

# 平成23年度後期 情報検定

<実施 平成24年2月12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）、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. 2011年度の受験者から、試験後にお知らせする合否結果（合否通知）、および合格者に交付する「合格証・認定証」はすべて、Webページ（PC、モバイル）での認証によるデジタル「合否通知」、デジタル「合格証・認定証」に移行しました。
  - ①団体宛にはこれまでと同様に合否結果一覧ほか、試験結果資料一式を送付します。
  - ②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	15 ページ～18 ページ
・ 表計算の問題	19 ページ～23 ページ
・ アセンブラの問題	24 ページ～28 ページ

## 必須問題

問題 1 次の決定表に関する記述を読み、決定表および設問中の  に入れるべき適切な字句を解答群から選べ。

J社ではネットバンキングを利用して、仕入先への銀行振込みを行っている。次に示す振込手数料を、決定表を利用して整理した。

法人契約の場合の振込手数料は次のとおりである。

- ・振込先がJ社の口座と同一銀行同一支店の場合、無料である。
- ・振込先がJ社の口座と同一銀行だが、支店が異なる場合は、振込額によって異なり、3万円未満の場合は105円、3万円以上の場合は315円である。
- ・振込先がJ社の口座と異なる銀行の場合は、支店に関係なく振込額によって決定する。3万円未満の場合は420円、3万円以上の場合は630円である。

表 振込手数料の決定表

同一銀行	Y				
同一支店	Y	(1)	(2)	(3)	(4)
振込額3万円未満	—				
手数料 0円	X				
手数料 105円		X			
手数料 315円			X		
手数料 420円				X	
手数料 630円					X

(1) ~ (4) の解答群

ア. 

Y
N
Y

イ. 

Y
N
N

ウ. 

Y
—
Y

エ. 

Y
—
N

オ. 

N
Y
Y

カ. 

N
Y
N

キ. 

N
—
Y

ク. 

N
—
N

<設問> 次の2件の振込み手数料の合計金額は  (5) である。

- 1 他銀行宛て25,000円
- 2 同一銀行他支店30,000円

(5) の解答群

ア. 525円

イ. 735円

ウ. 945円

エ. 1,050円

問題2 次の流れ図の説明を読み、設問に答えよ。

[流れ図の説明]

売上ファイルを入力し、売上一覧表を作成する流れ図である。

1. マスタファイルは順編成ファイルである。レコード形式は図1のようになっており、第1キー店舗コード、第2キー商品コードの昇順に整列されている。

キー項目		売上金額
店舗コード	商品コード	

図1 売上ファイルのレコード形式

2. 売上一覧表は図2のようになっている。

売上一覧表		
店舗コード	商品コード	売上金額
T01	S01	¥23,800
	S03	¥107,600
	:	:
	:	:
	店舗計	¥4,152,900
T02	S02	¥282,500
	:	:
	:	:
	店舗計	¥3,740,300
	総合計	¥72,619,400

図2 売上一覧表の形式

3. キーの退避領域は、図3のようになっている。

Wキー	
W店舗	W商品

図3 キーの退避領域

<設問> 流れ図中の  に入れるべき適切な字句を解答群から選べ。なお、流れ図中で使われている最大値とは、どの値と比較しても大きいと判断される値であり、マスタファイルのレコード中に存在しない。

[流れ図]

