

# 平成22年度後期 情報検定

<実施 平成23年2月13日（日）>

## プログラミングスキル

(説明時間 10:00~10:10)

(試験時間 10:10~11:40)

- ・試験問題は試験開始の合図があるまで開かないでください。
- ・解答用紙（マークシート）への必要事項の記入は、試験開始の合図と同時に行いますので、それまで伏せておいてください。
- ・試験開始の合図の後、次のページを開いてください。＜受験上の注意＞が記載されています。必ず目を通してから解答を始めてください。
- ・試験問題は、すべてマークシート方式です。正解と思われるものを1つ選び、解答欄の○をHBの黒鉛筆でぬりつぶしてください。2つ以上ぬりつぶすと、不正解になります。
- ・辞書、参考書類の使用および筆記用具の貸し借りは一切禁止です。
- ・電卓の使用が認められます。ただし、下記の機種については使用が認められません。

### <使用を認めない電卓>

1. 電池式（太陽電池を含む）以外の電卓
2. 文字表示領域が複数行ある電卓（計算状態表示の一行は含まない）
3. プログラムを組み込む機能がある電卓
4. 電卓が主たる機能ではないもの
  - \* パソコン（電子メール専用機等を含む）、携帯電話（PHS）、ポケットベル、電子手帳、電子メモ、電子辞書、翻訳機能付き電卓、音声応答のある電卓、電卓付腕時計等
5. その他試験監督者が不適切と認めるもの

## <受験上の注意>

1. この試験問題は28ページあります。ページ数を確認してください。  
乱丁等がある場合は、手をあげて試験監督者に合図してください。  
※問題を読みやすくするために空白ページを設けている場合があります。
2. 解答用紙（マークシート）に、受験者氏名・受験番号を記入し、受験番号下欄の数字をぬりつぶしてください。正しく記入されていない場合は、採点されませんので十分注意してください。
3. 試験問題についての質問には、一切答えられません。自分で判断して解答してください。
4. 試験中の筆記用具の貸し借りは一切禁止します。筆記用具が破損等により使用不能となった場合は、手をあげて試験監督者に合図してください。
5. 試験を開始してから30分以内は途中退出できません。30分経過後退出する場合は、もう一度、受験番号・マーク・氏名が記載されているか確認して退出してください。なお、試験終了5分前の合図以降は退出できません。試験問題は各自お持ち帰りください。
6. 合否通知の発送は平成23年3月中旬の予定です。
  - ①団体受験された方は、団体経由で合否の通知をいたします。
  - ②個人受験の方は、受験票に記載されている住所に郵送で合否の通知をいたします。
  - ③合否等の結果についての電話・手紙等でのお問い合わせには、一切応じられませんので、ご了承ください。

## <問題の構成>

必須問題 全員解答

問題 1 ～ 問題 4	2 ページ～13 ページ
-------------------------	--------------

選択問題 次の問題から 1 問選択し解答せよ。

(選択した問題は解答用紙「選択欄」に必ずマークすること)

※選択欄にマークがなく、解答のみマークした場合は採点を行いません。

・ C 言語の問題	16 ページ～19 ページ
・ 表計算の問題	20 ページ～23 ページ
・ アセンブラの問題	25 ページ～28 ページ

## 必須問題

問題 1 次の二分木構造に関する記述を読み、設問に答えよ。

二分木構造とは、1つの親ノードが最大2つの子ノードを持つデータ構造である。ここで、左部分木の全てのノードの値は、親ノードの値より小さく、右部分木の全てのノードの値は、親ノードの値より大きくなるデータ構造を考える。

なお、各ノードで同じ値は無いものとする。

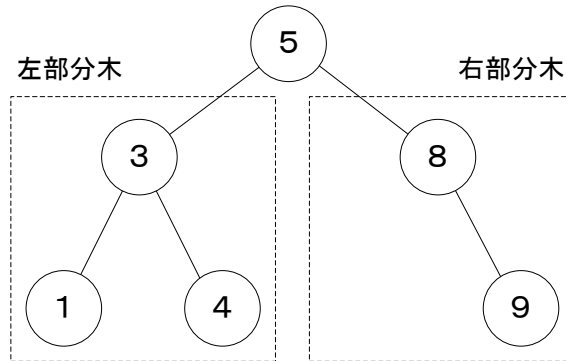


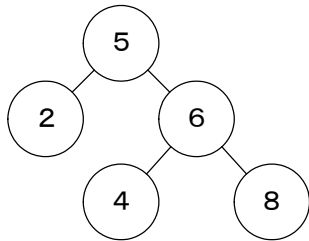
図 1 二分木構造

この構造に値の追加や削除を行う場合は、親ノードと子ノードの値の関係が維持できるように配置を変える場合がある。

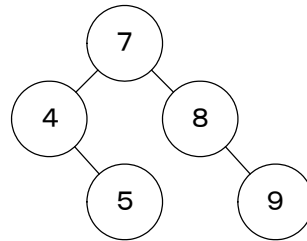
<設問 1> 二分木構造でないものを解答群から選べ。

(1) の解答群

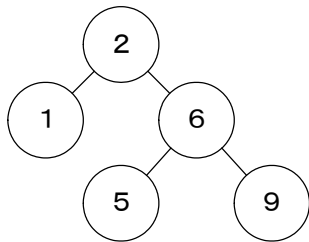
ア.



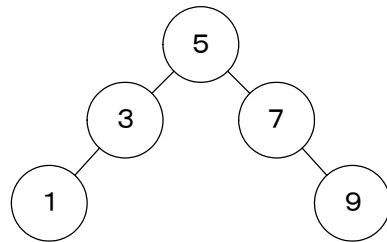
イ.



ウ.



エ.



