

1. プログラム例

以下、プログラムの動作とその CASL II のソースを示す。
ソース中の [ア] へに入る適当な文を書け。(各 1 点)

1.1 加算

[動作]

- ラベル A と B が示す値を算術加算して、その結果をラベル WA の領域に格納する。

[アセンブラソース]

```
PRG      START
          [ア]
          [イ]
          [ウ]
          RET
A        DC      20
B        DC      30
WA       DS      1
          END
```

1.2 加算と条件分岐

[動作]

- ラベル A と B が示す値を算術加算して、その結果をラベル WA の領域に格納する。
- ただし、計算結果がオーバーフローした場合は、`OVER` と表示します。

[アセンブラソース]

```
PRG      START
          LD      GR1,A
          ADDA   GR1,B
          [ア]
          [イ]
          [ウ]
L1
L2       ST      GR1,WA
          RET
A        DC      20000
B        DC      30000
WA       DS      1
BUFF    DC      `OVER`
LEN     DC      4
          END
```

;OF が 1 ならば L1 へ
;無条件で L2 へ
;`OVER` を表示
;計算結果の格納

1.3 マスク処理と条件分岐

[動作]

- ラベル A の示す値が偶数の場合 `GUUSUU`、奇数の場合 `KISUU` と表示する。

[アセンブラソース]

```
PRG      START
          LD      GR1,A
          [ア]
          [イ]
          OUT    GU,LEN
          [ウ]
KISUU    OUT    KI,LEN
FIN      RET
A        DC      1452
MASK     [エ]
KI       DC      `KISUU`
GU       DC      `GUUSUU`
LEN     DC      6
          END
```

;マスク処理
;結果≠0 時 KISUU
;終了へ

1.4 論理演算とアドレス修飾

[動作]

- ラベル A と B の示す値の論理積 (A AND B)、論理和 (A OR B)、排他的論理和 (A XOR B) を 3 語確保されたラベル ANS に格納する。
- 論理積をアドレス ANS、論理和を ANS+1、排他的論理和を ANS+2 に格納します。

[アセンブラソース]

```
PRG      START
          LAD    GR2,0
          LD      GR1,A
          AND    GR1,B
          [ア]
          [イ]
          [ウ]
          [エ]
          [オ]
          LAD    GR2,1,GR2
          LD      GR1,A
          XOR    GR1,B
          ST     GR1,ANS,GR2
          RET
A        DC      #0030
B        DC      #00F9
ANS     DS      3
          END
```

;ANS+0 番地に格納
;インクリメント
;ANS+1 番地に格納

1.5 シフト演算

[動作]

- ラベル A が示す値の 0.75 倍をラベル KOTAE が示す場所に格納する。
- ヒント 0.75=1-1/4

[アセンブラソース]

```

PRG      START
        LD      GR1,A
        LD      GR2,A
        _____ [ア] _____ ;1/4 倍
        _____ [イ] _____
        ST      GR1,KOTAE
        RET
A        DC      50
KOTAE    DS      1
        END

```

1.6 繰り返し処理

[動作]

- ラベル DATA (先頭アドレス) に入っている値の中から最大値を探すプログラムである。
- データの個数は、ラベル KOSUU が示している。
- 探した最大値は、ラベル MAX が示す場所に格納する。

[アセンブラソース]

```

PRG      START
        LAD     GR2,0 ;カウンタを 0 にする
        LD      GR1,KOSUU
        _____ [ア] _____ ;データ数をデクリメント
        _____ [イ] _____
        ST      GR0,MAX ;暫定最大値の保存
LOOP     LAD     GR2,1,GR2 ;カウンタをインクリメント
        _____ [ウ] _____ ;データの読み込み
        CPA     GR0,MAX ;データの比較
        _____ [エ] _____
        ST      GR0,MAX ;最大値の保存
SKIP    CPA     GR1,GR2 ;データ数とカウンタの比較
        _____ [オ] _____ ;データ数が多いとき LOOP へ
        RET
DATA    DC      10,15,8,20,7
KOSUU   DC      5
MAX     DS      1
        END

```

1.7 繰り返し処理とサブルーチン

[動作]