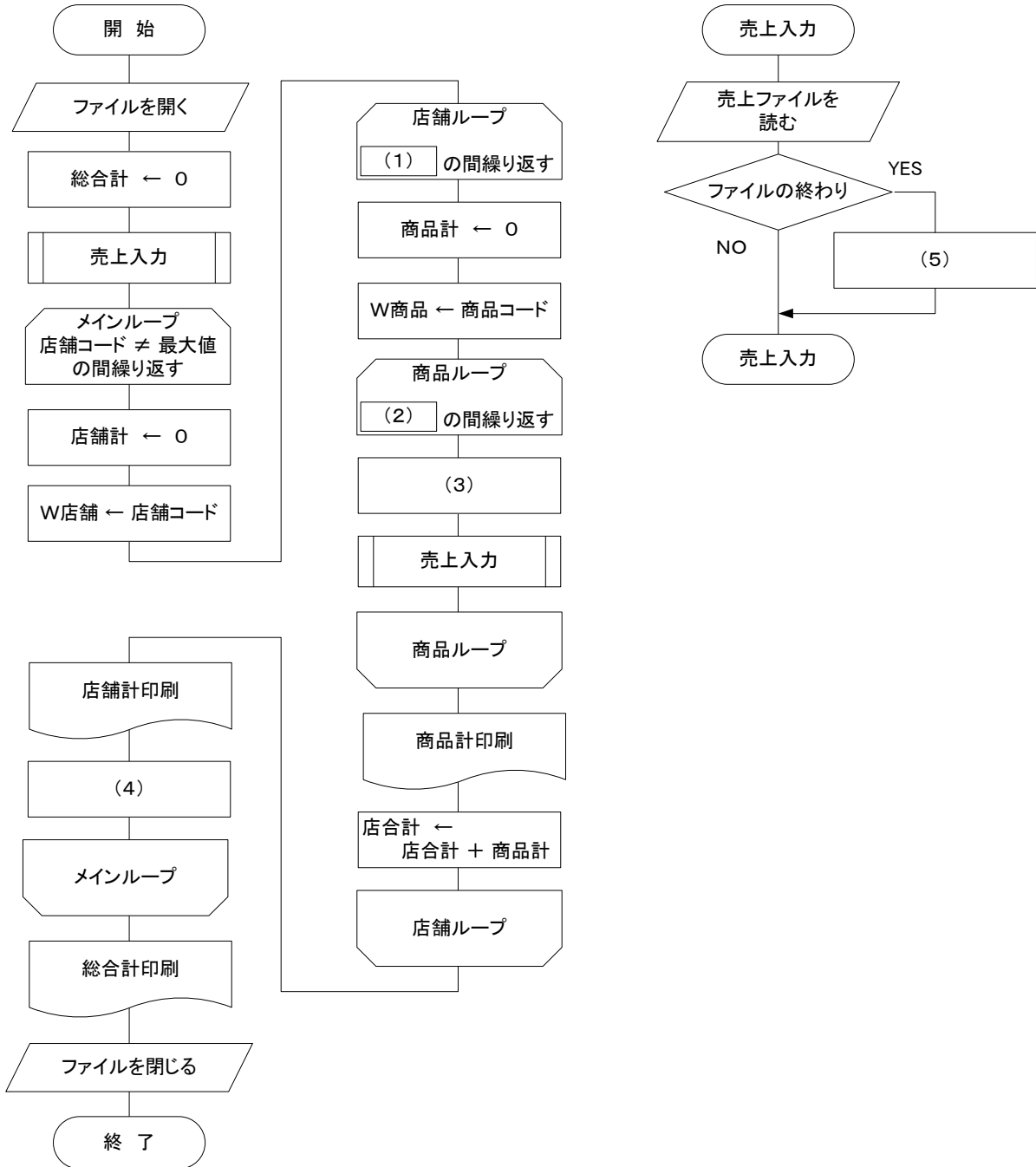


図 4 売上一覧表を作成する流れ図

(1) , (2) の解答群

- ア. W店舗=店舗コード
- イ. W店舗≠店舗コード
- ウ. W商品=商品コード
- エ. W商品≠商品コード
- オ. W店舗=店舗コード かつ W商品=商品コード
- カ. W店舗≠店舗コード かつ W商品≠商品コード
- キ. W店舗=店舗コード または W商品=商品コード
- ク. W店舗≠店舗コード または W商品≠商品コード

(3) , (4) の解答群

- ア. 総合計←総合計+売上金額
- イ. 総合計←総合計+商品計
- ウ. 総合計←総合計+店舗計
- エ. 店舗計←店舗計+売上金額
- オ. 店舗計←店舗計+商品計
- カ. 商品計←商品計+売上金額

(5) の解答群

- ア. 商品コード←最大値
- イ. 店舗コード←最大値
- ウ. キー項目←最大値
- エ. W商品←最大値
- オ. W店舗←最大値
- カ. Wキー←最大値

問題3 次の並べ替えに関する説明を読み、各設問に答えよ。

[並べ替えに関する説明]

配列の要素を並べ替えるアルゴリズムのひとつにバブルソートがある。

バブルソートは、配列の隣り合う要素を順番に比較し、昇順に並べ替えるのであれば、要素位置が小さい方に小さい値が入るように要素を入れ替える。これを何度か繰り返すことで、順番に並び替わる。

配列 `tbl` に格納されている要素を昇順に並べ替える。なお、要素数は  $N$  個とし、一番小さい要素位置（添字）は  $0$  とする。

<設問1> 次の流れ図の説明を読み、流れ図中の  に入るべき適切な字句を解答群から選べ。

[流れ図の説明]

配列 `tbl` の隣り合う要素を比較し、値が小さい方が要素位置の小さい場所に格納されるように並べ替える処理を要素位置の大きい場所から小さい場所に向かって実行する。

[実行例]  $N=5$  の場合

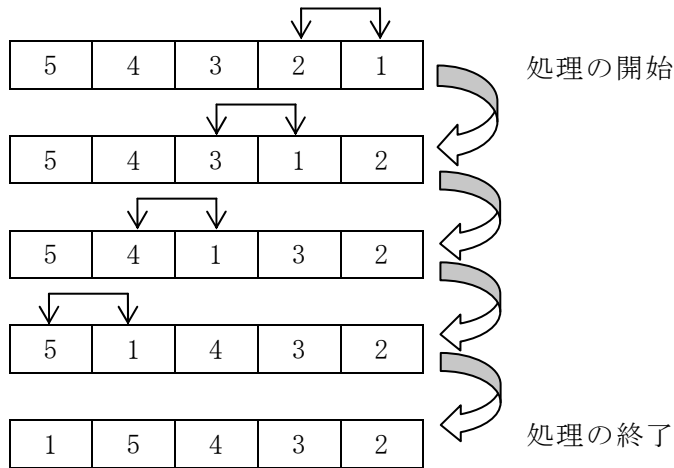


図1 実行例



[流れ図]