<設問 2> 二分木構造へのデータの挿入に関する次の記述中の  に入れるべき適切な字句を解答群から選べ。

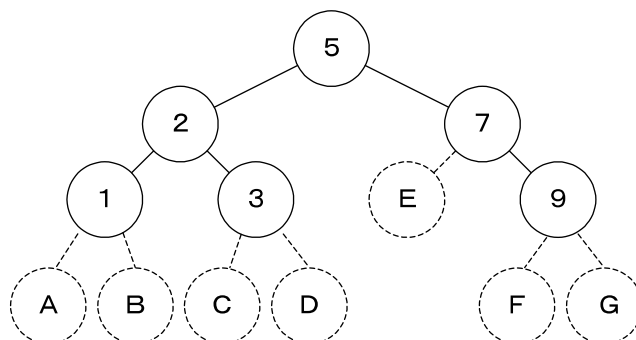


図 2 データを挿入する二分木構造

図 2 の二分木構造に値 4 を持つノードを挿入する場合は、  (2) に格納され、値 8 を持つノードを挿入する場合は、  (3) に格納される。

(2), (3) の解答群

- |      |      |      |      |
|------|------|------|------|
| ア. A | イ. B | ウ. C | エ. D |
| オ. E | カ. F | キ. G |      |

<設問 3> 次の二分木構造からのデータ削除に関する記述中の  に入れるべき、A と B の組み合わせを (4), C と D の組み合わせを (5) の解答群から選べ。

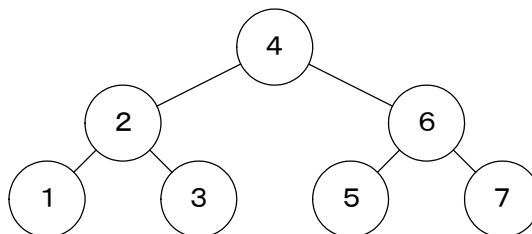


図 3 データ削除前の二分木構造

図 3 の二分木構造から値 4 を持つノードを削除する場合を考える。

値 4 を持つノードを削除すると、二分木構造の根にあたるノードが無くなるため、子ノードへのつながりが無くなってしまう。そこで、値 4 を持つノードの左部分木にあるノードの中で一番  A, または、値 4 を持つノードの右部分木にあるノードの中で一番  B を値 4 を持つノードがあった場所へ移動する。前者を選択すれば  C を、後者を選択すれば  D を、値 4 を持つノードがあった場所へ移動する。

(4) の解答群

	A	B
ア	大きい値	大きい値
イ	大きい値	小さい値
ウ	小さい値	大きい値
エ	小さい値	小さい値

(5) の解答群

	C	D
ア	1	5
イ	1	7
ウ	3	5
エ	3	7

問題2 次の流れ図の説明を読み、流れ図中の  に入れるべき適切な字句を解答群から選べ。

[流れ図の説明]

10進数の値と基数を入力し、基数変換した結果を表示する流れ図である。

入力する値と基数変換における桁の値については、以下のとおりである。

- ① 10進数は0~32767の範囲で入力し、それ以外はエラーとする。
- ② 基数は2~16の範囲で入力し、それ以外はエラーとする。
- ③ 基数が11以上の場合、10から15は、“A”~“F”を使用する。

流れ図で使用する配列については、以下のとおりである。

- ① 配列 `ch[i]` ( $i=0, 1, \dots, 15$ ) には、桁の値となる文字を順番に格納している。

	0	1	2	3	4	...	9	10	11	12	13	14	15
ch	"0"	"1"	"2"	"3"	"4"	...	"9"	"A"	"B"	"C"	"D"	"E"	"F"

図1 配列 `ch` の内容

- ② 配列 `Result[i]` ( $i=0, 1, \dots, 15$ ) には、基数変換後の結果を1桁ずつ、配列の最後から前の方へ順番に格納する。例えば、結果が“123”であれば、図2のように格納される。

	0	1	...	11	12	13	14	15
Result			...			"1"	"2"	"3"

図2 配列 `Result` へ格納される結果

図2のように格納された配列 `Result` の13番目~15番目までを順番に1文字ずつ改行しないで出力する。

なお、流れ図中における計算で、除算結果は小数点以下を切り捨てるものとする。

[流れ図]

