

プロセッサと機械語

2年次前期（第11回）

中島 克人

情報メディア学科

nakajima@im.dendai.ac.jp

1

本日の内容

- 入出力操作(マクロ命令)
- 文字コード操作

2

CASL II のマクロ命令

(教科書 p.31)

● 入力(in)命令

[書き方] IN **s**,**t** (**s**: 256語の領域のラベル, **t**: 1語の領域のラベル)

[説明] 1レコード(KB入力であれば1行)の文字データを読み込み、そこに含まれる文字を1語1文字の割りで領域**s**の先頭から順に格納する。第7～0ビットに**文字コード(ASCIIコード)**を、第15から8ビットに0を入れる。レコードの区切り符号(KB入力の際の復帰符号など)は格納されない。**t**には入力された文字数が入る。FRの値は「不定」となる。

1語	0	0	0	0	0	0	0	0	?	?	?	?	?	?	?	?
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

● 出力(out)命令

[書き方] OUT **s**,**t** (**s**: 256語の領域のラベル, **t**: 1語の領域のラベル)

[説明] 領域**s**に1語1文字の割りで格納された文字データを**t**に格納された文字数だけ、1レコードとして出力する。レコードの区切り記号は必要であれば自動的に付加される。

各語の第7～0ビットを**文字コード(ASCIIコード)**と見なし、第15から8ビットは無視される。FRの値は「不定」となる。

3

データの計算機内部表現

(教科書 p.31)

● ASCIIコード

ASCIIコードは7ビットであるが、COMET IIでは最上位1ビットに0を足した8ビットを文字コードとしている。

また、右記**赤文字**のコードは制御コードと称し、COMET IIでは文字コードとしては定義されていない。

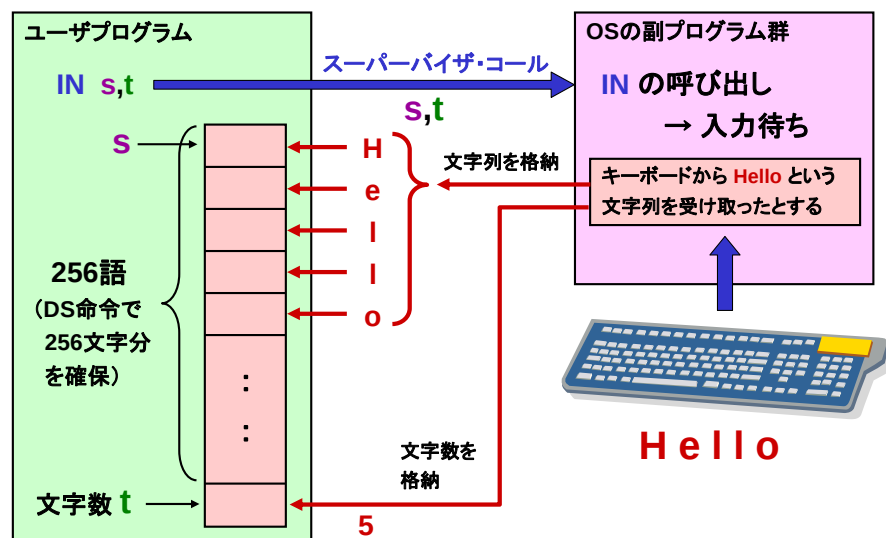
従って、文字列としてCASL IIプログラム上では表記できない

b3～b0	空白文字 b6～b4							
	000	001	010	011	100	101	110	111
0000	NUL	DLE	SP	0	@	P	'	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	\$	4	D	T	d	t
0101	ENQ	NAC	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	'	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF/NL	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	
1101	CR	GS	-	=	M	]	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	_	o	DEL