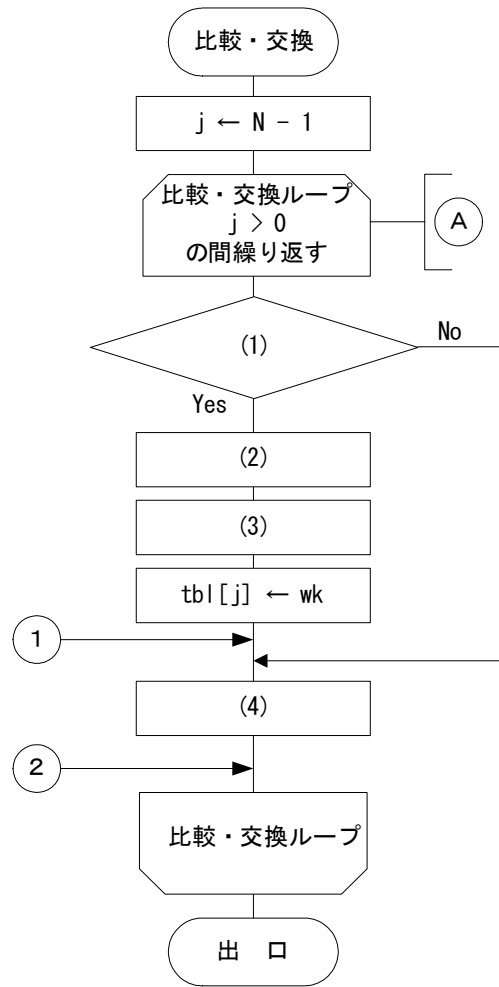


図2 比較・交換の流れ図

(1) の解答群

- ア.  $tbl[j] < tbl[j-1]$
- ウ.  $tbl[j] > tbl[j-1]$

- イ.  $tbl[j] < tbl[j+1]$
- エ.  $tbl[j] > tbl[j+1]$

(2) , (3) の解答群

- ア.  $wk \leftarrow tbl[j]$
- ウ.  $wk \leftarrow tbl[j+1]$
- オ.  $tbl[j-1] \leftarrow tbl[j]$

- イ.  $wk \leftarrow tbl[j-1]$
- エ.  $tbl[j] \leftarrow wk$
- カ.  $tbl[j+1] \leftarrow tbl[j]$

(4) の解答群

- ア.  $j \leftarrow j - 1$
- ウ.  $j \leftarrow N - 1$

- イ.  $j \leftarrow j + 1$
- エ.  $j \leftarrow N + 1$

<設問 2 > 配列 tbl の要素を昇順に並べ替えるため、図 2 の比較・交換を呼び出す流れ図を図 4 に示す。流れ図中の  に入れるべき適切な字句を (5) の解答群から選べ。ただし、図 2 の比較・交換ループの ㉔ 部分の条件式を変更する必要がある。変更後の条件式を (6) の解答群から選べ。

図 2 の比較・交換を何度か繰り返すことにより、配列の要素を昇順に並べ替えることができる。

図 2 の比較・交換を 1 回実行すると、一番小さい要素位置（配列の 0 番目）に格納される値が決まり、2 回続けて実行すると、二番目に小さい要素位置（配列の 1 番目）に格納される値が決まる。

よって、比較・交換を何度か繰り返す過程において、隣り合う要素を比較する範囲を小さくすることができる。具体的には、1 回目はすべての要素位置を対象とするが、2 回目は一番小さい要素位置を、3 回目は二番目に小さい要素位置を対象となる範囲から除くことができる（図 3 の塗りつぶし部分が比較・交換の範囲）。

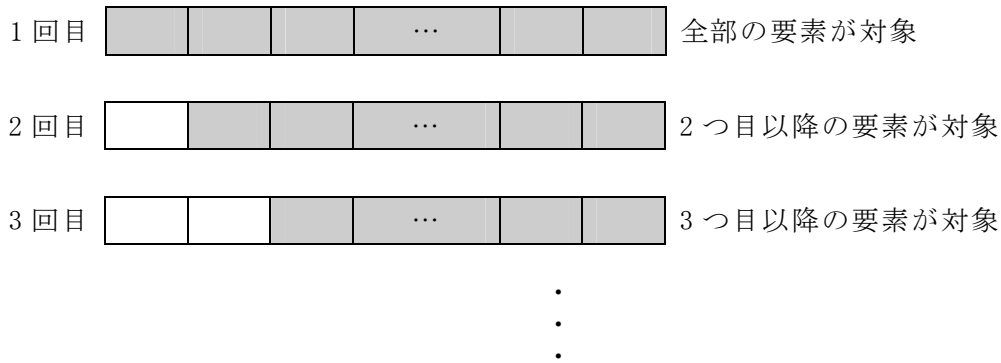


図 3 処理の範囲

[流れ図]

