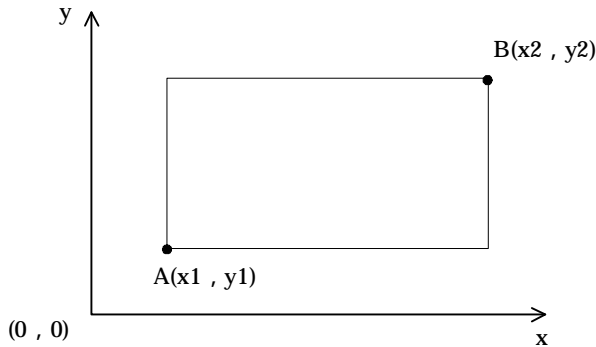


平成 18 年度 春期 F E 午後問題 CASL

問 9 次のアセンブラプログラムの説明及びプログラムを読んで、設問 1, 2 に答えよ。

〔プログラムの説明〕

(1) 副プログラム OBLONG は、各座標値が整数である 2 点 A (x1, y1), B (x2, y2) を対角の頂点とし、各辺が x 軸又は y 軸に平行な長方形の面積を求めるプログラムである。



次のパラメタの先頭アドレスが GR1 に格納されて、主プログラムから渡される。

(GR1)+0	x1
+1	y1
+2	x2
+3	y2

x1 x2, y1 y2 とする。

求めた面積は GR0 に格納して、主プログラムに返す。

副プログラム OBLONG から戻るとき、汎用レジスタ GR1 ~ GR7 の内容は元に戻す。

(2) 副プログラム ABS は、OBLONG から渡された GR2 の内容をその絶対値に置き換えるプログラムである。

副プログラム ABS の実行によって、汎用レジスタ GR1, GR3 ~ GR7 の内容は変わらない。

(3) プログラム中の演算で、あふれは起きないものとする。

〔プログラム〕

(行番号)

1 OBLONG START
2 RPUSH

```

3      LD      GR2, 0, GR1      ;
4      SUBA   GR2, 2, GR1      ; } GR3  |x1-x2|
5      CALL   ABS              ;
6      LD      GR3, GR2        ;
7      LD      GR2, 1, GR1      ;
8      SUBA   GR2, 3, GR1      ; } GR2  |y1-y2|
9      CALL   ABS              ;
10     LD      GR0, =0          ; 面積の初期化
11     LOOP   SRL      GR2, 1
12     [ ]      a
13     JZE    FIN1
14     [ ]      b
15     ADD    ADDA   GR0, GR3
16     SHIFT SLA     GR3, 1
17     JUMP   LOOP
18     FIN1   RPOP
19           RET
20     ;
21     ABS   LD      GR0, GR2
22           JMI    MINUS
23           JUMP   FIN2
24     MINUS LD      GR2, =0
25     [ ]      c
26     FIN2   RET
27           END
    
```

設問 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, b に関する解答群

- | | | | |
|---|------------|---|-----------|
| ア | JMI ADD | イ | JMI LOOP |
| ウ | JMI SHIFT | エ | JOV ADD |
| オ | JOV LOOP | カ | JOV SHIFT |
| キ | JUMP ADD | ク | JUMP LOOP |
| ケ | JUMP SHIFT | | |

c に関する解答群

- | | | | |
|---|---------------|---|---------------|
| ア | ADDA GR0, GR2 | イ | ADDA GR2, GR0 |
| ウ | SUBA GR0, GR2 | エ | SUBA GR2, GR0 |

設問 2 2 点 A と B に次の座標を与えたとき、行番号 15 の ADDA 命令は何回実行されるか。正しい答えを、解答群の中から選べ。

A の座標 : (3, 6)
B の座標 : (-5, 17)

解答群

- ア 1 イ 2 ウ 3 エ 4 オ 5

示現塾 プロジェクトマネージャ・テクニカルエンジニア(ネットワーク)など各種セミナーを開催中!!

