学年末試験 (3E 電子計算機)

2005 年 2 月 28 日

1　基本的なテックニック

以下の問いに答えよ。ただし、プログラムについての問いは、全てを書く必要はなく、重要な部分のみ記述すればよい。即ち、START や END 等は書く必要はない。

[**問 1**] 以下のプログラムは、ラベル A とラベル B の内容を加算して、C に格納するプログラムである。間違いを指摘せよ。

```
PGM   STARRT
ADDA  C,A,B
      RET
A     DC    5
B     DC    3
C     DS    1
      END
```

[**問 2**] GR1−GR2 が負ならばラベル L1 へ分岐する文を書け。

[**問 3**] レジスターの内容が ＊＊**1010＊＊**1100 となっていることを確認したい。結果は、ZF(ゼロフラグ) に設定されるものとし、その部分のプログラムを書け。ただし、＊は 0 でも 1 でもよく興味の対象外とする。

[**問 4**] GR1 をカウンターとして用いる。GR1 を 1 に初期化するには、プログラム中でどのように記述するか?。

[**問 5**] カウンター GR1 を+1 したい (値を 1 増加させる)。そのためには、プログラム中でどのように記述するか?

[**問 6**] 次のプログラムは、レジスター GR1 の値を 5 倍する。［　　］に入る適当な文を書け。

```
      LD    GR2,GR1
      [　　　　　　]
      ADDA  GR1,GR2
```

[**問 7**] 次のプログラムは、整数の 1〜100 までの合計を計算し、結果をラベル ANS に格納するプログラムである。［　　］に入る適当な文を書け。

```
PGM   START
      LAD   GR0,0
      LAD   GR1,0
LOOP  LAD   GR1,1,GR1    ; インクリメント
      ADDA  GR0,GR1      ; 加算
      [　ア　]          ;100 との比較
      [　イ　]          ; ジャンプ
      ST    GR0,ANS
      RET
ANS   DS    1
      END
```

[**問 8**] ほとんどのプログラミング言語はサブルーチンが使えるようになっている。サブルーチンを使う理由を説明せよ。

[**問 9**] CASL II でサブルーチン MYSUB を呼び出すには、プログラム中にどのように書くか?。さらに、サブルーチンから呼び出し元に戻る命令を書け。

[**問 10**] サブルーチンでの計算で、レジスターの GR0，GR1，GR2，GR3，GR4 を使う。ただし、このサブルーチンの使用前と使用後で、GR0，GR1，GR2 の値は変化させたくない。この場合のサブルーチンのプログラムの書き方を示せ。

[**問 11**] 次のようなプログラムがある。DATA で示される数列 1,2,3,4,5 の最後の 5 が格納されているアドレスとラベル LAST が示すアドレスの関係

を示せ。

```
DATA DC          1,2,3,4,5
LAST DS          0
```

[**問 12**] 汎用レジスター GR1 には、ある整数値 (0〜9) が格納されている。この整数値を文字コードに変換したい。変換された文字コードは、GR1 に格納されるとする。以下のプログラムでそれを実現するが、［　　］に入る文を書け。

```
      省略
      [    ア    ] ; 文字コードに変換
      省略
MOJI  DC   [  イ  ]
      省略
```

[**問 13**]

汎用レジスター GR0 の値を-1 倍したい。その部分のプログラムを書け。

2　プログラムの構造

右のプログラムについて、問いに答えよ。ただし、このプログラムは、ラベル PGM から実行されると仮定する。

[**問 1**] 答案用紙に、このプログラムのメインルーチンとサブルーチン、そしてデータの領域を示せ。サブルーチンが複数ある場合は、サブルーチン 1、サブルーチン 2、··· のように示すこと。

[**問 2**] このプログラムの 4 行目 LAD GR2,-1,GR2 という文がある。この文が実行されると GR2 が表すアドレスは何になるか?

