

平成24年度前期 情報検定

<実施 平成24年9月9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33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. 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～16 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	18 ページ～21 ページ
・ 表計算の問題	22 ページ～26 ページ
・ アセンブラの問題	28 ページ～33 ページ

必須問題

問題 1 次の 2 分木に関する設問に答えよ。

2 分木とは、木構造の中で各ノードが持つ枝の数が 2 以下のものをいう。また、どの親子関係においても、次の大小関係が成立する。

$$(\text{左の子の値}) \leq (\text{親の値}) \leq (\text{右の子の値})$$

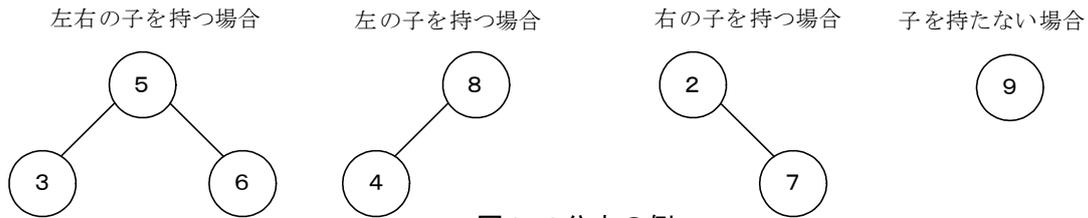


図1 2分木の例

2 分木に対する操作を、次に示す。

[追加]

本問では、追加データとノードの値を比較し、次のように大小関係を満たす方向の枝へたどっていく。

(追加データ) \leq (ノードの値) なら左の枝へ

(追加データ) $>$ (ノードの値) なら右の枝へ

ルートから比較を開始し、たどる枝が無くなったときに、追加データをその枝方向の子として追加する。

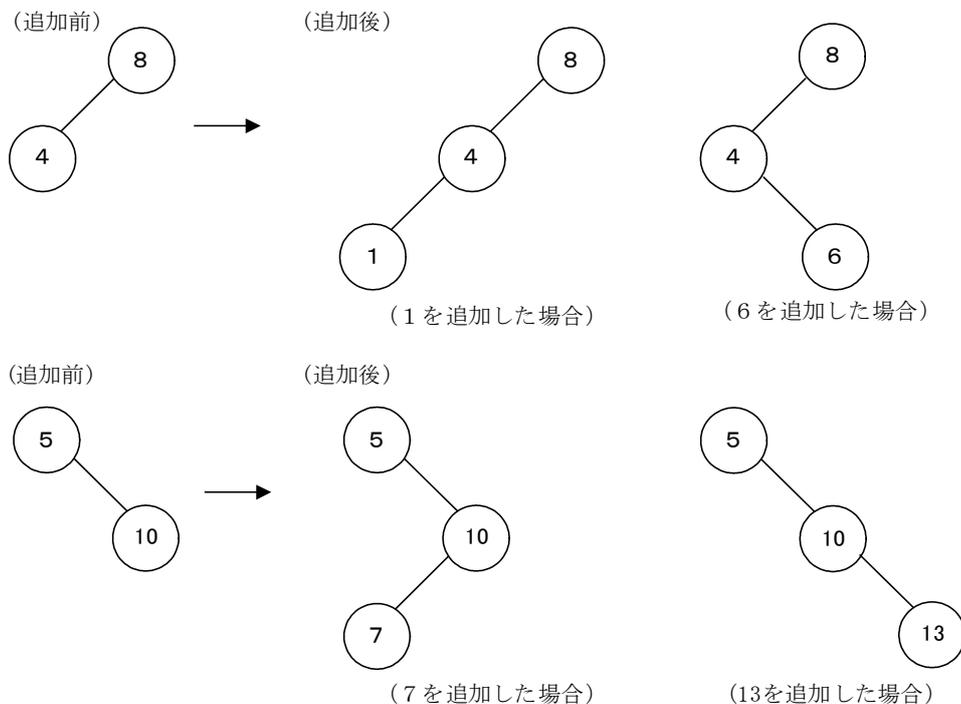
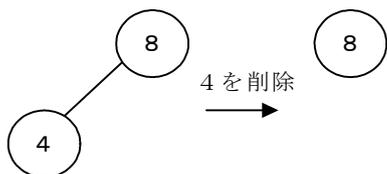