4

CASL II のマクロ命令

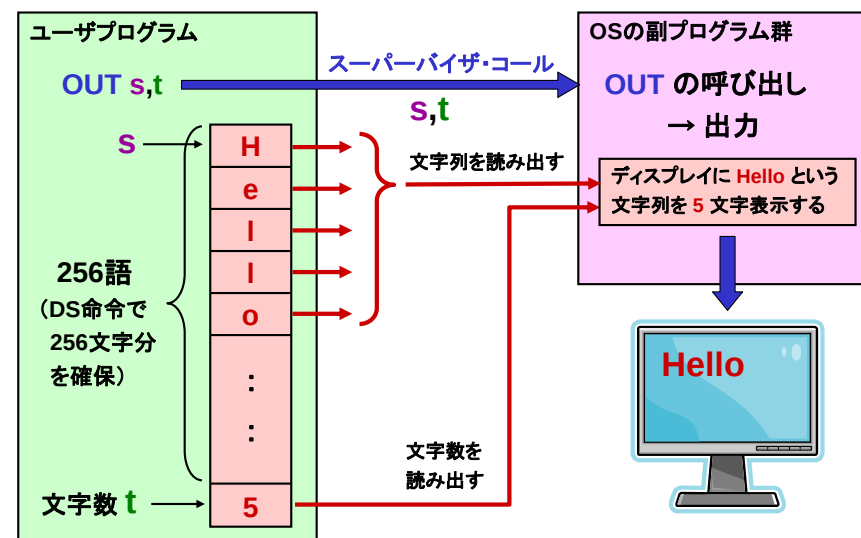
● 入力(in)命令の説明図



5

CASL II のマクロ命令

● 出力(out)命令の説明図



6

CASL II のIN命令

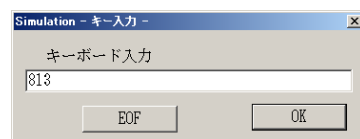
- IN命令で ABab という文字コードを STR番地に読み込むと、STR番地から連続した4語に #0041, #0042, #0061, #0062 が格納される



Label	Address	Value (HEX)	DEC)
LEN	=0039 (LEN)	=0004	4
STR	=003A (STR)	=0041	65
	003B	0042	66
	003C	0061	97
	003D	0062	98

- ◆ アルファベット大文字 (A~Z) のコードは #41 ~ #5A である
- ◆ アルファベット小文字 (a~z) のコードは #61 ~ #7A である

- IN命令で 813 という文字コードを STR番地に読み込むと、STR番地から連続した3語に #0038, #0031, #0033 が格納される



Label	Address	Value (HEX)	DEC)
LEN	=0039 (LEN)	=0003	3
STR	=003A (STR)	=0038	56
	003B	0031	49
	003C	0033	51

- ◆ 数字 0 ~ 9 の文字コードは #30 ~ #39 であるので、文字コードの入った語から #0030 をから引くと、その数字の値が求まる

7

プロセッサと機械語

文字コード操作

- 数字 (10進数 / 16進数) を表す文字コード (列) から値 (計算機の内部表現) への変換
- 値 (計算機の内部表現) から文字コード (列) (10進数 / 16進数) への変換

8

1桁の数字(10進数)を値に

- 数字 0 ~ 9 の文字コードは #30 ~ #39
メモリに10進数値 0~9 を表す文字コード(#0030~#0039)が入っている
値(計算機の内部表現)に変換するには #0030 を減算すれば良い

1桁の数字(16進数)を値に

- アルファベット大文字(A~F)のコードは #41 ~ #46
メモリに16進数値 0~F を表す文字コード(#0030~#0039, #0041~#0046)が入っている。
値(計算機の内部表現)に変換するには
まず #0030 を減算して、その値が 9 より小さければそのままに、
大きければ更に #0007 を減算する(#0037減算したことになる)

9

任意桁の数字(10進数)を値に

- メモリに任意桁の10進数値の文字コードが上位桁から並んで入っている。値に変換するには、
0~9 なので #0030 を減算し、下位桁があれば 10倍する。
それを合計に加え、下位桁がなくなるまでこれを繰り返す。

任意桁の数字(16進数)を値に

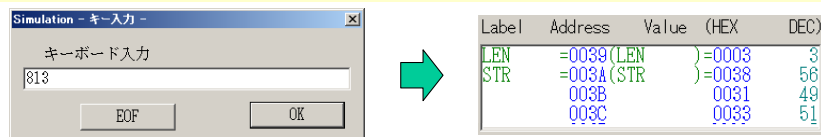
- メモリに任意桁の16進数値の文字コードが上位桁から並んで入っている。値に変換するには、
0~9 ならば #0030 を減算し、下位桁があれば 16倍(4ビット左シフト)する。
A~F ならば #0037 を減算し、下位桁があれば 16倍(4ビット左シフト)する。
それを合計に加え、下位桁がなくなるまでこれを繰り返す。