開催日、受講料、カリキュラム等、詳しくは、http://zigen.cosmoconsulting.co.jp 今すぐアクセス!!

問13 次のアセンブラプログラムの説明及びプログラムを読んで、設問1～3に答えよ。

〔プログラムの説明〕

32ビット符号なし2進整数と15ビット符号なし2進整数の乗算を行う副プログラムMULT32と、32ビット符号なし2進整数同士の加算を行う副プログラムADD32である。

(1) MULT32は、32ビットの被乗数が格納されている領域の先頭番地をGR2に、15ビットの乗数をGR1に設定して呼び出される。MULT32は、32ビットの乗算結果を被乗数の格納領域に上書きして呼出し元に戻る。乗算結果のうち、格納領域からはみ出した部分は無視する。MULT32は、その処理の過程でADD32を利用する。

(2) ADD32は、被加数の格納されている領域の先頭番地をGR2に、加数の上位語と下位語をそれぞれGR1とGR0に設定して呼び出される。ADD32は、32ビットの加算結果を被加数の格納領域に上書きして呼出し元に戻る。加算結果のうち、格納領域からはみ出した部分は無視する。

(3) 被乗数と被加数の格納形式を、次に示す。

上位語	下位語
(GR2) + 0	(GR2) + 1

(4) 副プログラムから戻るとき、汎用レジスタGR1～GR7の内容は元に戻る。

〔プログラム〕

(行番号)

```

1  MULT32  START           ; シフトによる乗算
2          RPUSH
3          LD   GR5, 0, GR2 ;
4          ST   GR5, H      ;
5          LD   GR5, 1, GR2 ;
6          ST   GR5, L      ;
7          LD   GR5, =0     ;
8          ST   GR5, 0, GR2 ;
9          ST   GR5, 1, GR2 ;
10         LAD  GR3, 15     ; 被乗数シフト用カウンタ
           ;              初期化
11         LAD  GR4, 1      ; GR4  16 - GR3
12         LD   GR6, GR1    ; 乗数を GR6 に設定,
           ;              GR1 は作業用に解放
13  LP     LAD  GR3, -1, GR3
14         LAD  GR4, 1, GR4 ; GR3 + GR4 = 16 の
           ;              関係を維持
15         SLL  GR6, 1
16         JZE  FIN
17         JPL  LP         ; 最上位ビットが0なら
           ;              何もしない
18         LD   GR1, H      ; 被乗数の取出し
19         LD   GR0, L
    
```

```

20         LD   GR5, GR0   ;
21         a           ;
22         SLL  GR1, 0, GR3 ;
23         SLL  GR0, 0, GR3 ;
24         OR   GR1, GR5   ;
25         CALL ADD32      ; シフトされた被乗数を
           ;              中間結果に加算
26         JUMP LP
27  FIN    RPOP
28         RET
29  H     DS   1          ; 被乗数上位語の退避場所
30  L     DS   1          ; " 下位語  "
31         ;
32  ADD32  RPUSH          ; Z   X + Y
33         ;              ; X の上位語を XH,
           ;              ; 下位語を XL などと表記
34         LD   GR4, GR0   ; YL を GR4 に設定
35         ADDL GR0, 1, GR2 ; ZL   XL + YL, ZL を
           ;              ; GR0 に設定
36         XOR  GR4, 1, GR2 ;
37         JMI  DIFF      ;
38         XOR  GR4, 1, GR2 ;
39         JUMP NEXT      ;
40  DIFF   XOR  GR4, GR0   ;
           ;              ; けた上がりの判定
           ;              ; XLとYLの最上位ビットが、
           ;              ; ともに1のときけた上がり
           ;              ; XLとYLの最上位ビットが
           ;              ; 異なり、ZLの最上位ビット
           ;              ; が0のときけた上がり
41  NEXT   SRL  GR4, 15   ; けた上がりを GR4 の
           ;              ; 最下位ビットへ移動
42         b
43         ADDL GR1, 0, GR2
44         ST   GR1, 0, GR2
45         ST   GR0, 1, GR2
46         RPOP
47         RET
48         END
    
```