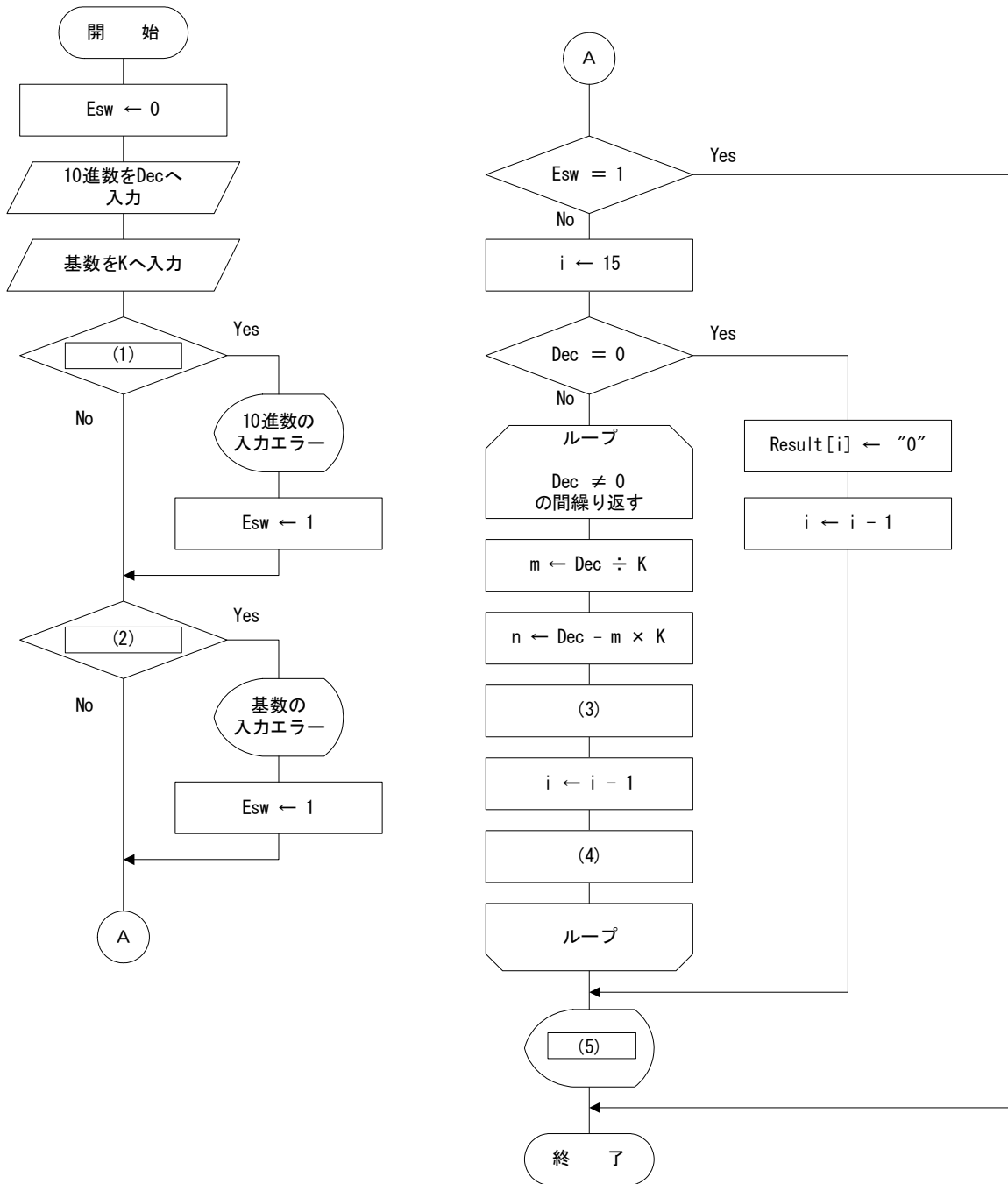


図3 流れ図



(1) の解答群

ア.  $Dec < 0$  または  $Dec > 32767$

ウ.  $Dec < 0$  かつ  $Dec > 32767$

イ.  $Dec < 1$  または  $Dec > 32767$

エ.  $Dec < 1$  かつ  $Dec > 32767$

(2) の解答群

ア.  $K < 1$  または  $K > 16$

ウ.  $K < 1$  かつ  $K > 16$

イ.  $K < 2$  または  $K > 16$

エ.  $K < 2$  かつ  $K > 16$

(3) の解答群

ア.  $Result[i] \leftarrow ch[Dec]$

ウ.  $Result[i] \leftarrow ch[m]$

イ.  $Result[i] \leftarrow ch[K]$

エ.  $Result[i] \leftarrow ch[n]$

(4) の解答群

ア.  $Dec \leftarrow i$

ウ.  $Dec \leftarrow m$

イ.  $Dec \leftarrow K$

エ.  $Dec \leftarrow n$

(5) の解答群

ア.  $Result[i] \sim Result[14]$  を順番に表示

イ.  $Result[i] \sim Result[15]$  を順番に表示

ウ.  $Result[i+1] \sim Result[14]$  を順番に表示

エ.  $Result[i+1] \sim Result[15]$  を順番に表示

問題3 次の流れ図の説明を読み、流れ図中の  に入れるべき適切な字句を解答群から選べ。

[流れ図の説明]

ある高校で行った全校生を対象にした一斉校内実力テストの平均点を集計して印字する流れ図である。

実力テストの成績は、成績ファイルとして図1の形式で保存されている。なお、実力テストは英語、数学、国語の3科目で、それぞれ英語、数学、国語という項目に点数が格納されている。

なお、このファイルは年、組、番号で昇順に整列済みであり、全員が3科目の試験を受けている。

年	組	番号	氏名	英語	数学	国語
---	---	----	----	----	----	----

図1 成績ファイルのレコード形式

集計結果は、図2のように、クラスごとの平均、学年の平均を印字し、最後に全体の平均を印字する。

	英語	数学	国語
1年1組	75.40	74.80	66.60
1年2組	82.00	72.70	80.50
⋮			
学年平均	76.25	75.58	76.83
2年1組	75.30	74.40	71.40
⋮			
学年平均	72.83	74.28	72.60
3年1組	74.70	76.00	72.80
⋮			
学年平均	77.00	73.25	75.05
全体平均	75.36	74.37	74.83

図2 出力する集計結果

[流れ図]