10

任意桁の数字(10進数)を値に (教科書 p.45)

- 例題 DECTOBIN ... IN命令で数字0~9を複数文字読み込み、それを10進数と解釈して値(計算機の内部表現)に変換してBIN番地に格納せよ

解説: IN命令で 813 という文字コードを STR番地に読み込むと、
STR番地から連続した3語に #0038, #0031, #0033 が格納される。



Label	Address	Value (HEX)	DEC)
LEN	=0039(LEN)	=0003	3
STR	=003A(STR)	=0038	56
	003B	0031	49
	003C	0033	51

そこで、#0008, #0001, #0003 を取り出し、
 $((8 \times 10) + 1) \times 10 + 3$ を求めて、BIN番地に格納すれば良い。

- 数字 0 ~ 9 の文字コードは #30 ~ #39 であるので、
文字コードの入った語から #0030 を引くと、その数字の値が求まる
- 先頭の値(8)から順に「10倍しては足し合わせ」を文字数回(3回)行くと、
求めたい値の内部表現(2進数)が下記のように求まる

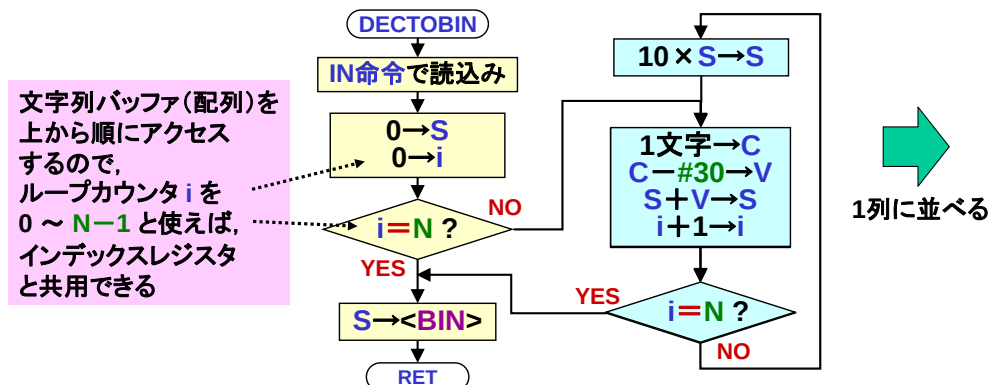
$$\begin{aligned} & ((8 \times 10) + 1) \times 10 + 3 \\ & \quad \text{10倍しては足し合わせ} \\ & \quad \text{10倍しては足し合わせ} \end{aligned}$$

11

任意桁の数字(10進数)を値に (教科書 p.45)

- 例題 DECTOBIN ... IN命令で数字0~9を複数文字を読み込み、それを10進数と解釈して値(計算機の内部表現)に変換してBIN番地に格納せよ

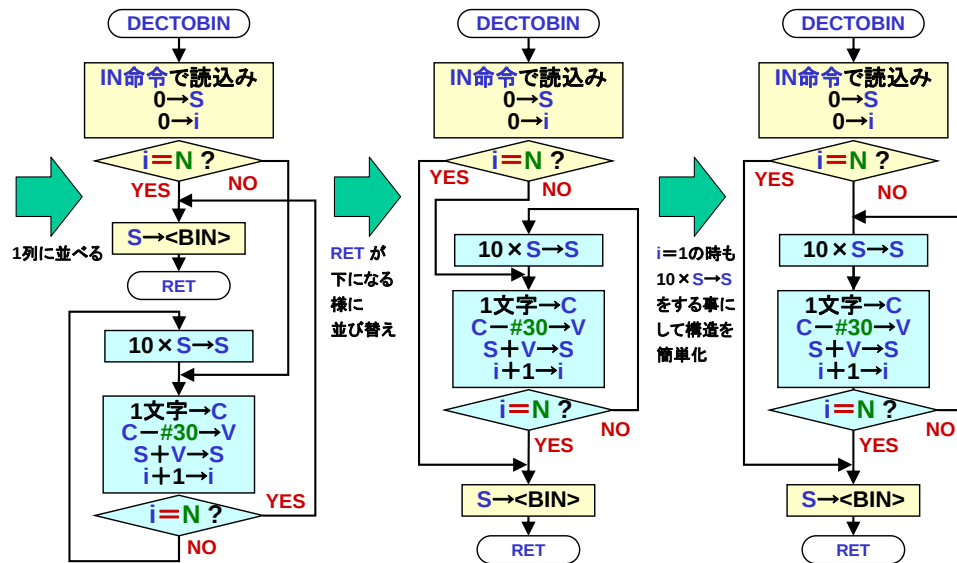
設計: 文字数を N, 文字コードから #30 を引いた値を V,
10進数の内部表現の値を S (初期値0) とすると、
N = 0 なら何もせずに終了、N = 1 なら V → S として終了、
N ≥ 2 なら $10 \times S + V \rightarrow S$, ループカウンタ i+1 → i で繰り返し。