設問1 プログラム中の a 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 SLL GR5, 0, GR3 | イ SLL GR5, 0, GR4 |
| ウ SRA GR5, 0, GR3 | エ SRA GR5, 0, GR4 |
| オ SRL GR5, 0, GR3 | カ SRL GR5, 0, GR4 |

b に関する解答群

- | | |
|-----------------|-----------------|
| ア ADDL GR0, GR4 | イ ADDL GR1, GR4 |
| ウ OR GR0, GR4 | エ OR GR1, GR4 |
| オ SUBL GR0, GR4 | カ SUBL GR1, GR4 |

設問2 次の記述中の a に入れる正しい答えを、解答群の中から選べ。

MULT32 の受け取る乗数を 16 ビット符号なし2進整数とするため、行番号 10、11を次のとおりに変更する。

示現塾 プロジェクトマネージャ・テクニカルエンジニア(ネットワーク)など各種セミナーを開催中!!

開催日、受講料、カリキュラム等、詳しくは、<http://zigen.cosmoconsulting.co.jp> 今すぐアクセス!!

```

10      LAD GR3,16      ; 被乗数シフト用
                          カウンタ初期化
11      LAD GR4,0       ; GR4 16-GR3
    
```

さらに、行番号 16, 17 を次の命令群で置き換える。

```

                          JUMP LP
ONBIT  NOP
    
```

解答群

- | | |
|------------------------|------------------------|
| ア JOV LP
JPL ONBIT | イ JOV ONBIT
JZE FIN |
| ウ JPL FIN
JOV ONBIT | エ JPL ONBIT
JZE FIN |
| オ JZE FIN
JOV ONBIT | カ JZE FIN
JPL ONBIT |

設問3 次の記述中の に入れる正しい答えを、解答群の中から選べ。

MULT32 を使用して、再帰的に階乗計算を行う副プログラム FACT を作成した。

(1) 正の整数 n の階乗 F(n) は、次式で求められる。

$$F(n) = n \times F(n-1) \quad (\text{ただし, } F(0) = 1)$$

$$\begin{aligned}
 \text{例: } F(3) &= 3 \times F(2) = 3 \times 2 \times F(1) \\
 &= 3 \times 2 \times 1 \times F(0) \\
 &= 3 \times 2 \times 1 \times 1 = 6
 \end{aligned}$$

(2) FACT は、n を GR1 に、計算結果を格納する領域の先頭番地を GR2 に設定して呼び出される。n は、その階乗 F(n) が 32 ビット符号なし 2 進整数の範囲に収まるように与えられる (1 ≤ n ≤ 12)。計算結果格納領域の形式は、MULT32 の被乗数の格納形式と同じとする。

(3) 副プログラムから戻るとき、汎用レジスタ GR1 ~ GR7 の内容は元に戻す。

(行番号)

```

1  FACT  START
2      RPUSH
3      CALL RMAIN      ; 再帰処理の本体を呼ぶ
4      RPOP
5      RET              ; 主プログラムに戻る
6  RMAIN LD  GR1,GR1   ; N=0?
7      JNZ  CONT      ; N=0 なら CONT へ
8      ST   GR1,0,GR2 ; N=0 の場合
9      LD   GR1,=1     ; F(0) の値 (=1) を
10     ST   GR1,1,GR2 ; 計算結果格納領域に設定し
11     RET              ; 行番号 15 に戻る
12  CONT PUSH 0,GR1   ; Nの退避(Nはn
                          n-1 ... 1の順に変化)
    
```

```

13      LAD GR1,-1,GR1 ; GR1 N-1
14      CALL RMAIN     ; F(N-1) を計算し計算
                          結果格納領域に設定
15      POP  GR1       ; N の復元 (N は 1
                          2 ... n の順に変化)
16      CALL MULT32    ; F(N-1) × N を計算し
                          計算結果格納領域に設定
17      RET              ; 再帰処理の終了時は
                          行番号 4 に、
18      END             ; それ以外のときは
                          行番号 15 に戻る
    
```