[**問 3**] サブルーチンの処理の始まりに書かれている PUSH 0,GR1 の役割を述べよ。

[**問 4**] サブルーチンの処理の終わりに書かれている POP GR1 の役割を述べよ。

[**問 5**] このプログラムの実行後のラベル SUM と MAX が示す内容はどうなっているか?。

```
PGM   START
      LAD  GR1,DATA  ;DATA の先頭アドレス
      LAD  GR2,LAST  ;DATA の最終アドレス+1
      LAD  GR2,-1,GR2
      CALL SBSUM
      ST   GR3,SUM
      CALL SBMAX
      ST   GR3,MAX
      RET
DATA  DC   9,5,6,8,4,1,8
LAST  DS   0
SUM   DS   1
MAX   DS   1
      END
SBSUM START
      PUSH 0,GR1
      LD   GR3,0,GR1 ; 合計
LOOP  LAD  GR1,1,GR1
      ADDA GR3,0,GR1
      CPA  GR2,GR1
      JPL  LOOP
      POP  GR1
      RET
      END
SBMAX START
      PUSH 0,GR1
      LD   GR3,0,GR1 ; 暫定最大
LOOP  LAD  GR1,1,GR1
      CPA  GR3,0,GR1
      JMI  BIG
      JUMP NEXT
BIG   LD   GR3,0,GR1
NEXT  CPA  GR2,GR1
      JPL  LOOP
      POP  GR1
      RET
      END
```

3 CASL IIのプログラム例

プログラム中の [　　] に入る適当な文を書け。

3.1 加算と条件分岐

- 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。
- 計算がオーバーフローした場合は'OVER' と表示する。

```
PRG   START           ;A の値を GR1 に入れる
      LD    GR1,A     ;GR1=GR1+B
      ADDA  GR1,B     ;OF が 1 ならば L1 へ
      [　　ア　　]    ;OF が 1 ならば L1 へ
      [　　イ　　]    ; 無条件で L2 へ
L1    [　　ウ　　]    ; "OVER "を表示
L2    ST    GR1,WA    ;GR1 の値を WA へ入れる
      RET
A     DC    20000
B     DC    30000
WA    DS    1
BUFF  DC    'OVER'    表示する文字列
LEN   DC    4         表示する文字列の長さ
      END
```

3.2 論理演算とアドレス修飾

- 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。

```
PRG   START
      LAD   GR2,0         ;GR2 に 0 を入れる
      LD    GR1,A         ;A の値を GR1 に入れる
      AND   GR1,B         ;GR1 と B の論理積
      [　　ア　　]        ;ANS+0 番地に入れる
      [　　イ　　]        ; インクリメント
      [　　ウ　　]        ;A の値を GR1 に入れる
      [　　エ　　]        ;GR1 と B の論理積
      [　　オ　　]        ;ANS+1 番地に入れる
      LAD   GR2,1,GR2     ; インクリメント
      LD    GR1,A         ;A の値を GR1 にいれる
      XOR   GR1,B         ;GR1 と B の排他的論理和
      ST    GR1,ANS,GR2   ;ANS+2 番地に入れる
      RET
A     DC    #0030
B     DC    #00F9
ANS   DS    3             ;3 語確保
      END
```

3.3 シフト演算

- 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　　$0.75 = 1 - 1/4$

```
PRG   START
      LD    GR1,A         ;A の値を GR1 に入れる
      LD    GR2,A         ;A の値を GR2 に入れる
      [　　ア　　]        ;GR2 の値を 1/4 倍する
      [　　イ　　]        ;A-1/4A の計算
      ST    GR1,KOTAE     ;GR1 を KOTAE に入れる
      RET
A     DC    50
KOTAE DS    1
      END
```

3.4 繰り返し処理

- 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。

```
PRG   START
      LAD   GR2,0         ; カウンタを 0 にする
      LD    GR1,KOSUU     ;GR1 にデータを入れる
      [　　ア　　]        ; データ数をデクリメント
      [　　イ　　]        ; 先頭のデータの読み込み
      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       ; 最大値と比較
      [　　エ　　]        ;GR0 が小さいとき SKIP へ
      ST    GR0,MAX       ; 最大値の保存
SKIP  CPA   GR1,GR2       ; データ数とカウントの比較
      [　　オ　　]        ; 数の方が多いとき LOOP へ
      RET
DATA  DC    10,15,8,20,7
KOSUU DC    5
MAX   DS    1
      END
```

3.5 数値データを文字データに変換

- ラベル A に入っている数値 (最大 5 桁) を文字列に変換して、OUT 命令で表示する。
- 表示には 6 桁 (カラム) 用意して、第 1 桁は符号で負の場合のみ表示する。2〜6 桁は絶対値を表す。ただし、上位の桁が 0 の場合、スペースを入れる。