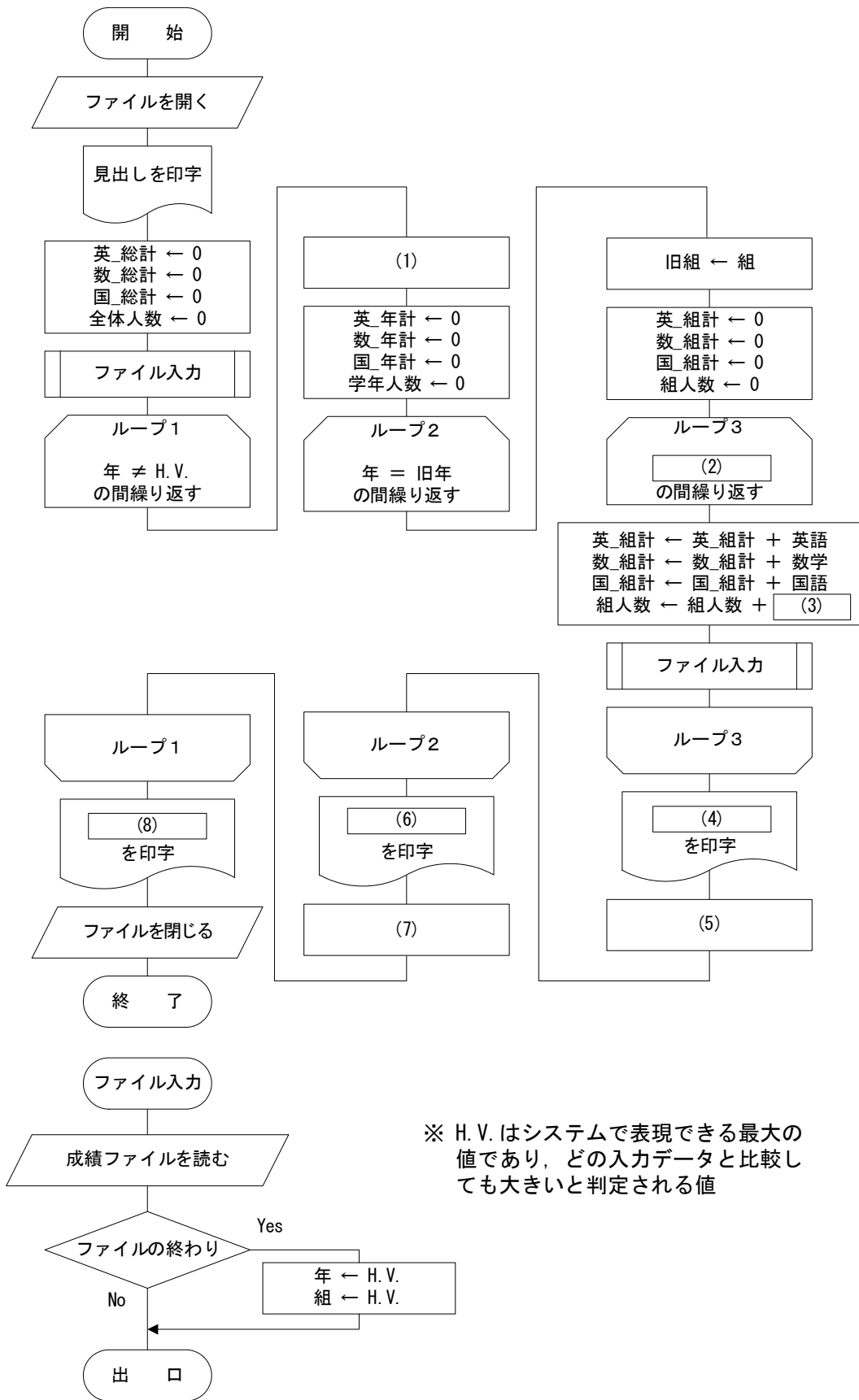


図3 流れ図

(1) の解答群

ア. 旧年  $\leftarrow 0$

ウ. 旧年  $\leftarrow$  年 + 1

イ. 旧年  $\leftarrow$  年

エ. 旧年  $\leftarrow$  H.V.

(2) の解答群

ア. 組  $\neq$  H.V.

ウ. 年 = 旧年 and 組 = 旧組

イ. 組 = 旧組

エ. 年 = 旧年 or 組 = 旧組

(3) の解答群

ア. 1

ウ. 番号

イ. 3

エ. 組\_人数

(4) , (6) , (8) の解答群

ア. "全体平均", 英\_総計  $\div$  学年人数, 数\_総計  $\div$  学年人数, 国\_総計  $\div$  学年人数

イ. "全体平均", 英\_総計  $\div$  全体人数, 数\_総計  $\div$  全体人数, 国\_総計  $\div$  全体人数

ウ. "学年平均", 英\_年計  $\div$  学年人数, 数\_年計  $\div$  学年人数, 国\_年計  $\div$  学年人数

エ. "学年平均", 英\_年計  $\div$  全体人数, 数\_年計  $\div$  全体人数, 国\_年計  $\div$  全体人数

オ. 年, 組, 英\_組計  $\div$  組人数, 数\_組計  $\div$  組人数, 国\_組計  $\div$  組人数

カ. 旧年, 旧組, 英\_組計  $\div$  組人数, 数\_組計  $\div$  組人数, 国\_組計  $\div$  組人数

(5) , (7) の解答群

ア. 英\_年計  $\leftarrow$  英\_年計 + 英語

数\_年計  $\leftarrow$  数\_年計 + 数学

国\_年計  $\leftarrow$  国\_年計 + 国語

学年人数  $\leftarrow$  学年人数 + 組人数

イ. 英\_年計  $\leftarrow$  英\_年計 + 英\_組計

数\_年計  $\leftarrow$  数\_年計 + 数\_組計

国\_年計  $\leftarrow$  国\_年計 + 国\_組計

学年人数  $\leftarrow$  学年人数 + 組人数

ウ. 英\_総計  $\leftarrow$  英\_総計 + 英\_年計

数\_総計  $\leftarrow$  数\_総計 + 数\_年計

国\_総計  $\leftarrow$  国\_総計 + 国\_年計

全体人数  $\leftarrow$  全体人数 + 学年人数

エ. 英\_総計  $\leftarrow$  英\_総計 + 英\_組計

数\_総計  $\leftarrow$  数\_総計 + 数\_組計

国\_総計  $\leftarrow$  国\_総計 + 国\_組計

全体人数  $\leftarrow$  全体人数 + 学年人数

問題4 次のプログラムの説明および疑似言語の記述形式の説明を読み、設問に答えよ。

[プログラムの説明]

1次元配列の中に格納されたデータの中から二分探索法によりデータを探索する関数 Bsearch である。ただし、1次元配列の内容は昇順に整列済みであり、要素位置は0から始まる。

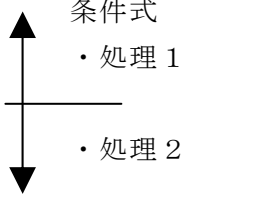
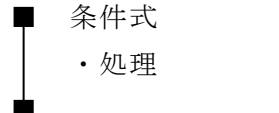
二分探索法とは、昇順に整列済みである配列を利用した探索方法で、探索しようとする値と1次元配列の中央の値を比べ、その大小関係によって探索範囲を狭くして目的のデータを検索するものである。

- ① 1次元配列の一番小さい要素位置をL、一番大きい要素位置をHとする。
- ② LとHを加えた値を2で割り、1次元配列の中央の要素位置とする。なお、割り算の結果は小数点以下を切り捨てる。
- ③ 1次元配列の中央の要素と探索する値を比べ、同じ値であればデータが見つかったことになる。同じ値でなければ、探索する値の方が大きい場合はLに中央の要素位置より1つ大きい値を、そうでなければHに中央の要素位置より1つ小さい値を代入して②へ戻る。
- ④ ②と③の処理をデータが見つかるまで、または、LがHより大きくなるまで繰り返す。