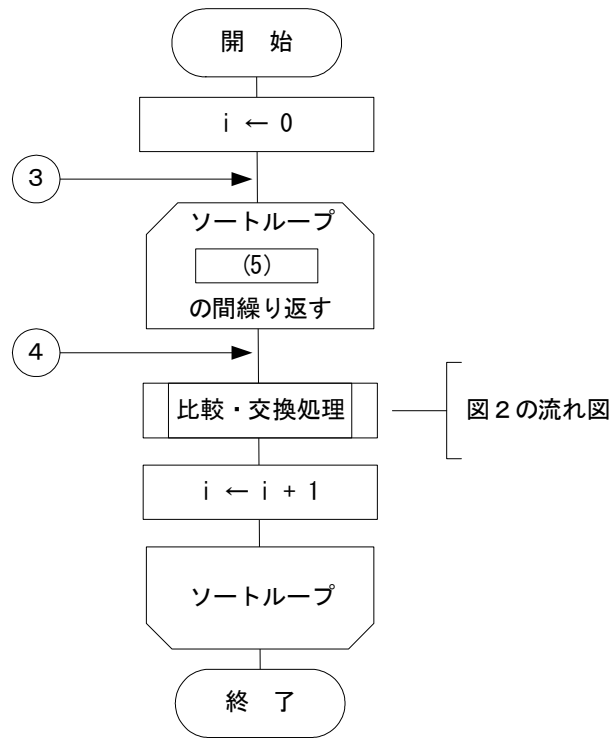


図4 並べ替えの流れ図

(5) の解答群

ア.  $i > 0$

ウ.  $i < N - 1$

イ.  $i > 1$

エ.  $i < N + 1$

(6) の解答群

ア.  $j > 1$

ウ.  $j > i - N$

イ.  $j > i$

エ.  $j > N$

<設問 3 > 次の繰り返しの制御に関する記述を読み、記述中の  に入れるべき適切な字句を解答群から選べ。

配列 `tbl` のデータが昇順に並んだことがわかれば、繰り返しを終えるようにする。昇順に並んだことを判断するため、変数 `sw` を用意する。

そこで、図 4 の流れ図の (5) の判断を  (7) に変更する。

さらに、変数 `sw` に値を代入する処理を図 2 と図 4 の流れ図中の①～④のいずれかに挿入する。0 を代入する処理を  (8), 1 を代入する処理を  (9) に挿入する。

(7) の解答群

ア.  $i > 0$  かつ  $sw = 1$

イ.  $i > 1$  かつ  $sw = 0$

ウ.  $i < N - 1$  かつ  $sw = 1$

エ.  $i < N + 1$  かつ  $sw = 0$

(8) , (9) の解答群

ア. ①

イ. ②

ウ. ③

エ. ④

オ. ①と③

カ. ①と④

キ. ②と③

ク. ②と④

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 word の中に格納された文字データを配列内で中央揃えする関数 cent である。

1次元配列 word の要素数は N 個とし、添字は 0 から始まる。よって、1次元配列 word の 0~N-1 番目に文字データが格納されている。

0	1	2	3	4	5	6	...	N-2	N-1
△	m	o	j	i	△	△	...	△	△

△は空白を表す

図1 配列 word

配列 word に与えられる文字は、1つの単語だけであり、単語の前後には空白が格納されている。

プログラムでは、配列 word を作業用配列 wk に転記し、単語の前後の空白を数えて、一旦すべて空白で埋められた配列 word の適切な位置に単語が戻される。作業用配列 wk の要素数は処理に十分な大きさであり、添字は 0 から始まる。

空白の個数が奇数の場合は、単語の後ろの空白が前より 1 個多くなる。

[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。

<設問> 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。

[プログラム]

○ cent (文字型 : word[], 整数型 : N)

○ 文字型 : wk[]

○ 整数型 : i, st, en

・ i ← 0

■ [ ] (1)  
・ wk[i] ← word[i]  
・ word[i] ← " "  
・ i ← i + 1

・ st ← 0

■ wk[st] = " "  
・ [ ] (2)

・ en ← N - 1

■ wk[en] = " "  
・ [ ] (3)

・ [ ] (4)

■ st <= en  
・ [ ] (5)  
・ i ← i + 1  
・ st ← st + 1

・ return

(1) の解答群

ア. i < N - 1

イ. i < N

ウ. i < N + 1

(2) , (3) の解答群

ア. i ← i + 1

イ. i ← i - 1

ウ. st ← st + 1

エ. st ← st - 1

オ. en ← en + 1

カ. en ← en - 1

(4) の解答群

- ア.  $i \leftarrow en - st + 1$
- イ.  $i \leftarrow en + st - 1$
- ウ.  $i \leftarrow N - en - st + 1$
- エ.  $i \leftarrow N - en + st - 1$
- オ.  $i \leftarrow ( N - en - st + 1 ) / 2$
- カ.  $i \leftarrow ( N - en + st - 1 ) / 2$

(5) の解答群

- ア.  $wk[i] \leftarrow word[st]$
- イ.  $wk[st] \leftarrow word[i]$
- ウ.  $word[i] \leftarrow wk[st]$
- エ.  $word[st] \leftarrow wk[i]$