12

任意桁の数字(10進数)を値に

- 例題 DECTOBIN ... IN命令で数字0~9を複数文字を読み込み、それを10進数と解釈して値(計算機の内部表現)に変換してBIN番地に格納せよ



13

任意桁の数字(10進数)を値に (教科書 p.45)

- 例題 DECTOBIN ... IN命令で数字0~9を複数文字を読み込み、それを10進数と解釈して値(計算機の内部表現)に変換してBIN番地に格納せよ

```

DECTOBIN START
IN      STR,LEN      ; 0 → GR0 (S:2進数集計用)
XOR     GR0,GR0      ; 0 → GR1 (i:ループカウンタ&インデックス)
XOR     GR1,GR1
CPA     GR1,LEN       ; i = 文字数 ?
JZE     LAST         ; 文字数=0の場合も有り得る
LOOP    LD      GR2,GR0 ; Sのコピー (GR2はワーク用)
        SLA     GR0,3   ; Sの8倍
        SLA     GR2,1   ; Sの2倍
        ADDA    GR0,GR2 ; Sの10倍
        LD      GR2,STR,GR1 ; 1文字取り出し
        SUBA    GR2,#0030 ; 文字コード-#0030
        ADDA    GR0,GR2 ; S+V→S
        LAD     GR1,1,GR1 ; ループカウンタを+1
        CPA     GR1,LEN ; i = 文字数 ?
        JNZ     LOOP    ; if ループカウンタ≠文字数 goto LOOP
LAST    ST      GR0,BIN ; 結果の格納
        RET
BIN     DS      1      ; 結果の2進数格納場所
LEN     DS      1      ; 文字数格納場所
STR     DS      256    ; 入出力バッファ
END
    
```

14

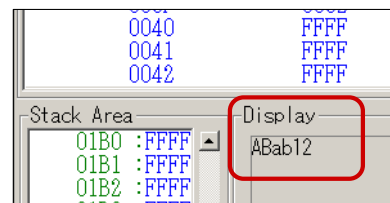
CASL II のOUT命令

- IN命令で ABab12 という文字コードをSTR番地に読み込み、それをそのままOUT命令で出力して見よう



Label	Address	Value (HEX)	DEC)
LEN	=0039(LEN)	=0006	6
STR	=003A(STR)	=0041	65
	003B	0042	66
	003C	0061	97
	003D	0062	98
	003E	0031	49
	003F	0032	50
	0040	FFFF	-1
	0041	FFFF	-1

- ◆アルファベット大文字(A~Z)のコードは #41 ~ #5A である
- ◆アルファベット小文字(a~z)のコードは #61 ~ #7A である
- ◆10進数字(0~9)のコードは #30 ~ #39 である



15

値0~9を1桁の数字(10進数)に

- 数字 0 ~ 9 の文字コードは #30 ~ #39
レジスタに値 0~9 (= #0000 ~ #0009) が入っているなら、出力(印字)するには #0030 を加算して文字コードに変換し、メモリに置いてOUT命令で出力(文字数は1)

値0~Fを1桁の数字(16進数)に

- アルファベット大文字(A~F)のコードは #41 ~ #46
レジスタに値 0~F (= #0000 ~ #000F) が入っている場合、16進表記で出力するならその値を調べ、0~9 ならば #0030 を加算し、A~F ならば #0037 を加算して文字コードに変換し、メモリに置いてOUT命令で出力(文字数は1)