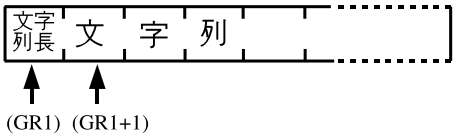
```
REI5A  START
       LAD   GR1,0          ; カウンタを 0 にする
       LD    GR0,A          ;A の値を GR0 に入れる
       [      ア      ]     ; 負数か判断
       LD    GR3,=#0020     ; 正のときはスペース
       ST    GR3,BUFF,GR1   ; 文字を保存
       [      イ      ]     ; 数値処理にジャンプ
FUSUU  LD    GR3,='-'       ; 負数の時はマイナス
       ST    GR3,BUFF,GR1   ; 文字を保存
       LAD   GR0,0          ; 絶対値処理
       SUBA  GR0,A          ; 絶対値処理
KEISAN LAD   GR1,1,GR1      ; カウンタをインクリメント
       LD    GR2,WORK,GR1   ; 除数を GR2 に読み込み
       [      ウ      ]     ; 割り算処理
       PUSH  0,GR3          ;GR3 を一時退避
       OR    GR3,WORK       ; 文字コードの判別
       ST    GR3,WORK       ; 結果を WORK に
       POP   GR3            ;GR3 を復帰
       JZE   SPACE          ;OR の結果が 0 のとき分岐
       [      エ      ]     ;#0030 を加算
       JUMP  HOZON          ; 文字コードの保存処理へ
SPACE  [      オ      ]     ;OR が 0 のときスペース
HOZON  ST    GR3,BUFF,GR1   ; 文字コードの保存
       CPA   GR1,C4         ; ループ回数の比較
       JMI   KEISAN         ;4 回ループする
       OR    GR0,MOJI       ;1 の位を文字コードに
       LAD   GR1,1,GR1      ; カウンタをインクリメント
       ST    GR0,BUFF,GR1   ; 文字コードの保存
       OUT   BUFF,LEN       ; 結果の表示
       RET
A      DC    -2050
BUFF   DS    6              ; 文字コードの保存領域
LEN    DC    6              ; 文字コードの長さ
WORK   DC    0,10000,1000,100,10
MOJI   DC    #0030
C4     DC    4
       END
DIV    START
       LAD   GR3,0          ; 商を 0 にする
DLOOP  CPA   GR0,GR2        ; 除数で引けるか判別
       JMI   FIN            ; 引けなければ FIN へ
       SUBA  GR0,GR2        ; 被除数-除数
       LAD   GR3,1,GR3      ; 商をインクリメント
       JUMP  DLOOP          ; 繰り返し処理へ
FIN    RET
       END
```

4 応用問題

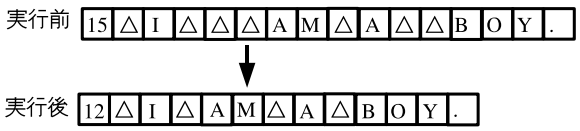
以下の問題は難しいが、配点は低い。時間が余った人はトライせよ。これは、平成 16 年秋の基本情報処理技術者試験に出題された問題である。

文字列中の二つ以上連続する間隔文字（以下，空白という）を一つに詰める副プログラム SPCSUP である。副プログラムとは、サブルーチンのことである。

- 主プログラムは，先頭に文字列長，その後に文字列が格納された領域の先頭アドレスを GR1 に設定して，副プログラム SPCSUP を呼ぶ



- 副プログラム SPCSUP は，文字列中の空白を詰めた後，文字列長を再設定する。
- 副プログラム SPCSUP から戻るとき，汎用レジスタの内容は元に戻す。



注　△は空白を示す。

[問 1] プログラム中の［　　］に入れる正しい答えを、解答群の中から選べ。

[a に関する解答群]

```
ア ADDL GR3,0,GR1    イ ADDL GR4,0,GR1
ウ LAD GR1,1,GR1     エ LAD GR3,1,GR3
オ LAD GR4,1,GR4
```

[b に関する解答群]

```
ア JNZ CHMOVE        イ JNZ CONT
ウ JNZ RESET         エ JZE CHMOVE
オ JZE CONT          カ JZE RESET
```