## < 選 択 問 題 >

選択問題は問題から1つ選択し解答せよ。

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  | 15 ページ～18 ページ |
| ・ 表計算の問題   | 19 ページ～23 ページ |
| ・ アセンブラの問題 | 24 ページ～28 ページ |



次のC言語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]

0から9の数値の中から、4つの数値を当てるゲームがある。4つの数値はコンピュータが重複しないようにランダムに設定する。

この数値を当てるために、キーボードから4つの数値を入力する。コンピュータの設定した値と比べ、入力した値に対して次のような判断を行う。

判断 1: キーボードから入力した値の1つが、コンピュータが設定した値の1つと一致し、その位置も一致している場合、「1ヒット」とする。

判断 2: キーボードから入力した値の1つが、コンピュータが設定した値の1つと一致するが、位置は異なっている場合、「1ブロー」とする。

キーボードから4つの値を入力するごとに、「ヒット」の数と「ブロー」の数を表示してヒントとする。このゲームは、4つの値とその並びが完全に一致すれば終了となる。

(例) コンピュータの設定した値: 0468

キーボードから入力した値: 0346

この例では、キーボードから入力した「0」は値と位置が合っており、「4」と「6」は一致する値はあるが位置は異なる。この場合は、1ヒット2ブローとなる。

このゲームの流れは、次のようになる。

1. コンピュータで4つの数値を設定し、配列へ格納する。
2. 以下の処理を4つの数値の値と位置が一致するまで繰り返す。
  - 2.1 キーボードから値を入力する。
  - 2.2 入力した数値が4つでなければエラーメッセージを表示し、2.1へ戻る。
  - 2.3 入力した数値を1つずつ配列に格納する。
  - 2.4 入力した値に重複があればエラーメッセージを表示し、2.1へ戻る。
  - 2.5 ヒットの数を求める。
  - 2.6 ブローの数を求める。
  - 2.7 ヒットとブローの数を表示する。

この中の、2.4の処理を関数 check、2.5の処理を関数 hit、2.6の処理を関数 blowで行う。

[関数の説明]

check 関数

引 数：整数型配列

機 能：配列中の値に重複があるか調べる

戻り値：重複がなければ 0, 重複があれば 1

hit 関数

引 数：整数型配列 1, 整数型配列 2

機 能：コンピュータの設定した 4 つの数値（整数型配列 1）とキーボードから入力した 4 つの数値（整数型配列 2）の値と位置が一致する要素位置の数を数える

戻り値：一致した要素位置の数

blow 関数

引 数：整数型配列 1, 整数配列 2

機 能：コンピュータの設定した 4 つの数値（整数型配列 1）とキーボードから入力した 4 つの数値（整数型配列 2）の値は一致するが位置は異なる要素位置の数を数える

戻り値：一致した要素位置の数

[プログラム]

```
#define NUMS 4          /* 要素の数 */
```

```
int check(int *man) {
```

```
    int i, j, ret;
```

```
    ret = 0;
```

```
    for(i = 0; i < NUMS; i++) {  
        for(j = 0; j < NUMS; j++) {  
            if (  ) ret = 1;  
        }  
    }
```

```
    return ret;
```

```
}
```

```

int hit(int comp[], int man[]) {
    int i, ret;

    ret = 0;
    for(i = 0; i < NUMS; i++) {
        if (  ) ret++;
    }
    return ret;
}

```

```

int blow(int comp[], int man[]) {
    int i, j, ret;

    ret = 0;
    for(i = 0; i < NUMS; i++) {
        for (j = 0; j < NUMS; j++) {
            if (  ) ret++;
        }
    }
    return ret;
}

```

<設問 1> プログラム中のに入れるべき適切な式を解答群から選べ。

(1) の解答群

- |  |  |
|--|--|
| ア. <code>man[i] &gt; man[j]</code>                 | イ. <code>man[i] &lt; man[j]</code>                 |
| ウ. <code>i == j &amp;&amp; man[i] == man[j]</code> | エ. <code>i != j &amp;&amp; man[i] == man[j]</code> |

(2) の解答群

- |                                     |                                     |
|-------------------------------------|-------------------------------------|
| ア. <code>comp[i] == man[i]</code>   | イ. <code>comp[i] != man[i]</code>   |
| ウ. <code>comp[i] &lt; man[i]</code> | エ. <code>comp[i] &gt; man[i]</code> |

(3) の解答群

- |   |   |
|---|---|
| ア. <code>comp[i] &gt; man[j]</code>                 | イ. <code>comp[i] &lt; man[j]</code>                 |
| ウ. <code>i == j &amp;&amp; comp[i] == man[j]</code> | エ. <code>i != j &amp;&amp; comp[i] == man[j]</code> |

<設問 2> 関数 check は、点線で囲まれた繰り返し処理に無駄な部分がある。そこで、繰り返し処理を以下のように書き換える。書き換えるプログラム中の [ ] に入れるべき適切な式を解答群から選べ。

```
for(i = 0; i < [ (4) ]; i++) {  
    for (j = i + 1; j < [ (5) ]; j++) {  
        if ( [ (1) ] ) ret = 1;  
    }  
}
```

