHECK AND UP TO A1 FORMAT
      WRITE(6,301) J,ITRANS(I),K
  301 FORMAT(2I4,1X,A1)
   20 CONTINUE
      STOP
      END
```

APPENDIX 5

ASCII to EBCDIC translation table

12/20

ASCII TO EBCDIC TRANSLATE TABLE (ASC, EBC, CHAR)

0	64		59	94	;	119	166	w
1	64		60	76	<	120	167	x
2	64		61	126	=	121	168	y
3	64		62	110	>	122	169	z
4	64		63	111	?	123	192	{
5	64		64	124	@	124	106	|
6	64		65	193	A	125	208	}
7	64		66	194	B	126	161	~
8	64		67	195	C	127	64	
9	64		68	196	D			
10	64		69	197	E			
11	64		70	198	F			
12	64		71	199	G			
13	64		72	200	H			
14	64		73	201	I			
15	64		74	209	J			
16	64		75	210	K			
17	64		76	211	L			
18	64		77	212	M			
19	64		78	213	N			
20	64		79	214	O			
21	64		80	215	P			
22	64		81	216	Q			
23	64		82	217	R			
24	64		83	226	S			
25	64		84	227	T			
26	64		85	228	U			
27	64		86	229	V			
28	64		87	230	W			
29	64		88	231	X			
30	64		89	232	Y			
31	64		90	233	Z			
32	64		91	74	[			
33	79	!	92	224	\			
34	127	"	93	90	]			
35	123	#	94	95	↑			
36	91	$	95	109	_			
37	108	%	96	121	`			
38	80	&	97	129	a			
39	125	'	98	130	b			
40	77	(	99	131	c			
41	93	)	100	132	d			
42	92	*	101	133	e			
43	78	+	102	134	f			
44	107	,	103	135	g			
45	96	-	104	136	h			
46	75	.	105	137	i			
47	97	/	106	145	j			
48	240	0	107	146	k			
49	241	1	108	147	l			
50	242	2	109	148	m			
51	243	3	110	149	n			
52	244	4	111	150	o			
53	245	5	112	151	p			
54	246	6	113	152	q			
55	247	7	114	153	r			
56	248	8	115	162	s			
57	249	9	116	163	t			
58	122	:	117	164	u			
			118	165	v			

APPENDIX 6

Program TPTRAN

```
$TITL  TPTRAN - FOR TRANSFERRING FILES FROM LU6 TO LU3
C FILES MAY BE IN ASCII OR EBCDIC, BLOCKED OR UNBLOCKED.
C INPUT IS LU 6
C OUTPUT IS LU 3
C CONTROL IS LU 5 (THE CONSOLE)
      INTEGER*2 IIN(5000),IOUT(127),NREC,NBLK,NOUT,NTOP,NBLK2,NREC2,
     .          ITMP1,IINK,ITMP2,NOUT2,TTRANS(256),
     .          ITRANS(256),IEND,IER,ITMP,AYES,YES,ISPACE
      LOGICAL EBCDIC,DISK
      DATA ISPACE,YES/2H  ,1HY/
      DATA ITRANS/
     .  32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32,
     .  32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32,
     .  32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32,
     .  32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32,
     .  32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 91, 46, 60, 40, 43, 33,
     .  38, 32, 32, 32, 32, 32, 32, 32, 32, 32, 93, 36, 42, 41, 59, 94,
     .  45, 47, 32, 32, 32, 32, 32, 32, 32, 32,124, 44, 37, 95, 62, 63,
     .  32, 32, 32, 32, 32, 32, 32, 32, 32, 96, 58, 35, 64, 39, 61, 34,
     .  32, 97, 98, 99,100,101,102,103,104,105, 32, 32, 32, 32, 32, 32,
     .  32,106,107,108,109,110,111,112,113,114, 32, 32, 32, 32, 32, 32,
     .  32,126,115,116,117,118,119,120,121,122, 32, 32, 32, 32, 32, 32,
     .  32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32,