16

値を任意桁の数字(10進数)に

- レジスタに任意の非負整数値(#0000～#7FFF)が入っている
10進表記で出力するならその値を除算できなくなる10で除算を繰り返す
最初の除算の余りが10進表記の最下位桁、最後の余りが最上位桁
各桁に#0030を加算し文字コードに変換して桁数だけの文字数を出力

値を任意桁の数字(16進数)に

- レジスタに任意の非負整数値(#0000～#7FFF)が入っている
16進表記で出力するならその値を4ビットずつ上位から1桁ずつ調べ、
0～9 ならば#0030を加算し、A～F ならば#0037を加算して文字コードに
桁数だけこれを繰り返し、最後に出力

17

値を任意桁の数字(10進数)に

- 例題 BINTODEC ... M番地の値を10進数字(0～9を表す文字)に直して
BUF番地から始まる文字用バッファ領域に格納し、OUT命令で表示せよ。
ただし、M番地の値は非負整数とし、出力時には前ゼロは無しとする。

例: M番地に813(10進数)という値が入っている場合、BUF番地から連続した3語に#0038, #0031, #0033を格納し、LEN番地に3を格納した後に、OUT命令を実行する。

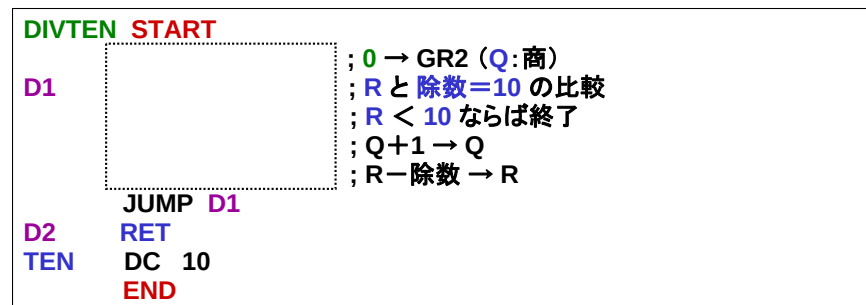
ポイントおよび設計

- M番地の2進数(内部表現)mを10進数(BCD)dに変換する必要がある
- mを10で割ると、余りがdの最下位桁の値となり、商をまた10で割った余りがその上の桁の値となり、... 以下同様である
- 10で割るのに サブルーチンDIVTENを用意したい
- 最下位桁から求まるが、BUFには最上位から格納しなくてはならないので、スタックを用いる
- push時に#0030を加え、文字コードに変換してしまう
- 何桁になるかは予め分からないので、スタックに番兵(終了印:例えば、数字以外の文字として0)を入れておき、それが取り出されたら終了とする。

18

CASLプログラミング(サブルーチン)

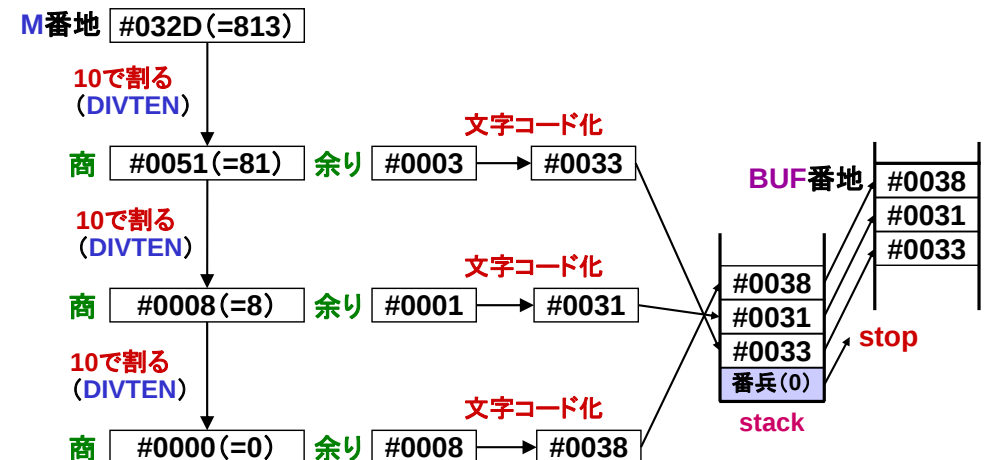
- 演習 DIVTEN ... GR1 の内容(≥ 0)を 10_{10} で除算(割算)し、
商を GR2 に、余りを GR1 に戻すサブルーチンを作成せよ。
なお、GR1 と GR2 以外のGRの内容は破壊しない事。