(4) , (5) の解答群

- |             |                 |
|-------------|-----------------|
| ア. NUMS - i | イ. NUMS - 1     |
| ウ. NUMS     | エ. NUMS + 1     |
| オ. NUMS + i | カ. NUMS + i - 1 |

<設問 3> 次のキーボードから入力した値をチェックするための手順に関する記述中の [ ] に入れるべき適切な字句を解答群から選べ。

プログラムの説明 2.2 で「入力した数値が 4 つでなければエラーメッセージを表示し、2.1 へ戻る」の判断は、入力した 4 つの値を 4 桁の整数値と考え、ある範囲内に収まるかを調べることで対処できる。

この場合、重複した値があればエラーになることも考慮し、[ (6) ] より小さい、または [ (7) ] より大きい値であれば、エラーメッセージを表示する。

(6) , (7) の解答群

- |         |         |
|---------|---------|
| ア. 123  | イ. 1000 |
| ウ. 1111 | エ. 1234 |
| オ. 9876 | カ. 9999 |

<設問 4> このゲームの繰り返しを終了するための条件「4 つの数値の値と位置が一致する」となる場合、ヒットの数とブローの数の関係を解答群から選べ。

(8) の解答群

- ア. ヒットの数 が 0 でブローの数 が 0
- イ. ヒットの数 が 0 でブローの数 が 4
- ウ. ヒットの数 が 4 でブローの数 が 0
- エ. ヒットの数 が 4 でブローの数 が 4

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。

#### VLOOKUP 関数

検索値と検索範囲のセルの値（範囲値）とを比較し、検索の型を 0 とした場合は「範囲値＝検索値」となる行、検索の型を 1 とした場合、検索値以下で最も大きい値となる行に対し、指定された列の値を返す。なお、検索の型を 1 とした場合、検索範囲の左端の列は昇順に並んでいる必要がある。

書式：VLOOKUP（検索値， 検索範囲， 列番号， 検索の型）

#### HLOOKUP 関数

検索値と検索範囲のセルの値（範囲値）とを比較し、検索の型を 0 とした場合は「範囲値＝検索値」となる列、検索の型を 1 とした場合、検索値以下で最も大きい値となる列に対し、指定された行の値を返す。なお、検索の型を 1 とした場合、検索範囲の上端の行は昇順に並んでいる必要がある。

書式：HLOOKUP（検索値， 検索範囲， 行番号， 検索の型）

#### 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。

書式：COUNTIF（範囲， 検索条件）

#### SUMIF 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に対応する合計範囲の合計を求める。なお、合計範囲は省略可能であり、省略された場合は範囲の合計を求める。

書式：SUMIF（範囲， 検索条件， [合計範囲]）

#### INT 関数

数値の整数部を返す。

書式：INT（数値）

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

J社では4つの支店の売上を、四半期ごとに表計算ソフトを利用して集計している。利用する表は、次のようにまとめている。

表1 「売上」シート（四半期分の売上データ）

	A	B	C	D	E	F	G
1							単位:千円
2		月	支店コード	支店名	売上額	1人当たりの売上額	評価
3		10	S01	東京支店	246,700	4,934	4
4		10	S02	神奈川支店	41,000	1,464	2
5		10	S03	千葉支店	68,100	2,724	3
6		10	S04	埼玉支店	15,900	795	1
7		11	S01	東京支店	126,300	2,526	3
8		11	S02	神奈川支店	99,300	3,546	4
9		11	S03	千葉支店	50,100	2,004	3
10		11	S04	埼玉支店	21,600	1,080	2
11		12	S01	東京支店	294,000	5,880	5
12		12	S02	神奈川支店	107,200	3,829	4
13		12	S03	千葉支店	75,300	3,012	4
14		12	S04	埼玉支店	44,700	2,235	3

B列（月）、C列（支店コード）、E列（売上額）を入力する。他の列は表2を参照して、関数を用いて求めている。

表2 「集計」シート

	A	B	C	D	E	F	G
1		評価基準					単位:千円
2		評価下限	0	1,000	2,000	3,000	5,000
3		評価値	1	2	3	4	5
4							
5		支店集計表				単位:千円	
6		支店コード	支店名	従業員数	四半期合計	1人月平均	
7		S01	東京支店	50	667,000	4,446	
8		S02	神奈川支店	28	247,500	2,946	
9		S03	千葉支店	25	193,500	2,580	
10		S04	埼玉支店	20	82,200	1,370	
11							
12		月別評価値集計表					
13		評価値	件数				
14		5	1				
15		4	4				
16		3	4				
17		2	2				
18		1	1				