     . 123, 65, 66, 67, 68, 69, 70, 71, 72, 73, 32, 32, 32, 32, 32, 32,
     . 125, 74, 75, 76, 77, 78, 79, 80, 81, 82, 32, 32, 32, 32, 32, 32,
     .  92, 32, 83, 84, 85, 86, 87, 88, 89, 90, 32, 32, 32, 32, 32, 32,
     .  48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 32, 32, 32, 32, 32,
     .  32/
C END OF LOW BYTE TRANSLATE TABLE
C
C NOW TO CREATE A HIGH BYTE TABLE
      DO 3 I=1,256
    3 TTRANS(I)=ITRANS(I)*256
C THIS SAVES LATER MULTIPLIES
C ASSUME INPUT BLOCKSIZE < 10001 CHARS, OUTPUT RECORDLENGTH < 255 CHARS
    5 WRITE(5,100)
  100 FORMAT(' RECORDSIZE? - MUST BE N*2'/' RRRR CHAR')
      READ(5,101) NREC
  101 FORMAT(I5)
      IF (MOD(NREC,2) .NE. 0) GOTO 5
C CHECK FOR EVEN NUMBER OF CHARACTERS IN RECORD
    6 WRITE(5,102)
  102 FORMAT(' BLOCKSIZE?'/' BBBB CHAR')
      READ(5,101) NBLK
      IF (MOD(NBLK,NREC) .NE. 0) GOTO 6
C ENSURE THAT INPUT BLOCKS ARE MULTIPLES OF THE RECORD SIZE
C GET THE RECORDSIZE AND BLOCKSIZE
    7 WRITE(5,103)
  103 FORMAT(' OUTPUT RECORDSIZE?'/' OOOO CHAR')
      READ(5,101) NOUT
      IF (MOD(NOUT,2) .NE. 0) GOTO 7
C CHECK THAT IT IS MULTIPLE OF 2
      NTOP=MIN0(NREC,NOUT)/2
C DEALING WITH N*2 RECORDS SO CONVERT FROM CHARACTERS TO WORDS
C IF NOUT > NREC WILL NEED TO PAD WITH SPACES LATER
      WRITE(5,104)
  104 FORMAT(' EBCDIC TO ASCII TRANSLATION?  Y OR N')
      READ(5,105) AYES
  105 FORMAT(A1)
      EBCDIC=AYES .EQ. YES
```

```
C TRANSLATION IS NEEDED
      WRITE(5,107)
  107 FORMAT(' OUTPUT TO EITHER DISK OR TAPE?  Y OR N')
      READ(5,105) AYES
      DISK=AYES .EQ. YES
C FOR DISK OR TAPE ONLY NEED UNFORMATTED WRITE OPERATION
      NBLK2=NBLK/2
      NREC2=NREC/2
      NOUT2=NOUT/2
   10 CALL SLURP(IIN,NBLK2,IEND,IER,6)
C BINARY READ WITH CHECKS FOR END AND ERROR
      IF (IEND .NE. 0) GOTO 20
C END OF INPUT
      IF (IER .NE. 0) GOTO 25
C SOME SORT OF ERROR ON INPUT FILE
      DO 11 I=1,NBLK2,NREC2
      K=I
      DO 15 J=1,NTOP
      IF (EBCDIC) GOTO 22
C FOR ASCII NEED ONLY STRAIGHT COPY
      IOUT(J)=IIN(K)
      GOTO 23
C
C OTHERWISE NEED TO GET BYTES AND TRANSLATE
   22 IINK=IIN(K)
      ITMP1=IBYTE(IINK,1)
      ITMP2=IBYTE(IINK,2)
      IOUT(J)=TTRANS(ITMP1+1)+ITRANS(ITMP2+1)
   23 K=K+1
   15 CONTINUE
C DETACH INDIVIDUAL RECORDS FROM BLOCK WITH TRANSLATION IF DESIRED
      IF (NTOP .EQ. NOUT2) GOTO 16
C DONT NEED TO PAD
      DO 17 J=NTOP,NOUT2
      IOUT(J)=ISPACE
   17 CONTINUE
C HAVING GOT SPACES WHERE NEEDED
C
C CAN NOW START OUTPUT
   16 IF (DISK) CALL OUTBIN(IOUT,NOUT2,3)
      IF (.NOT. DISK) CALL PUTOUT(IOUT,NOUT2,3)
C OUTPUT THROUGH A SUBROUTINE
   11 CONTINUE
      GOTO 10
C BACK ROUND FOR THE NEXT BLOCK
C
C IF AT THE END TIDY UP
   20 CONTINUE
      ENDFILE 3
      STOP
C
C IF AN ERROR
   25 WRITE(5,106)
  106 FORMAT(' CHECK BLOCKSIZE AND TAPE')