19

値を任意桁の数字(10進数)に

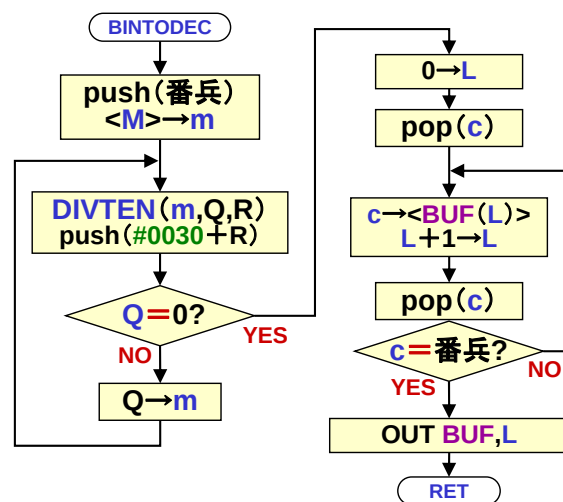
- 例題 BINTODEC ... M番地の値を10進数字(0～9を表す文字)に直して
BUF番地から始まる文字用バッファ領域に格納し、OUT命令で表示せよ。
ただし、M番地の値は非負整数とし、出力時には前ゼロは無しとする。



20

値を任意桁の数字(10進数)に

- 例題 BINTODEC ... M番地の値を10進数字(0~9を表す文字)に直して BUF番地から始まる文字用バッファ領域に格納し、OUT命令で表示せよ。ただし、M番地の値は非負整数とし、出力時には前ゼロは無しとする。



21

値を任意桁の数字(10進数)に

- 例題 BINTODEC ... M番地の値を10進数字(0~9を表す文字)に直して BUF番地から始まる文字用バッファ領域に格納し、OUT命令で表示せよ。ただし、M番地の値は非負整数とし、出力時には前ゼロは無しとする。

```

BINTODEC START ; M番地の内容の10進数に変換して出力する
PUSH 0 ; push(番兵=0)
LD GR1,M ; <M>=>m (変換対象内部表現)
D1 CALL DIVTEN ; GR1/10 -> 商GR2, 剰余GR1
; push #0030+GR1(剰余の文字コード)
; GR2 -> GR1
; GR1≠0 なら繰り返し
POP GR2 ; pop(c), GR1=L は既に 0
D2 ; c-><BUF(L)>
; L+1->L
; pop(c)
; GR2のゼロチェック
OR GR2,GR2 ; 番兵=0 でなければ繰り返し
JNZ D2 ; 文字数の格納
ST GR1,L ; 出力
OUT BUF,L
RET

M DS 1 ; 10進数で出力したい値
L DS 1 ; 出力文字数
BUF DS 256 ; 出力バッファ
END
    
```

22

値0~Fを1桁の数字(16進数)に

- 演習 SUBHEX ... GR1に16進数1桁(#0000~ #000F)が入っている。これを文字コード(#0030~#0039,#0041~#0046)に変換し、GR1に戻すサブルーチンを作成せよ。なお、GR1以外のGRの内容は破壊しない事。

```

SUBHEX START
; 0~9 のつもりで #0030 を加算
; #003A と比較
; < #003A ならば終了
; A~F だったので更に #0007 を加算
FIN RET
ZZ3A DC #003A
END
    
```

23

値を任意桁の数字(16進数)に

- 演習 VTOHEX ... GR0に非負整数(#0000~#7FFF)が入っている。これをサブルーチンSUBHEXを使用して、4桁の16進数文字コードに変換し、STR~STR+3番地に格納するプログラムを作成せよ。

```

VTOHEX START
LAD GR2,0 ; 格納用インデックス・レジスタ
LAD GR3,12 ; シフト数
LOOP ; GR0は保存
; 当該桁を1番下に
; 最下位桁以外をマスクアウト
; GR1の内容は文字コードになる
; メモリに格納
; インデックス・レジスタを+1
; シフト数を-4
; その結果、シフト数が正ならループ
; シフト数が0でもループ
RET
STR DS 4
END
    
```

24