DECODEDKU

DECODE PACKAGE
PDP-15
ROGER BRANDT
08/01-70
GENERAL DESCRIPTION

LANGUAGE:

MACRO-15.

PURPOSE:

TO DO FORMAT CONVERSION OF DATA IN ASCII-REPRESENTATION IN MEMORY TO INTERNAL BINARY REPRESENTATION.
DECODE CAN HANDLE I,F,A,R,O AND X -FORMATS AND PERMITS THE PROGRAMMER TO KEEP CONTROL OF THE CALLING PROGRAM AFTER INVALID CHARACTERS ARE DETECTED IN I,F AND O -FORMAT.
THE DECODE-PACKAGE CONSISTS OF THE FOLLOWING ROUTINES:
DECODE,AFMT,AFMT1,FFMT,IBASE,IDCPNT,IFMT,IOFMT,IRFMT, KFMT,XFMT,IFMTX,IFDEL AND SETDEL

USE:

1.
INPUT:
THE INFORMATION MAY HAVE TO BE TRANSMITTED TO THE MEMORY BY STATEMENTS SUCH AS:

```
      READ (INUNIT,10) ARRAY
   10 FORMAT (16A5)
```

2.
CALLING SEQUENCE:
INITIATE THE CONVERSION WITH ONE OF THE FOLLOWING CALLS:

A) CALL DECODE (ARRAY(1))

WHICH SETS A POINTER TO THE FIRST CHARACTER IN THE DOUBLEWORD A(1).

B) CALL DECEOR(ARRAY(1))

THIS CALL SETS A POINTER TO THE FIRST CHARACTER IN DOUBLE-WORD ARRAY(1), IT ALSO ALLOWS THE PROGRAMMER TO DETECT THE OCCUREN(OF 'CR' AND 'ALT MOD'. IF ONE OF THESE TWO CHARACTERS IS FOUND,THE (END OF RECORD) .EOR.-FLAG IS CHANGED FROM -1 TO 0.
CONVERT ONE VARIABLE AT A TIME WITH ONE OF THE FOLLOWING CALLS

```
I=IFMT(IW)              I-FORMAT
F=FFMT(IW)              F-FORMAT
A=AFMT(IW)              A-FORMAT
CALL AFMT1(A,IW)        A-FORMAT
I=IRFMT(DUMMY)          R1-FORMAT
J=IOFMT(IW)             O-FORMAT
CALL XFMT(IW)           X-FORMAT
```

WHERE IW IS AN INTEGER CONSTANT,VARIABLE OR EXPRESION SPECIFYING THE FIELDWIDTH. FOR EACH CALL THE DECODE-POINTER WILL BE MOVED IW POSITIONS FORWARD EXEPT IN XFMT WHERE A NEG. ARGUMENT WILL MOVE IT IW POSITIONS BACKWARD.
.EJECT

ERROR CHECKING:

ROUTINES IFMT,IFMTX,IOFMT AND FFMT DETECT ILLEGAL CHARACTERS.

NAME	LEGAL CHARACTERS
IFMT	+-1234567890
IFMTX	-//-
IOFMT	+-12345670
FFMT	+-.1234567890

```
        ALL     SPACE
ALL OTHER CHARACTERS ARE CONSIDERED ILLEGAL.
        BY CALLING THE FUNCTION KFMT AN ERROR TEST CAN BE DONE:
        K=KFMT(DUMMY)
        GIVES  K=-1 IF NO ERROR IN LAST CONVERSION
               K= 0 IF ERROR IN LAST CONVERSION
CALLING KFMT SETS THE ERROR INDICATOR IN DECODE TO -1.
. (DUMMY ABOVE IS A DUMMY-ARGUMENT)

ALARMS AND PRINTOUTS:
IF DATA ERROR HAS APPEARED IN IFMT,IOFMT OR FFMT AND THE
PROGRAMMER HAS NOT TESTED THE KFMT-INDICATOR THE PROGRAM
WILL BE TERMINATED AT THE NEXT CALL TO ANY OF THE ROUTINES
IN THE DECODE-PACKAGE AND AN ERROR MESSAGE IS PRINTED.

PROGRAM ABORT:
1.
IF IW (FIELD-WIDTH) IS NEGATIVE IN CALL TO IFMT,IOFMT,FFMT,AFMT
OR AFMT1.
2.
IF IB (NEW BASE) IN IFMT IS LESS THAN 2 OR IB IS GREATER
THAN 10.
.EJECT
                                                    DECODE PACKAGE

SPECIAL NOTES:
1.
BY CALLING IBASE THE PROGRAMMER CAN CHANGE THE BASE TO ANY
VALUE FROM 2 TO 10. THE NEW BASE IS USED BY IFMT UNTIL
NEXT CALL TO IBASE.
        CALL IBASE (IB)
ERROR CHECKING IS DONE AS FOR IFMT EXEPT THAT AFTER A
'CALL IBASE(IB)' ONLY DIGITS 0 TO IB-1 ,+,- AND SPACE ARE LEGAL
CHARACTERS.
2.
NEGATIVE ARGUMENT IN XFMT MOVES THE POINTER BACKWARD, ZERO
ARGUMENT CALL ACTS AS DO NOTHING
3.
THE RESULTING VALUE AFTER CALLING IFMT,IOFMT OR FFMT
DEPENDS ON THE LAST SIGN IN THE ASCII-INPUT.
SPACES ARE NOT CONVERTED TO ZERO EVEN IF THEY APPEAR IN THE
MIDDLE OF A NUMBER.

EX.
        INPUT   INTERPRETED AS
        1-9 0           -190
        +1-1+ 0         110
        +1.-1           -1.1
        +1.1.2-         ?
NO ERRORS WILL BE DETECTED IN THE ABOVE EX.
4.
IF DECEOR WAS CALLED THE PROGRAMMER CAN TEST FOR 'CR' OR 'ALT MOD'
WITH:
        K=IFEOR(DUMMY)
        K=-1 NO 'CR' OR ' ALT MOD'
        K= 0 YES, 'CR' OR 'ALTMOD'
THIS CALL ALSO CLEARS THE .EOR. FLAG (END OF RECORD FLAG).
5.
IF ARGUMENT=0
```

```
          J=IFMT(0)                 J=0
          AJ=AFMT(0)                AJ=5H
          CALL AFMT1(AJ,0)          AJ=5H
          J=IOFMT(0)                J=0
          F=FFMT(0)                 F=0.0
```