      STOP
      END
      SUBROUTINE OUTBIN(IOUT,NOUT,ILUN)
      INTEGER*2 IOUT(NOUT),NOUT,ILUN
      WRITE(ILUN) IOUT
      RETURN
```

```
C BINARY WRITE ONLY FOR DISK OR TAPE
      END
      SUBROUTINE PUTOUT(IOUT,NOUT,ILUN)
      INTEGER*2 IOUT(NOUT),ILUN,NOUT
      WRITE(ILUN,100) IOUT
  100 FORMAT(65A2)
C WRITE OUT AT UP TO 130 CHARACTERS PER RECORD
      RETURN
      END
      SUBROUTINE SLURP(IIN,NWORD,IEND,IERR,ILUN)
      INTEGER*2 IIN(NWORD),NWORD,IEND,IERR
      IEND=0
      IERR=0
      READ(ILUN,END=10,ERR=20) IIN
      RETURN
   10 IEND=1
      RETURN
   20 IERR=1
      RETURN
      END
```

17/20

APPENDIX 7

Program DSKTAP

```
$TITL  DSKTAP - TRANSFER FILES FROM DISK TO TAPE OR DISK
C OUTPUT FILES MAY BE IN ASCII OR EBCDIC, BLOCKED OR UNBLOCKED
C OUTPUT IS LU 6
C INPUT IS LU 3, A DISK FILE WITH A DYNAMICALLY SET FILENAME
C CONTROL IS LU 5, THE CONSOLE
      INTEGER*2 IIN(127),IOUT(5000),ITRANS(128),ASPACE,YES,ESPACE,
     . NREC,NREC2,NBLK,NBLK2,AYES,NIN,NIN2,NTOP,COUNT,SPACE,IEND,
     . IER,I,IINI,IOUTI,NO,ITMP1,ITMP2
      LOGICAL EBCDIC
      DATA ASPACE,YES,ESPACE,NO/2H  ,1HY,10280,1HN/
C ASCII AND EBCDIC SPACE
      DATA ITRANS/
     .  64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64,
     .  64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64,
     .  64, 79,127,123, 91,108, 80,125, 77, 93, 92, 78,107, 96, 75, 97,
     . 240,241,242,243,244,245,246,247,248,249,122, 94, 76,126,110,111,
     . 124,193,194,195,196,197,198,199,200,201,209,210,211,212,213,214,
     . 215,216,217,226,227,228,229,230,231,232,233, 74,224, 90, 95,109,
     . 121,129,130,131,132,133,134,135,136,137,145,146,147,148,149,150,
     . 151,152,153,162,163,164,165,166,167,168,169,192,106,208,161,
     .  64/
C TRANSLATE TABLE
C
C INPUT RECODRSIZE<255 CHAR,OUTPUT BLOCKSIZE<10001 CHAR
C ALL BLOCKSIZE MUST BE N*2 CHARACTERS
    5 WRITE(5,100)
  100 FORMAT(' OUTPUT RECORDSIZE?'/' 0000 CHAR')
      READ(5,101) NREC
  101 FORMAT(I5)
      NREC2=NREC/2
      IF (NREC .NE. NREC2*2) GOTO 5
C MUST BE AN EVEN NUMBER
    6 WRITE(5,102)
  102 FORMAT(' BLOCKSIZE?'/' BBBB CHARS')
      READ(5,101) NBLK
      IF (MOD(NBLK,NREC) .NE. 0) GOTO 6
C OUTPUT BLOCKSIZE MUST BE A MULTIPLE OF THE RECORDSIZE
      NBLK2=NBLK/2
C NOW KNOW ALL ABOUT THE OUTPUT FILE
C
      COUNT=0
C SET WORD COUNT FOR OUTPUT BLOCK
C
C NOW SET UP INPUT FILE
   23 CALL ACTIV8(3,5,5,1)
C INPUT FILE READ ON 5, SETUP AS UNIT 3 READONLY
    7 WRITE(5,105)
  105 FORMAT(' INPUT RECORDSIZE?'/' IIII CHAR')
      READ(5,101) NIN
      NIN2=NIN/2
      IF (NIN .NE. NIN2*2) GOTO 7
C CHECK IS N*2
C NOW GOT ALL RECORDSIZES
      WRITE(5,103)
  103 FORMAT(' ASCII TO EBCDIC TRANSLATE? Y OR N')
      READ(5,104) AYES
  104 FORMAT(A1)
      EBCDIC=AYES .EQ. YES
C TRANSLATE??