[c に関する解答群]

```
ア LAD GR2,1,GR2     イ LAD GR3,1,GR3
ウ LD GR6,1          エ LD GR6,=1
```

```
SPCSUP  START  RPUSH
        LD     GR2,GR1       ; 転送元ポインタ（GR2）初期化
        LD     GR3,GR1       ; 転送先ポインタ（GR3）初期化
        LD     GR4,GR1       ; 文字列終了位置ポインタ（GR4）初期化
        [      a       ]
        LD     GR6,=0        ; 連続空白識別フラグ（ GR6 ）初期化
        LP     LAD           GR2,1,GR2
        LD     GR5,0,GR2
        CPL    GR5,=' '
        [      b       ]
        LD     GR6,GR6
        JNZ    CONT
        [      c       ]
        JUMP   CHMOVE
RESET   LD     GR6,=0
CHMOVE  LAD    GR3,1,GR3
        ST     GR5,0,GR3
CONT    CPL    GR2,GR4
        JMI    LP
        SUBL   GR3,GR1       ; 新文字列長の算出
        ST     GR3,0,GR1
        RPOP
        RET
        END
```

5　参考資料

5.1　命令語の構成

命令語の構成は定義しないが、次のような構成を想定する。ここで、OP の数値は 16 進数表示で示す。

第 1 語				第 2 語	命令語長	命令語とアセンブラとの対応	
OP		r/r1	x/r2	adr		機械語命令	意味
主 OP	副 OP						
0	0	-	-	-	1	NOP	no operation
1	0				2	LD r,adr,x	load
	1				2	ST r,adr,x	store
	2				2	LAD r,adr,x	load address
	4				1	LD r1,r2	load
2	0				2	ADDA r,adr,x	add arithmetic
	1				2	SUBA r,adr,x	subtract arithmetic
	2				2	ADDL r,adr,x	add logical
	3				2	SUBL r,adr,x	subtract logical
	4			-	1	ADDA r1,r2	add arithmetic
	5			-	1	SUBA r1,r2	subtract arithmetic
	6			-	1	ADDL r1,r2	add logical
	7			-	1	SUBL r1,r2	subtract logical
3	0				2	AND r,adr,x	and
	1				2	OR r,adr,x	or
	2				2	XOR r,adr,x	exclusive or
	4			-	1	AND r1,r2	and
	5			-	1	OR r1,r2	or
	6			-	1	XOR r1,r2	exclusive or
4	0				2	CPA r,adr,x	compare arithmetic
	1				2	CPL r,adr,x	compare logical
	4			-	1	CPA r1,r2	compare arithmetic
	5			-	1	CPL r1,r2	compare logical
5	0				2	SLA r,adr,x	shift left arithmetic
	1				2	SRA r,adr,x	shift right arithmetic
	2				2	SLL r,adr,x	shift left logical
	3				2	SRL r,adr,x	shift right logical
6	1				2	JMI adr,x	jump on minus
	2				2	JNZ adr,x	jump on non zero
	3				2	JZE adr,x	jump on zero
	4				2	JUMP adr,x	unconditional jump
	5				2	JPL adr,x	jump on plus
	6				2	JOV adr,x	jump on overflow
7	0	-			2	PUSH adr,x	push
	1		-	-	1	POP r	pop
8	0	-			2	CALL adr,x	call subroutine
	1	-	-	-	1	RET	return form subroutine
9 ～ E						その他の命令	
F	0	-			2	SVC adr,x	supervisor call

5.2　文字の符号表

JIS X 0201 ラテン文字・片仮名用 8 ビット符号で規定する文字の符号表を使用する。

右に符号表の一部を示す。1 文字は 8 ビットからなり、上位 4 ビットを列で、下位 4 ビットを行で示す。例えば、間隔、4、H、\ のビット構成は、16 進数表示で、それぞれ 20、34、48、5C である。16 進数表示で、ビット構成が 21〜7E(及び表では省略している A1〜DF) に対応する文字を図形文字という。図形文字は、表示 (印刷) 装置で、文字として表示 (印字) できる。

この表にない文字とそのビット構成が必要な場合は、問題中で与える。

行＼列	02	03	04	05	06	07
0	間隔	0	@	P	‘	p
1	!	1	A	Q	a	q
2	"	2	B	R	b	r
3	#	3	C	R	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	