表 Bsearch の引数の仕様

変数名	入力／出力	意味
NUM	入力	探索する値
SIZE	入力	1次元配列の要素数
DT[]	入力	データが格納された1次元配列
返却値	出力	検索した場所 (ret ≥ 0) 1次元配列中に存在しなければ-1とする

[疑似言語の記述形式の説明]

記述形式	説明
○	手続き, 変数などの名前, 型などを宣言する
・変数 ← 式	変数に式の値を代入する
{文}	注釈を記述する
 条件式 ・処理 1 ・処理 2	選択処理を示す。 条件式が真の時は処理 1 を実行し, 偽の時は処理 2 を実行する。
 条件式 ・処理	前判定繰り返し処理を示す。 条件式が真の間, 処理を実行する。

[プログラム]

○整数型 : Bsearch (整数型 : NUM, 整数型 : SIZE, 整数型 : DT[])

○整数型 : H, L, M, Ret

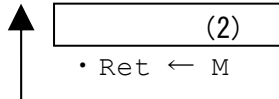
・ (1)

・ L ← 0

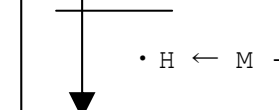
・ H ← SIZE - 1

■ (Ret = -1) and (L ≤ H)

・ M ← ( L + H ) / 2 { 小数点以下切り捨て } ①


  
(2)  
 ・ Ret ← M


  
(3)  
 ・ L ← M + 1


  
 ・ H ← M - 1

■  
 ・ Return Ret

<設問 1> プログラム中の  に入れるべき適切な字句を解答群から選べ。

(1) の解答群

ア. Ret ← -1

イ. Ret ← 0

ウ. Ret ← 1

エ. Ret ← SIZE - 1

(2) , (3) の解答群

ア. DT[M] = NUM

イ. DT[M] < NUM

ウ. DT[M] > NUM

エ. DT[Ret] = NUM

オ. DT[Ret] < NUM

カ. DT[Ret] > NUM

<設問 2> Bsearch に与える引数の内容が、次のような場合、プログラム中のⒶが実行される回数を答えよ。

NUM 30

SIZE 10

DT	2	4	7	9	11	17	21	30	35	40
----	---	---	---	---	----	----	----	----	----	----

(4) の解答群

ア. 1

イ. 2

ウ. 3

エ. 4

<設問 3> 次のデータ探索に関する記述中の  に入れるべき適切な字句を解答群から選べ。

二分探索は、配列の先頭からデータを探索する線形探索に比べると高速な探索が可能である。

100 個の要素からなる配列からデータを探索する時に行う、配列要素の値と探索するデータの比較回数を比べてみると、線形探索の場合は、最大 100 回の比較を行うが、二分探索では最大 7 回である。

また、配列の要素数が 200 になった場合、線形探索では最大 200 回の比較を行い、二分探索の場合は最大  (5) 回行う。

(5) の解答群

ア. 7

イ. 8

ウ. 9

エ. 14

これより

## < 選 択 問 題 >

選択問題は次の問題から 1 問選択し解答せよ。

選択した問題は必ず、解答用紙「選択欄」にマークすること。

※選択欄にマークがなく、解答のみの場合は採点を行いません。

各構成は以下のとおり。

### 選択問題

- |            |               |
|------------|---------------|
| ・ C 言語の問題  | 16 ページ～19 ページ |
| ・ 表計算の問題   | 20 ページ～23 ページ |
| ・ アセンブラの問題 | 25 ページ～28 ページ |



問題を読みやすくするために、  
このページは空白にしてあります。

## 選択問題 C言語の問題

次のC言語プログラムの説明を読み、プログラム中の□□□□に入れるべき適切な命令を解答群から選べ。

[プログラムの説明]

逆ポーランド記法で表現された式の計算を行う関数 Polish である。

逆ポーランド記法は、演算を行う2数の後に演算子を書くもので、演算子の優先順位に関係なく、式の先頭から順に計算を行えるものである。

(例)

通常表記	逆ポーランド記法
$1 + 2$	1 2 +
$(1 + 2) * (3 + 4)$	1 2 + 3 4 + *

逆ポーランド記法で表現された式を計算するには、スタックを用いる。

数式の見出しから順番に要素を取り出し、数値であればスタックへ格納し、演算子であればスタックから2つのデータを取り出して演算を行う。

例えば、スタックから取出したデータが、1番目が「3」、2番目が「4」、演算子が「-」であれば、 $4 - 3$ という計算になる。

数値演算結果はスタックへ格納する。例えば、「 $2 5 + 1 3 + *$ 」を計算する場合は、次のようになる。

- ① 「2」をスタックへ格納する。
- ② 「5」をスタックへ格納する。
- ③ 「+」なのでスタックの先頭から2つのデータを取り出し、「 $2 + 5$ 」の結果をスタックへ格納する。
- ④ 「1」をスタックへ格納する。
- ⑤ 「3」をスタックへ格納する。
- ⑥ 「+」なのでスタックの先頭から2つのデータを取り出し、「 $1 + 3$ 」の結果をスタックへ格納する。
- ⑦ 「\*」なのでスタックの先頭から2つのデータを取り出し、「 $7 * 4$ 」の結果をスタックへ格納する。

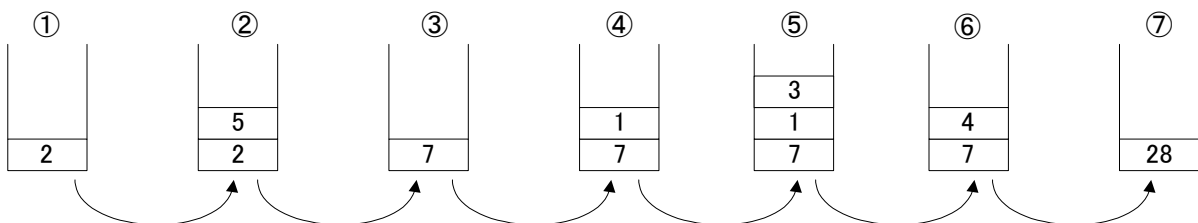


図1 逆ポーランド記法の計算とスタックの遷移

最後にスタックの先頭に格納されている値が計算結果となる。

このプログラムでは、数値の前後を “[” と “]” で囲み、負数を表す場合はハイフン ( - ) を先頭に付ける。ただし、“[” と “]” で囲んだ中では、小数点以下の数値を扱わない。