F(3) を計算するとき、行番号 6 のラベル RMAIN の命令には c 回制御が移る。また、行番号 11 の RET 命令は d 回実行される。

解答群

- | | | |
|-----|-----|-----|
| ア 1 | イ 2 | ウ 3 |
| エ 4 | オ 5 | カ 6 |

平成18年度 秋期 F E 午後問題 CASL

問9 次のアセンブラプログラムの説明及びプログラムを読んで、設問1, 2に答えよ。

〔プログラムの説明〕

2語の中にあるビット列を結合してレジスタに設定する副プログラム COMBINE と、レジスタ中のビット列を分割して2語に格納する副プログラム DIVIDE である。

(1) 図1に COMBINE と DIVIDE のデータ形式と結果を示す。ただし、m + n ≤ 16 である。

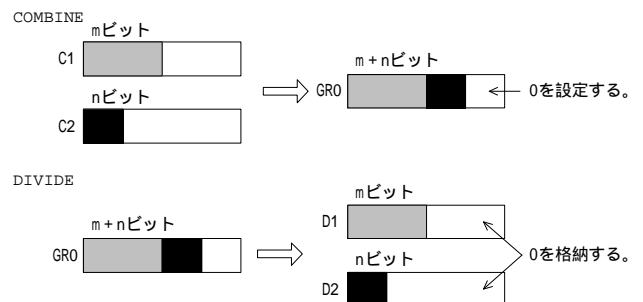


図1 COMBINE と DIVIDE のデータ形式と結果

(2) COMBINE は、GR1 ~ GR4 にそれぞれ次の内容が設定されて主プログラムから呼ばれる。

- GR1 : C1 のアドレス
- GR2 : C1 のビット列の長さ (m)
- GR3 : C2 のアドレス
- GR4 : C2 のビット列の長さ (n)

- (3) COMBINE は、結合した $m + n$ ビットを GR0 に左詰めで設定し、残りのビットには 0 を設定する。
- (4) DIVIDE は、GR0 ~ GR4 にそれぞれ次の内容が設定されて主プログラムから呼ばれる。
 GR0 : 分割するビット列
 GR1 : D1 のアドレス
 GR2 : D1 に格納するビット列の長さ (m)
 GR3 : D2 のアドレス
 GR4 : D2 に格納するビット列の長さ (n)
- (5) DIVIDE は、D1 には m ビットを、D2 には n ビットをそれぞれ左詰めで格納し、残りのビットには 0 を格納する。
- (6) COMBINE 及び DIVIDE から戻るとき、汎用レジスタ GR1 ~ GR7 の内容は元に戻す。

〔プログラム 1〕

```

COMBINE START
  RPUSH
  LD GR0,0,GR1
  LD GR5,=16
  SUBL GR5,GR2 ; 16 - m
  [ a ] ; }
  SLL GR0,0,GR5 ; } 左側mビット以外の
  ; } ビットを0にする。
;
  LD GR6,0,GR3
  LD GR7,=16
  SUBL GR7,GR4 ; 16 - n
  SRL GR6,0,GR7
  SUBL GR5,GR4 ; 16 - m - n
  [ b ]
  OR GR0,GR6
  RPOP
  RET
  END
    
```

〔プログラム 2〕

```

DIVIDE START
  RPUSH
  LD GR5,GR0
  LD GR6,=16
  SUBL GR6,GR2 ; 16 - m
  SRL GR0,0,GR6
  SLL GR0,0,GR6
  ST GR0,0,GR1
;
  SUBL GR6,GR4 ; 16 - m - n
  [ c ]
  LD GR6,=16
  SUBL GR6,GR4 ; 16 - n
  SLL GR5,0,GR6
  ST GR5,0,GR3
  RPOP
    
```

```

RET
END
    
```