      SPACE=ASPACE
```

```
      IF (EBCDIC) SPACE=ESPACE
C GET THE RIGHT SPACE CHARACTERS IF USING TRANSLATE
      NTOP=MIN0(NREC2,NIN2)
C FIND THE SMALLEST RECORD LENGTH TO ALLOW CORRECTION IF THEY ARE NOT
C THE SAME
   27 CALL SLURP(IIN,NIN2,IEND,IER,3)
      IF (IEND .NE. 0) GOTO 8
C AT END OF INPUT FILE
      IF (IER .NE. 0) GOTO 25
C AN ERROR ON INPUT IS MOST UNLIKELY
      DO 15 I=1,NTOP
      COUNT=COUNT+1
      IF (EBCDIC) GOTO 22
C FOR ASCII NEED ONLY STRAIGHT COPY
      IOUT(COUNT)=IIN(I)
      GOTO 15
C
C OTHERWISE GET BYTES ( 7 BIT ASCII!!!!!!!!!!) AND TRANSLATE
   22 IINI=IIN(I)
      ITMP1=IINI/256
C MOST SIGNIFICANT BYTE
      ITMP2=IINI-ITMP1*256
C LEAST SIG BYTE
      CALL SBYTE(IOUTI,1,ITRANS(ITMP1+1))
      CALL SBYTE(IOUTI,2,ITRANS(ITMP2+1))
      IOUT(COUNT)=IOUTI
   15 CONTINUE
C GO ROUND TILL EXHAUST INPUT OR OUTPUT RECORDLENGTH
      IF (NTOP .EQ. NREC2) GOTO 26
C DONT NEED TO PAD TO FILL RECORD
      ITMP1=NTOP+1
      DO 17 I=ITMP1,NREC2
      COUNT=COUNT+1
      IOUT(COUNT)=SPACE
   17 CONTINUE
C PAD WITH BLANKS
   26 IF (COUNT .EQ. NBLK2) CALL OUTBIN(IOUT,NBLK2,6,COUNT)
C IF FILLED A BLOCK OUTPUT IT
      GOTO 27
C BACK ROUND FOR MORE
    8 WRITE(5,106)
  106 FORMAT(' ANY MORE FILES?')
      READ(5,104) AYES
      IF (AYES .EQ. YES) GOTO 23
C ROUND FOR ANOTHER
      IF (AYES .NE. NO) GOTO 8
C AT THIS POINT OUTPUT ANY PARTLY FILLED BLOCK
      IF (COUNT .EQ. 0) GOTO 24
C NO PARTLY FILLED BLOCK
      COUNT=COUNT+1
      DO 20 I=COUNT,NBLK2
      IOUT(I)=SPACE
   20 CONTINUE
      CALL OUTBIN(IOUT,NBLK2,6,COUNT)
   24 ENDFILE 6
      ENDFILE 6
      CALL SYSCOM(I,'AS 0320*')
      STOP
   25 WRITE(5,107)
  107 FORMAT(' SOME SORT OF ERROR ON INPUT - TERMINATED')
```

```
      ENDFILE 6
      STOP
      END
      SUBROUTINE OUTBIN(IOUT,NOUT,ILUN,COUNT)
      INTEGER*2 IOUT(NOUT),NOUT,ILUN,COUNT
      COUNT=0
      WRITE(ILUN) IOUT
      RETURN
      END
      SUBROUTINE SLURP(IIN,NWORD,IEND,IER,ILUN)
      INTEGER*2 IIN(NWORD),NWORD,IEND,IER,ILUN
      IEND=0
      IER=0
      READ(ILUN,END=10,ERR=20) IIN
      RETURN
10    IEND=1
      RETURN
20    IER=1
      RETURN
      END
```