支店集計表の四半期合計，1人月平均および月別評価値集計表の件数については，関数を用いて求めている。

<設問> 表1, 表2で用いている関数について答えよ。

(1) 支店名を表示するため、表1のセルD3に入力する式を解答群から選べ。ただし、セルD3の式は、セルD4~D14に複写する。

(2) 従業員1人当たりの売上金額を求めるため、表1のセルF3に入力する次の式の中の  に入力する式を解答群から選べ。ただし、セルF3の式は、セルF4~F14に複写する。

$$= E3 / \text{  (2) }$$

(1), (2) の解答群

- ア. VLOOKUP( C3, 集計!\$B\$7:\$D\$10, 2, 0 )
- イ. VLOOKUP( C3, 集計!\$B\$7:\$D\$10, 2, 1 )
- ウ. VLOOKUP( \$C\$3, 集計!\$B\$7:\$D\$10, 2, 0 )
- エ. VLOOKUP( \$C\$3, 集計!\$B\$7:\$D\$10, 2, 1 )
- オ. VLOOKUP( C3, 集計!\$B\$7:\$D\$10, 3, 0 )
- カ. VLOOKUP( C3, 集計!\$B\$7:\$D\$10, 3, 1 )
- キ. VLOOKUP( \$C\$3, 集計!\$B\$7:\$D\$10, 3, 0 )
- ク. VLOOKUP( \$C\$3, 集計!\$B\$7:\$D\$10, 3, 1 )

(3) 各月各支店の従業員1人当たりの売上金額を評価するため、表1のG3に入力する式を解答群から選べ。ただし、G3の式は、G4~G14に複写する。

なお、評価は、1人当たりの売上額を表2の評価基準に準じて決定する。

(3) の解答群

- ア. HLOOKUP( F3, 集計!\$C\$2:\$G\$3, 2, 0 )
- イ. HLOOKUP( F3, 集計!\$C\$2:\$G\$3, 2, 1 )
- ウ. HLOOKUP( \$F\$3, 集計!\$C\$2:\$G\$3, 2, 0 )
- エ. HLOOKUP( \$F\$3, 集計!\$C\$2:\$G\$3, 2, 1 )

(4) 支店ごとの四半期合計を求めるため、表2のセルE7に入力する式を解答群から選べ。ただし、セルE7の式は、セルE8~E10に複写する。

(4) の解答群

- ア. SUMIF( 売上!E3:E14, C3 )
- イ. SUMIF( 売上!\$E\$3:\$E\$14, C3 )
- ウ. SUMIF( 売上!C3:C14, B7, 売上!E3:E14 )
- エ. SUMIF( 売上!\$C\$3:\$C\$14, B7, 売上!\$E\$3:\$E\$14 )



(5) 支店ごとに従業員 1 人当たりの四半期平均売上を求めるため、表 2 のセル F7 に入力する式を解答群から選べ。ただし、セル F7 の式は、セル F8～F10 に複写する。

**(5) の解答群**

- ア.  $\text{INT}(E7 / 3)$
- イ.  $\text{INT}(E7 / D7)$
- ウ.  $\text{INT}(E7 / D7 / 3)$
- エ.  $\text{INT}(E7 / D7 / 3 / 3)$

(6) 表 1 の売上表で従業員 1 人当たりの売上に対する評価値の件数を求めるため、表 2 の月別評価値集計表のセル C14 に入力する式を解答群から選べ。ただし、セル C14 の式は、セル C15～C18 に複写する。

**(6) の解答群**

- ア.  $\text{COUNTIF}(\text{売上!G3:G14}, B14)$
- イ.  $\text{COUNTIF}(\text{売上!G3:G14}, \$B\$14)$
- ウ.  $\text{COUNTIF}(\text{売上!$G$3:$G$14}, B14)$
- エ.  $\text{COUNTIF}(\text{売上!$G$3:$G$14}, \$B\$14)$

## 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]

2次元のビットデータ（7語×15ビット）に対し、水平・垂直のパリティチェックを行いビットの誤り位置を求める副プログラム PRC である。

メインプログラムにはチェックする2次元データ（7語×15ビット）が水平・垂直パリティチェックビットとともに図1のように設定されている。

	水平パリティチェックビット																
ビット	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
1 語目	0	0	0	1	0	1	0	1	0	0	1	1	0	0	0	1	垂直 パ リ テ ィ チ ェ ッ ク ビ ッ ト
2 語目	1	1	0	0	0	0	1	0	0	0	1	0	0	1	1	0	
3 語目	0	0	0	1	1	1	0	1	0	0	0	0	1	0	1	0	
4 語目	0	1	1	0	0	1	0	1	0	0	1	1	0	1	1	0	
5 語目	1	1	0	0	0	1	0	1	0	1	1	0	0	0	1	1	
6 語目	0	0	0	1	0	0	1	0	1	0	0	0	1	0	1	1	
7 語目	1	1	0	0	0	1	0	1	1	0	0	0	0	0	0	1	
	1	0	1	1	1	1	0	1	0	1	0	0	0	0	1	0	

図1 2次元データとパリティチェックビット

PRC は、GR1 に2次元データの先頭語のアドレスを設定して呼び出される。PRC で各パリティビットを含めて誤りチェックを行い、誤りがあった場合はその位置情報を表のように設定してメインプログラムに返す。

表 GR5 と GR6 の戻り情報

GR5	水平パリティチェックを行い、誤りがあった語の位置を上位7ビットに設定する
GR6	垂直パリティチェックを行い、誤りがあったビットの位置を16ビットに設定する