(入力例) [12][-34]+[43][25]-\*

また、このプログラムを実行する場合に、スタック用配列の大きさは十分な容量を持っており、スタックのオーバーフローは起こらないものとする。プログラムでは、スタック領域として配列 STACK、スタック領域の添え字として SP を使用する。SP の初期値は、関数 Polish が呼び出される前に 0 が設定される。

なお、関数 Polish に渡される逆ポーランド記法の式は、必ず正しいものとし、0 による除算やオーバーフローなどのエラーは起こらないものとする。

#### [関数の説明]

##### **Polish** 関数

引 数：文字列。

機 能：逆ポーランド記法で表現された式を計算する。なお、文字列の最後は必ず '¥0' である。

戻り値：計算結果。

##### **Push** 関数

引 数：数値。

機 能：スタックに引数の数値を格納する。

戻り値：なし。

##### **Pop** 関数

引 数：なし。

機 能：スタックから数値を取出す。

戻り値：数値。

#### [プログラム]

```
float Polish( unsigned char * );
void Push( float );
float Pop( void );

float STACK[SIZE]; /* SIZEはプログラム実行に十分な大きさである */
int SP;

float Polish( unsigned char *buff ) {
    unsigned int i, num, num_sw, sign;
```

```

float x, y;
i = 0;          /* 引数で受取った式の添え字 */
num_sw = 0;    /* 数値領域を示すスイッチ */
sign = 0;      /* 負数を示すスイッチ */
while(  ) {
    switch (buff[i]) {
        case '[' : num = 0;
                    num_sw = 1;
                    ;
                    break;

        case ']' : if (sign == 1) ; /* 負数にする */
                    Push( num );
                    num_sw = 0;
                    break;

        case '+' : y = Pop();
                    x = Pop();
                    Push(x + y);
                    break;

        case '-' : if (  ) /* 数値領域の負数か */
                    sign = 1;
                    else {
                        y = Pop();
                        x = Pop();
                        Push(x - y);
                    }
                    break;

        case '*' : y = Pop();
                    x = Pop();
                    Push(x * y);
                    break;

        case '/' : y = Pop();
                    x = Pop();
                    Push(x / y);
                    break;

        default : ; /* 数値への変換 */
    }
    ; /* 次の文字を取出す準備 */
}
return Pop();
}

```

```

/* スタックポインタを1つ上げてスタックヘデータを格納する */
void Push( float d ) {
    SP++;
    STACK[SP] = d;
}

```

```

/* スタックからデータを取り出してスタックポインタを1つ下げる */
float Pop( void ) {
    float ret;
    ret = STACK[SP];
    (7);
    return ret;
}

```

(1) の解答群

ア. <code>buff[i] == '¥0'</code>	イ. <code>buff[i] != '¥0'</code>
ウ. <code>buff[i] == EOF</code>	エ. <code>buff[i] != EOF</code>

(2) の解答群

ア. <code>sign = 0</code>	イ. <code>sign = 1</code>
ウ. <code>sign = 0 - sign</code>	エ. <code>sign = 1 - sign</code>

(3) の解答群

ア. <code>num = num * 0</code>	イ. <code>num = num - 1</code>
ウ. <code>num = 0 - num</code>	エ. <code>num = 1 - num</code>

(4) の解答群

ア. <code>num_sw == 0</code>	イ. <code>num_sw == 1</code>
ウ. <code>num_sw &gt; 1</code>	エ. <code>num_sw &lt; 0</code>

(5) の解答群

ア. <code>num = num + buff[i] - '0'</code>
イ. <code>num = num * 10 + buff[i]</code>
ウ. <code>num = num * 10 + buff[i] - '0'</code>
エ. <code>num = (num + buff[i] - '0') * 10</code>

(6) , (7) の解答群

ア. <code>i++</code>	イ. <code>i--</code>
ウ. <code>SP++</code>	エ. <code>SP--</code>
オ. <code>STACK[SP]++</code>	カ. <code>STACK[SP]--</code>

次の表計算ソフトに関する設問に答えよ。

この問題で使用する表計算ソフトの仕様は下記のとおりである。

AVERAGE 関数

範囲に含まれる数値の平均値を返す。

書式：AVERAGE (範囲)

SUM 関数

範囲に含まれる数値の合計値を返す。

書式：SUM (範囲)

IF 関数

条件が真の時に真の場合、偽の時に偽の場合の計算結果や値を返す。

書式：IF (条件式, 真の場合, 偽の場合)

COUNT 関数

指定された範囲に含まれるセルのうち、数値が入っているセルの個数を返す。

書式：COUNT (範囲)

COUNTIF 関数

指定された範囲に含まれるセルのうち、検索条件に一致するセルの個数を返す。

ただし、検索条件に数値を用いた比較を記述した場合は、数値の入っているセルのみが集計の対象となる。

書式：COUNTIF (範囲, 検索条件)

式

=に続けて計算式や関数などを入力する。

セル番地の参照

セル番地に\$を付けることで、絶対番地（絶対参照）を表す。

別シートの参照

ワークシート名に「!」を付けてセル位置を指定することにより別シートを参照できる。

例：「集計シート」のセル A1 を参照する場合は、集計シート!A1 と記述する。

J学院では、ある検定試験の結果を集計するために、表計算ソフトを用いて資料を作成することにした。

[検定試験の実施概要]

- ・科目1から科目4の4科目からなる試験である。
- ・各科目とも25点満点で、16点以上で科目合格となる。
- ・4科目すべて合格すると検定試験の合格となる。
- ・科目合格制度があり、過去1年の間に合格した科目は受験しなくてもよい。