設問 1 プログラム 1 及びプログラム 2 中の [] に入れる正しい答えを、解答群の中から選べ。

解答群

- | | | | | | |
|---|-----|-----------|---|-----|-----------|
| ア | SLL | GR5,0,GR6 | イ | SLL | GR6,0,GR5 |
| ウ | SRL | GR0,0,GR5 | エ | SRL | GR0,0,GR6 |
| オ | SRL | GR5,0,GR6 | カ | SRL | GR6,0,GR5 |

設問 2 次の記述中の [] に入れる正しい答えを、解答群の中から選べ。

DIVIDE において、データを格納しない部分は元の内容を保持したままとするためには、プログラム 2 中の [] の部分を次のように変更する必要がある。図 2 に変更後のデータ形式と結果を示す。

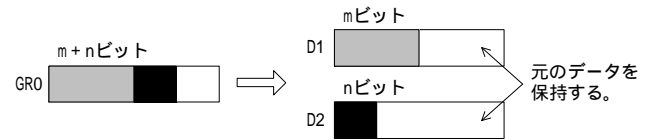


図 2 変更後のデータ形式と結果

処置	変更内容
を置換	LD GR7,=#FFFF SRL GR7,0,GR2 [d] ; 元のデータを取り出す。 OR GR0,GR7 ST GR0,0,GR1
を置換	LD GR7,=#FFFF SRL GR7,0,GR4 AND GR7,0,GR3 [e] ; データを結合する。 ST GR5,0,GR3

解答群

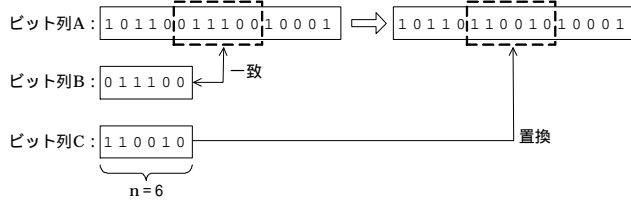
- | | | | | | |
|---|-----|-----------|---|-----|-----------|
| ア | AND | GR6,0,GR1 | イ | AND | GR6,0,GR2 |
| ウ | AND | GR7,0,GR1 | エ | AND | GR7,0,GR2 |
| オ | OR | GR5,GR6 | カ | OR | GR5,GR7 |
| キ | OR | GR7,GR5 | ク | OR | GR7,GR6 |

問 13 次のアセンブラプログラムの説明及びプログラムを読んで、設問 1, 2 に答えよ。

〔プログラムの説明〕

(1) 副プログラム BREP は、16 ビットからなるビット列 A の中に、 n ビットのビット列 B と一致する部分ビット列があれば、それを n ビットのビット列 C に置き換えるプログラムである。

例：



GR1 ~ GR4 にはそれぞれ次の内容を設定して，主プログラムから渡される。

GR1：ビット列 A を格納している語のアドレス

GR2：ビット列 B（左詰めで設定され，残りのビットには 0 が設定されている。）

GR3：ビット列 C（左詰めで設定され，残りのビットには 0 が設定されている。）

GR4：n (1 n 16)

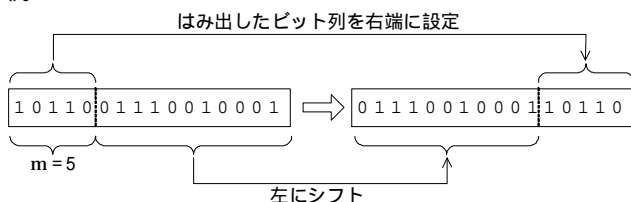
ビット列 A の中にビット列 B に一致する部分ビット列が複数個あれば，それらのすべてをビット列 C に置き換える。