- ラベル DATA (先頭アドレス) に入っている値の中から最大値を探すプログラムである。
- データの個数は、ラベル KOSUU が示している。
- 探した最大値は、ラベル MAX が示す場所に格納する。
- 最大値探索の部分はサブルーチン化している。そのサブルーチンへは、引数として GR1 を用いて、データの個数を渡している。

[アセンブラソース]

```

PRG      START
        LD      GR1,KOSUU
        _____ [ア] _____
        RET
DATA    DC      10,15,8,20,7
KOSUU   DC      5
MAX     DS      1
;サブルーチン化した部分
SAIDAI  LAD     GR1,-1,GR1
        LAD     GR2,0
        LD      GR0,DATA
        _____ [イ] _____ ;暫定最大値の保存
LOOP    LAD     GR2,1,GR2
        LD      GR0,DATA,GR2
        _____ [ウ] _____ ;データの比較
        JMI     SKIP
        _____ [エ] _____ ;最大値の保存
SKIP    CPA     GR1,GR2 ;データ数とカウンタの比較
        JPL     LOOP ;データ数が多い時 LOOP へ
        _____ [オ] _____
        END

```

1.8 アドレスの受け渡し

[動作]

- ラベル DATA (先頭アドレス)に入っている値の中から最大値を探すプログラムである。
- データの個数は、ラベル KOSUU が示している。
- 探した最大値は、ラベル MAX が示す場所に格納する。
- 最大値探索の部分はサブルーチン化している。そのサブルーチンへは、引数として GR1、GR2、GR3 を用いている。

[アセンブラソース]

```

PGM      START
        [ア]
        [イ]
        [ウ]
CALL     SAIDAI
RET
DATA     DC      10,15,8,20,7
KOSUU    DC      5
MAX      DS      1
END
SAIDAI   START
        PUSH    0,GR1
        PUSH    0,GR2
        PUSH    0,GR4
        PUSH    0,GR5
        LAD     GR4,0
        LAD     GR1,-1,GR1 ;データの数をデクリメント
        LD      GR5,0,GR2 ;DATAの先頭を読み込む
        ST      GR5,0,GR3 ;暫定最大値を保存
LOOP     LAD     GR2,1,GR2 ;アドレスをインクリメント
        LAD     GR4,1,GR4 ;カウンタをインクリメント
        LD      GR5,0,GR2
        CPA     GR5,0,GR3 ;GR5と最大値の比較
        JMI     SKIP
        ST      GR5,0,GR3
SKIP     CPA     GR1,GR4 ;データ数とカウンタの比較
        JPL     LOOP
        POP     GR5
        POP     GR4
        POP     GR2
        POP     GR1
        RET
        END
    
```

2. プログラムテクニック

2.1 PUSH と POP (5点)

前問 1.8 アドレスの受け渡しのプログラムのサブルーチンの最初と最後に PUSH と POP 命令が使われている。このプログラム中でのこれらの PUSH と POP 命令の役割を示せ。

```

SAIDAI  START
        PUSH    0,GR1
        PUSH    0,GR2
        PUSH    0,GR4
        PUSH    0,GR5
        }
        POP     GR5
        POP     GR4
        POP     GR2
        POP     GR1
        RET
        END
    
```

2.2 数値を文字データに変換(各5点)

表1に CASL II で使われるも文字コード JIS X0201 の一部を示す。これを参考にして数値を文字に変換する方法についての以下の問いに答えよ。

- 数値 $(2050)_{10}$ を文字に変換したい。変換された文字はメモリー上にどのように格納されるか?。16進数で答えよ。
- 数値の各桁、例えば $(2050)_{10}$ の千の位の 2 を文字に変換する方法を述べよ。

表1 JIS X0201の一部

下位 4ビット	上位4ビット					
	2	3	4	5	6	7
0	間隔	0	@	P	`	p
1	!	1	A	Q	a	q
2	”	2	B	R	b	r
3	#	3	C	S	c	s
4	\$	4	D	T	d	t
5	%	5	E	U	e	u
6	&	6	F	V	f	v
7	'	7	G	W	g	w
8	(8	H	X	h	x
9)	9	I	Y	i	y
10	*	:	J	Z	j	z
11	+	;	K	[k	{
12	,	<	L	\	l	
13	-	=	M]	m	}
14	.	>	N	^	n	~
15	/	?	O	_	o	