6.
USE OF AFMT1 DOES NOT REQUIRE REAL PACKAGE.

.EJECT

DECODE PACKAGE

SAVE,RESTORE DECODE POINTER:
THE PROGRAMMER CAN SAVE THE DECODE POINTER BY THE FOLLOWING CALL:

```
          IPEK=IDCPNT(DUMMY)
```

AFTER THIS CALL IPEK CONTAINS:

```
          BIT  0-14   G.WP=WORDPOINTER
              15-17   G.BP=CHARACTERPOINTER
```

TO RESTORE THE DECODE POINTER USE:

```
          CALL DECRES(IPEK)
```

7.
IFMTX PROVIDES THE PROGRAMMER WITH FREE-FORMAT INTEGER-
CONVERSION. IFMTX CONVERTES CHARACTERS UP TO THE
FIRST RECOGNIZED DELIMITER.
BEFORE FIRST CALL TO IFMTX THE USER MUST CALL
SETDEL TO SET UP DELIMITERS FOR IFMTX.

EX

```
C         PROGRAM IFMTX-TEST
          DIMENSION IDEL(2),R(17),INT(5)
C         THIS PROGRAM READS 5 INTEGERS FROM UNIT 1
C         AND STOPS.
C         DELIMITERS USED ARE COMMA AND CARRIAGE-RETURN,
C         COMMA=44 IN 5/7-ASCII
C         CARRIAGE RETURN=13 IN 5/7-ASCII
          DATA IDEL(1)/13/,IDEL(2)/44/
C         SETDEL IS USED TO SET UP DELIMITERS FOR IFMTX
C         CALL SETDEL(N,IDEL(1))
C         WHERE N=NUMBER OF DELIMITERS IN ARRAY IDEL
C         IDEL=INTEGER ARRAY OF SIZE N OR GREATER
C         AND CONTAINS ONE DELIMITER PER WORD IN
C         5/7-ASCII RIGHT-JUSTIFIED.
          CALL SETDEL(2,IDEL(1))
          JK=0
100       CALL BUFIN(1,0,R(1),R(17))
          CALL DECODE(R(2))
1         K=IFUNIT(1)
          GO TO (1,2,3,4),K
2         JK=JK+1
C         NOW GET AN INTEGER
          INT(JK)=IFMTX(DUMMY)
C         WAS THIS INTEGER NR 5?
          IF(JK.EQ.5) GO TO 9999
C         NOW CHECK THE DELIMITER IF IT WAS A CARRIAGE RETURN
C         THEN THE DELIMITER NUMBER RETURN BY IFDEL(DUMMY)
C         EQUALS 1 (SAME AS NR OF CR IN ARRAY IDEL)
          IF(IFDEL(DUMMY).EQ.1)GO TO 100
C         NO,IT WAS NOT A CR
          GO TO 2
```

```
3       END OF FILE MUST BE HANDELD HERE
4       PARITY ERROR    -//-
9999    STOP 123
        END
.EJECT
```


GLOBALS:

NAME	INTERNAL GLOBAL	EXTERNAL GLOBAL
DECODE	G.WP	.DA
	IRFMT	
	G.CH	.ERROR
	G.BP	
	DECEOR	
	IFEOR	
	ERR.TS	
	ERR.OR	
	DECODE	
AFMT	AFMT	.DA
	AFMT1	.ERROR
		G.CH
		ERR.TS
FFMT	FFMT	.DA
		.ERROR
		ERR.TS
		ERR.OR
		G.CH
IBASE	IBASE	.DA
	IGETRX	.HIGH
		.RADIX
		.ERROR
IDCPNT	IDCPNT	.DA
	DECRES	G.WP
		G.BP
IFMT	IFMT	.DA
	.RADIX	.ERROR
	.HIGH	ERR.OR
		ERR.TS
		G.CH
IOFMT	IOFMT	.DA
		.RADIX
		.HIGH
		IFMT
KFMT	KFMT	ERR.OR
XFMT	XFMT	.DA
	G.WP	
	G.BP	