ビット列 A の左端から検索する。ビット列 B に一致する部分ビット列をビット列 C に置き換えた場合，その部分ビット列の右隣のビットから検索を続ける。

副プログラムから戻るとき，汎用レジスタ GR1 ~ GR7 の内容は元に戻す。

(2) 副プログラム ROTSL は，16 ビットからなるビット列を m ビットだけ左に循環シフトするプログラムである。

例：



GR0 と GR1 にはそれぞれ次の内容を設定して，主プログラムから渡される。

GR0：m (0 m 16)

GR1：ビット列を格納している語のアドレス

副プログラムから戻るとき，汎用レジスタ GR1 ~ GR7 の内容は元に戻す。

〔プログラム〕

```
BREP START
  RPUSH
  LD GR5,=16 ; 未検査ビット数の初期化
```

```
LD GR6,=#8000 ; } マスクの作成
SRA GR6,-1,GR4 ; }
LOOP LD GR7,0,GR1 ; GR7 ビット列 A
      [a] ; 左端 n ビット以外の
      ; ビットを 0 にする。
CPL GR7,GR2 ; 左端 n ビットと
              ; ビット列 B を比較
JZE MATCH ; 一致
LD GR0,=1 ; 左循環シフトする
           ; ビット数
JUMP CONT
MATCH LD GR7,0,GR1 ; GR7 ビット列 A
      SLL GR7,0,GR4 ; } 左端 n ビットを
      SRL GR7,0,GR4 ; } 0 にする。
      [b] ; ビット列 C を左端 n
           ; ビットに設定
ST GR7,0,GR1 ; 元の領域に戻す。
LD GR0,GR4 ; 左循環シフトする
           ; ビット数
CONT CALL ROTSL ; ビット列 A を左循環
           ; シフト
SUBA GR5,GR0 ; 未検査ビット数の更新
CPA GR5,GR4 ; 未検査ビット数と n を
             ; 比較
      [c] ; n 未満であれば
           ; 終了処理へ
JUMP LOOP
FIN LD GR0,GR5 ; 未検査ビット数
    CALL ROTSL ; ビット位置を元に戻す。
    RPOP
    RET
;
ROTSL RPUSH
      LD GR3,GR0 ; GR3 m
      LD GR4,=16
      SUBA GR4,GR3 ; GR4 (16 - m)
      LD GR5,0,GR1
      LD GR6,GR5
      SLL GR5,0,GR3
      [d] ; はみ出すビット列
      OR GR5,GR6 ; はみ出すビット列を
                  ; 右端に設定
      ST GR5,0,GR1 ; 元の領域に戻す。
      RPOP
      RET
      END
```

設問 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, b に関する解答群

ア AND GR7,GR3 イ AND GR7,GR6
ウ OR GR7,GR3 エ OR GR7,GR6

c に関する解答群

ア JMI FIN イ JNZ FIN
ウ JPL FIN エ JZE FIN

d に関する解答群

- ア SLL GR6,0,GR3 イ SLL GR6,0,GR4
ウ SRL GR6,0,GR3 エ SRL GR6,0,GR4

設問 2 次の記述中の に入れる正しい答えを、
解答群の中から選べ。

主プログラムから渡されたビット列 A, B, C 及び n
が次のとおりであったとき、副プログラム BREP は副
プログラム ROTSL を 呼び出す。

ビット列 A: 0001110100111001
ビット列 B: 0111
ビット列 C: 1001
n: 4

解答群

- ア 2 イ 6 ウ 8
エ 12 オ 16

平成 18 年度 春期 F E 午後解答 CASL

問 9

設問 1

- a - 工 b - ケ c - 工

設問 2

ウ

問 13

設問 1

- a - カ b - イ

設問 2

イ

設問 3

- c - 工 d - ア

平成 18 年度 秋期 F E 午後解答 CASL

問 9

設問 1

- a - ウ b - イ c - オ

設問 2

- e - ウ f - カ

問 13

設問 1

- a - イ b - ウ c - ア d - 工

設問 2

ウ