3. プログラム

以下の動作の CASL II のプログラムを書け。ただし、注釈文(コメント)は、書かなくてもよい。(各 12 点)

3.1 データを 5.75 倍

- ラベル DATA が示す領域に $(100)_{10}$ を格納する。
- シフト命令を利用して、この値を 5.75 倍する。
- 5.75 倍された値は、ラベル名 KEKKA が示す領域に格納する。

3.2 合計

- ラベル DATA には、(54, 34, 82, 49, 99, 37) の 6 つの値が格納されている。この合計を求めるプログラムである。
- ラベル KOSUU には、データ数、すなわち 6 が格納されている。
- 求められた合計は、ラベル SUM に格納する。
- 合計を計算する部分はサブルーチンを用いること。

4. コンピューターの仕組み

4.1 チューリング機械

チューリング機械の動作を以下に、構造を図 1 に示す。

- 書き換え可能な無限に長いテープと、オートマトンと言われる移動可能な機械からできている。
- テープには、いろいろな記号が書かれている。オートマトンには、テープの内容を読み書き可能なヘッドと内部状態を記憶する装置、テープの任意の位置に移動する装置から構成されている。
- オートマトンの動作(テープの読み書き)や移動は、今の場所のテープの記号と内部状態により決まる。

チューリング機械と COMET II との対応で正しいものを選択肢から選べ。(各 2 点)

チューリング機械	コンピューター
書き換え可能テープ	[ア]
オートマトン	[イ]
テープに書かれた記号	[ウ]
内部状態を記憶する装置	[エ]

[選択肢]

- | | |
|----------|-----------|
| ① ディスプレイ | ② データや命令 |
| ③ 高級言語 | ④ メモリー |
| ⑤ CPU | ⑥ ハードディスク |
| ⑦ 文字コード | ⑧ レジスタ |
| ⑨ データバス | ⑩ アドレスバス |

4.2 ノイマン型コンピューター

ノイマン型コンピューターの特徴を 2 つ述べよ。(各 5 点)

4.3 コンピューターの改良

チューリング機械をコンピューターにすると、図 2 のような構成になる。しかし、このコンピューターは実用に適さない。あなた(3E 受験者)がこれを改良する立場にあるとして、どのような機能を追加すればよいか考えよ。改良すべき点とその理由を記入せよ。答は何個書いても良い。(13 点)

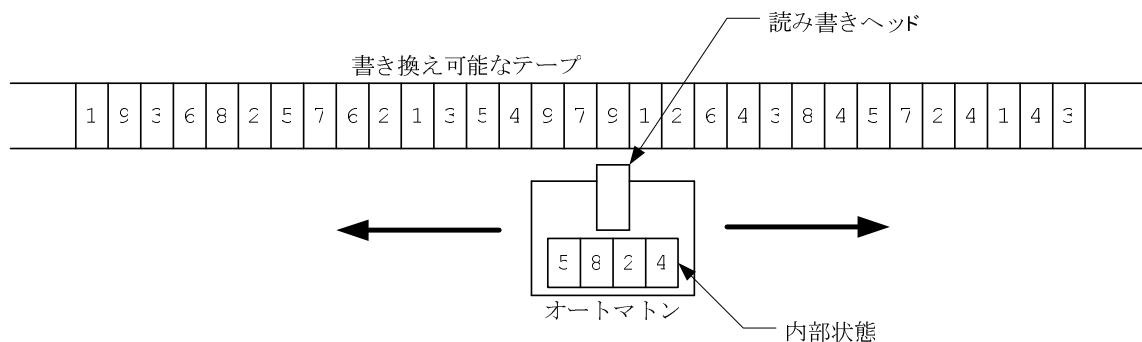


図 1 チューリング機械

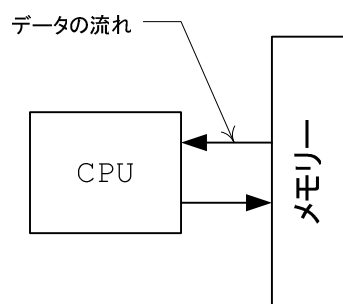


図 2 最も単純なコンピューター