今年の12月に行われた試験を表1の学生別成績表にまとめた。

表1 学生別成績表

	A	B	C	D	E	F	G	H
1	学生番号	氏名	科目1	科目2	科目3	科目4	平均	判定
2	J220101	相田 悟	20	20	12	25	19.25	
3	J220102	井上 晴	22	13	18	0	13.25	
4	J220103	梅田 金吾	19	21	16	12	17.00	
5	J220104	江藤 守	20	16	12	24	18.00	
6	J220105	小野 玲	科目合格	14	21	13	16.00	
7	J220106	甲木 英太	13	13	16	科目合格	14.00	
8	J220107	久藤 真一	10	9	11	欠席	10.00	
9	J220108	久木田 昇	21	17	16	22	19.00	合格
10	J220109	駒田 清	科目合格	科目合格	欠席	欠席		
11	J220110	斎藤 忍	科目合格	17	科目合格	19	18.00	合格
12	J220111	佐々木 香	15	7	22	16	15.00	
13	J220112	須藤 夢	16	科目合格	22	22	20.00	合格
14	J220113	内藤 信二	21	15	18	19	18.25	
15	J220114	新井田 太郎	13	11	22	22	17.00	
16	J220115	布部 佳子	欠席	欠席	欠席	欠席		
17	J220116	野村 大輔	14	4	11	15	11.00	
18	J220117	羽場 二郎	13	8	9	欠席	10.00	
19	J220118	武藤 恵一	15	16	11	欠席	14.00	
20	J220119	目黒 祐樹	21	25	21	22	22.25	合格
21	J220120	矢田 瞳	23	科目合格	科目合格	22	22.50	合格

学生ごとに科目1～科目4の点数を入力した。ただし、科目合格がある場合は「科目合格」、欠席した場合は「欠席」と入力した。

<設問1> 学生別成績表の作成に関する次の記述中の  に入るべき適切な式を解答群から選べ。

セルG2に平均を計算する式を入力し、セルG3～G21にコピーした。欠席科目および科目合格の科目は計算対象から除外するものとし、1科目も受験していない場合(セルの表示が欠席と科目合格のみ)は空白を表示する。

セルG2に入力する式は、次のようになる。

= IF(  (1) ,  (2) , "" )

また、セル H2 に合否判定をするため、次の式を入力してセル H3～H21 にコピーした。

=IF( (3), "合格", "")

(1) の解答群

- ア. COUNT(C2:F2) > 0
- ウ. COUNT(C2:F2) < 4

- イ. COUNT(\$C\$2:\$F\$2) > 0
- エ. COUNT(\$C\$2:\$F\$2) < 4

(2) の解答群

- ア. AVERAGE(C2:F2)
- ウ. SUM(C2:F2) / 4

- イ. AVERAGE(\$C\$2:\$F\$2)
- エ. SUM(\$C\$2:\$F\$2) / 4

(3) の解答群

- ア. COUNTIF(C2:F2, ">=16") + COUNTIF(C2:F2, "科目合格") = 4
- イ. COUNTIF(C2:F2, ">=16") + COUNTIF(C2:F2, "科目合格") > 4
- ウ. COUNTIF(C2:F2, ">=16") + COUNTIF(C2:F2, "欠席") < 4
- エ. COUNTIF(C2:F2, ">=16") + COUNTIF(C2:F2, "") = 4

<設問 2> スコア別成績一覧表の作成に関する次の記述中の            に入るべき適切な式を解答群から選べ。

表 1 の学生別成績表を基にして、表 2 のスコア別成績一覧表を作成する。

表 2 スコア別成績一覧表

	A	B	C	D	E	F	G	H	I
1	スコア	科目 1	科目 2	科目 3	科目 4	科目 1 比率	科目 2 比率	科目 3 比率	科目 4 比率
2	25～21	5	2	5	7	31.3%	12.5%	31.3%	50.0%
3	20～16	4	5	5	3	25.0%	31.3%	31.3%	21.4%
4	15～11	6	5	5	3	37.5%	31.3%	31.3%	21.4%
5	10～6	1	3	1	0	6.3%	18.8%	6.3%	0.0%
6	5～0	0	1	0	1	0.0%	6.3%	0.0%	7.1%
7									
8	合格者	12	10	12	11				

表 2 のセル B2～E6 の各セルは、対象のスコアの人数を表 1 から計算する。

なお、表 2 に入力する式の中で使われる COUNTIF の条件式において、数値との比較を条件とする場合は、「欠席」や「科目合格」の文字列は集計の対象とならない。

まず、セル B2 には以下の式を入力した。

= COUNTIF( 学生別成績表!C\$2:C\$21, ">=21" )



次に、セル B3 に次の式を入力した。

= COUNTIF( 学生別成績表!C\$2:C\$21,  ) -

セル B3 に入力した式はセル B4 と B5 にコピーし、 の条件式を変更した。

続けて、セル B6 に次の式を入力した。

= COUNT( 学生別成績表!C\$2:C\$21 ) -

セル B2～B6 に入力した式は、セル C 列～E 列の同じ行にコピーした。

セル F2～I6 のセルには、欠席者と科目合格者を除いた、各科目の受験者数に対する各スコアの人数の割合を計算する。セル F2 に次の式を入力し、セル F3～F6 および G2～I6 にコピーした。なお、これらのセルの表示形式はパーセント表示にした。

= B2 /

最後に、すでに科目合格となっている学生も含めた、科目ごとの合格者数を表示するために次の式をセル B8 に入力し、セル C8～E8 にコピーした。

=  + COUNTIF( 学生別成績表!C2:C21, "=科目合格" )

#### (4) の解答群

ア. ">=10"

イ. ">=11"

ウ. ">=15"

エ. ">=16"

#### (5) , (6) の解答群

ア. SUM(B2:B2)

イ. SUM(B\$2:B2)

ウ. SUM(\$B\$2:B2)

エ. SUM(B2:B5)

オ. SUM(\$B2:\$B5)

カ. SUM(\$B\$2:\$B\$5)

#### (7) の解答群

ア. COUNT(B2:B6)

イ. COUNT(B\$2:B\$6)

ウ. SUM(B2:B6)

エ. SUM(B\$2:B\$6)