PRC はラベル HP, VP 番地に水平、垂直の検査結果を設定し、検査終了後に GR5 及び GR6 に設定してメインプログラムに戻る。例えば、3語目の第6ビットと6語目の第12ビットに誤りがあった場合は図2のようになる。

ビット	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	HP
1 語目	0	0	0	1	0	1	0	1	0	0	1	1	0	0	0	1	0
2 語目	1	1	0	0	0	0	1	0	0	0	1	0	0	1	1	0	0
3 語目	0	0	0	1	1	1	0	1	0	1	0	0	1	0	1	0	1
4 語目	0	1	1	0	0	1	0	1	0	0	1	1	0	1	1	0	0
5 語目	1	1	0	0	0	1	0	1	0	1	1	0	0	0	1	1	0
6 語目	0	0	0	0	0	0	1	0	1	0	0	0	1	0	1	1	1
7 語目	1	1	0	0	0	1	0	1	1	0	0	0	0	0	0	1	0
	1	0	1	1	1	1	0	1	0	1	0	0	0	0	1	0	

VP	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0
----	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

図2 2ビットの誤りがある場合

この場合、水平検査を行うと3語目と6語目に誤りがあるのでHPの上から3番目と6番目が1となり、ラベルHPの内容は16進数で(2400)に設定される。

また、垂直検査を行うと第6ビットと12ビットに誤りがあるので、ラベルVPの内容は16進数で(1040)と設定される。

<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。

[プログラム]

行番号	ラベル	命令	オペランド	コメント
100	PRC	START		
110		ST	GR1, DATA	
120		LAD	GR0, 0	
130		ST	GR0, HP	; 水平パリティの初期化
140		ST	GR0, VP	; 垂直パリティの初期化
150		LD	GR0, GR1	
160		ADDL	GR0, =8	
170		ST	GR0, ENDAD	; 終了判定条件の設定
180		LD	GR2, =#8000	; 水平パリティチェックのマスクビット
190	LOOP1	LD	GR3, =#8000	; 垂直パリティチェックのマスクビット
200		LD	GR0, 0, GR1	
210	LOOP2	SLL	GR0, 1	
220			<input type="text" value="(1)"/>	
230		JUMP	NSET	
240	SET	LD	GR4, GR2	
250			<input type="text" value="(2)"/>	; 水平パリティの検査ビット設定
260		ST	GR4, HP	
270		LD	GR4, GR3	
280			<input type="text" value="(3)"/>	; 垂直パティの検査ビット設定
290		ST	GR4, VP	
300	NSET	LD	GR0, GR0	; GR0 の内容が 0 なら次の語へ
310		JZE	NEXT	
320		SRL	GR3, 1	; マスクビットの変更
330		JUMP	LOOP2	
340	NEXT		<input type="text" value="(4)"/>	; マスクビットの変更
350		LAD	GR1, 1, GR1	
360		CPL	GR1, ENDAD	
370		JMI	LOOP1	
380		LD	GR5, HP	
390		LD	GR6, VP	
400		RET		
410	HP	DS	1	
420	VP	DS	1	
430	ENDAD	DS	1	
440	DATA	DS	1	
450		END		

(1) の解答群

- ア. JNZ SET                   イ. JNZ NEXT  
ウ. JOV SET                   エ. JOV NEXT

(2) , (3) の解答群

- ア. AND GR2,HP               イ. AND GR3,VP               ウ. AND GR4,HP  
エ. XOR GR3,VP               オ. XOR GR4,HP               カ. XOR GR4,VP

(4) の解答群

- ア. SLL GR2,1               イ. SLL GR3,1  
ウ. SRL GR2,1               エ. SRL GR3,1

<設問2> 1ビット以下の誤りを修正するプログラムを副プログラム PRC の 390 行目と 400 行目の間に追加した。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]

行番号	ラベル	命令	オペランド	コメント
		LD	GR5,GR5	
		JZE	EXIT	
		LD	GR1,DATA	; 2次元データの先頭番地
		LAD	GR2,0	
		LAD	GR3,0	
	L1	SLL	GR5,1	
		<input type="text"/>	(5)	
		LAD	GR2,1,GR2	; 語数のカウント
		JUMP	L1	
	L2	SLL	GR6,1	
		<input type="text"/>	(6)	
		LAD	GR3,1,GR3	; ビット位置のカウント
		JUMP	L2	
	L3	ADDL	GR1,GR2	
		LD	GR4,=#8000	; マスクビットの初期設定
		<input type="text"/>	(7)	; マスクビットの生成
		XOR	GR4,0,GR1	
		ST	GR4,0,GR1	
	EXIT	LD	GR5,HP	
		LD	GR6,VP	

(5) , (6) の解答群

ア. JOV L2

ウ. JZE L2

イ. JOV L3

エ. JZE L3

(7) の解答群

ア. SLL GR4,1

ウ. SRL GR4,1

イ. SLL GR4,0,GR3

エ. SRL GR4,0,GR3

<メモ欄>