#### (8) の解答群

ア. COUNT(B2:B3)

イ. COUNT(\$B\$2:\$B\$3)

ウ. SUM(B2:B3)

エ. SUM(\$B\$2:\$B\$3)

問題を読みやすくするために、  
このページは空白にしてあります。

## 選択問題 アセンブラの問題

次のアセンブラ言語CASL II プログラムの説明を読み、各設問に答えよ。

[プログラムの説明]

携帯電話の機種を検索する副プログラムである。希望する機能と一致した機種を携帯電話の仕様表から検索し、希望条件に該当する機種のコードを指定された番地以降にすべて格納する。

携帯電話の仕様表は機種コードと仕様がそれぞれ1語ずつ連続する領域に格納され、合計  $n$  機種分  $2n$  語の領域に格納されている。

PARAM 表には希望仕様、仕様表の先頭番地、仕様表の機種数( $n$ )、検索結果を格納する先頭番地が下記のように設定される。PARAM 番地の先頭番地が GR1 に設定され、副プログラムに渡される。

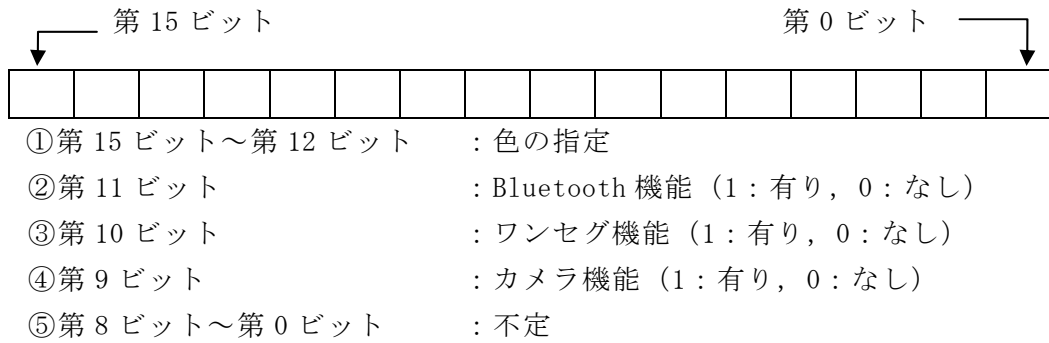


図 1 携帯電話の仕様

表 1 携帯電話の仕様表

機種コード 1
機種コード 1 の仕様
機種コード 2
機種コード 2 の仕様
⋮
機種コード n
機種コード n の仕様

表 2 PARAM 表

PARAM+0	希望仕様
+1	携帯電話の仕様表の先頭番地
+2	機種数(n)
+3	検索結果を格納する先頭番地

(※)PARAM 番地が GR1 に設定されて、副プログラムが呼び出される。

[プログラム]

行番号	ラベル	命令	オペランド	コメント
100	<b>SUBRTN</b>	<b>START</b>		
110		<b>LD</b>	<b>GR0,0,GR1</b>	;希望仕様の取得
120		<b>AND</b>	<b>GR0,#FE00</b>	
130		<b>LD</b>	<b>GR2,1,GR1</b>	;仕様表の先頭番地の取得
140		<b>LD</b>	<b>GR3,2,GR1</b>	;機種数(n)の取得
150			(1)	
160		<b>ADDL</b>	<b>GR3,GR2</b>	
170		<b>LD</b>	<b>GR4,3,GR1</b>	;検索結果を格納する先頭番地の取得
180	<b>LOOP</b>	<b>LD</b>	<b>GR5,1,GR2</b>	
190		<b>AND</b>	<b>GR5,#FE00</b>	
200		<b>LD</b>	<b>GR6,GR0</b>	
210			(2)	
220		<b>JNZ</b>	<b>L1</b>	
230		<b>LD</b>	<b>GR7,0,GR2</b>	
240			(3)	
250		<b>LAD</b>	<b>GR4,1,GR4</b>	
260	<b>L1</b>		(4)	
270		<b>CPL</b>	<b>GR2,GR3</b>	
280		<b>JMI</b>	<b>LOOP</b>	
290		<b>RET</b>		
300		<b>END</b>		

<設問 1> プログラム中の  に入れるべき適切な命令を解答群から選べ。

(1) の解答群

ア. **ADDA GR3,=1**

イ. **ADDA GR3,=2**

ウ. **SLA GR3,1**

エ. **SLA GR3,2**

(2) の解答群

- ア. AND GR5,GR6
- ウ. SRL GR5,0,GR6

- イ. OR GR5,GR6
- エ. XOR GR5,GR6

(3) の解答群

- ア. ST GR7,0,GR2
- ウ. ST GR7,0,GR4

- イ. ST GR7,1,GR2
- エ. ST GR7,1,GR4

(4) の解答群

- ア. LAD GR2,1,GR2
- ウ. LAD GR3,1,GR3

- イ. LAD GR2,2,GR2
- エ. LAD GR3,2,GR3

<設問 2> 希望仕様と携帯電話の仕様表が下記のように設定されていた場合、検索される仕様の組合せを解答群から選べ。

[希望仕様] 

#AE00
-------

[携帯電話の仕様表]

#AB01	… A
#BC2B	… B
#AC03	… C
#AE5C	… D
#AD02	… E
#AE2F	… F
#AE00	… G
#AC01	… H
#AF02	… I
#AEBD	… J

(5) の解答群

- ア. CとE
- ウ. CとEとI

- イ. CとG
- エ. EとGとI

<設問 3> 行番号120と190は仕様の中から有効なビットを取り出すための処理である。行番号120と同じ結果になるように、以下のように命令を書き換えた。



に入れるべき命令を解答群から選べ。

AND GR0,=#FE00 → 

(6)
(7)

(6) , (7) の解答群

ア. **SLA GR0,8**

ウ. **SLA GR0,9**

オ. **SRA GR0,8**

キ. **SRA GR0,9**

イ. **SLL GR0,8**

エ. **SLL GR0,9**

カ. **SRL GR0,8**

ク. **SRL GR0,9**

<メモ欄>