図2 ノードの追加

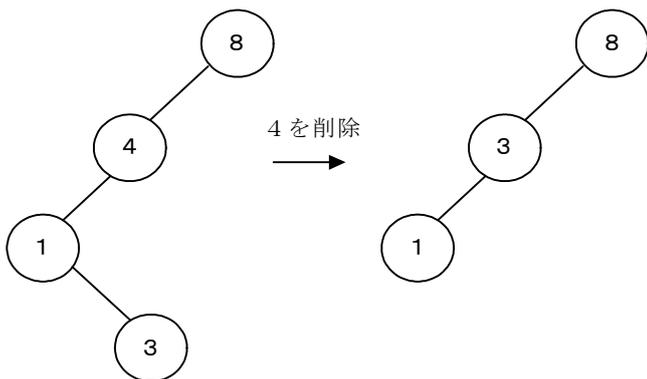
[削除]

①削除するノードが子を持たない場合



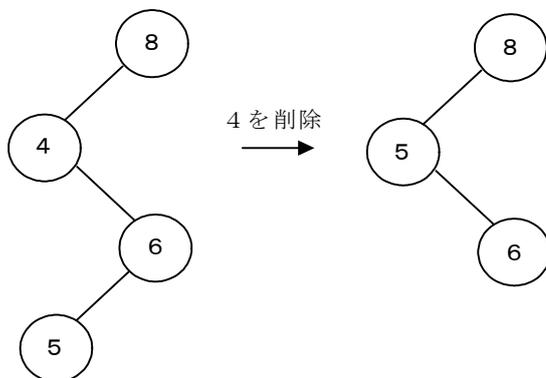
削除するノードの親から、削除するノードへの枝を削除する。

②削除するノードが左の部分木だけ持つ場合



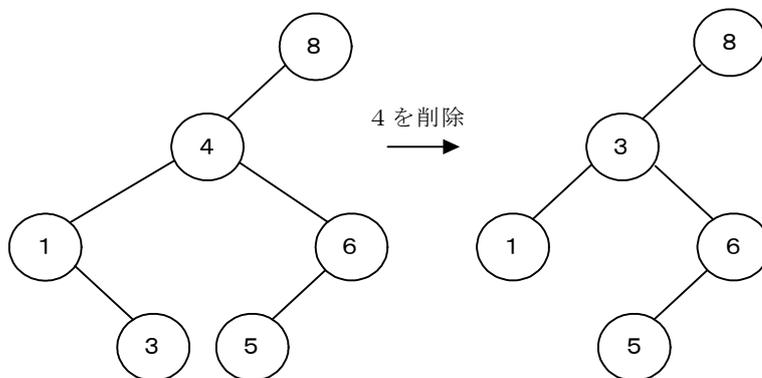
削除するノードが持つ左部分木中の最大値を、削除ノードへ移動させる。

③削除するノードが右の部分木だけ持つ場合



削除するノードが持つ右部分木中の最小値を、削除ノードへ移動させる。

④削除するノードが左右両方の部分木を持つ場合



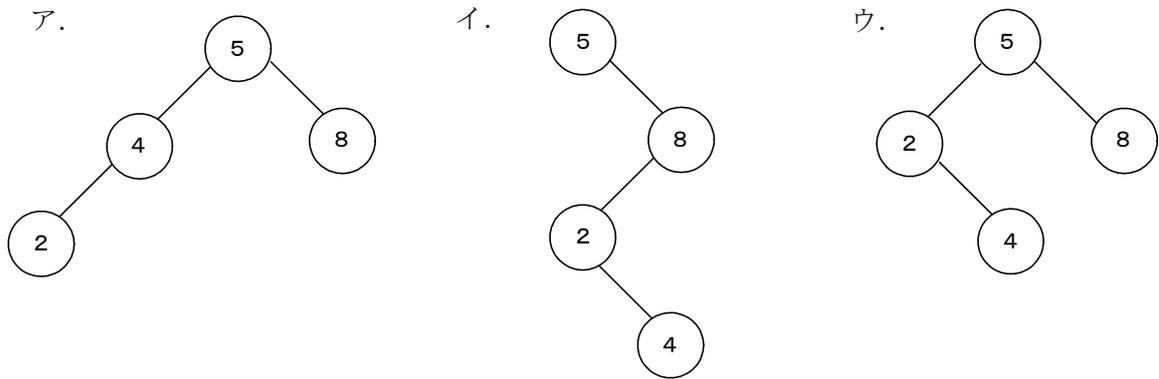
②, ③の両方の操作が可能であり、一般的には部分木の深さなどを考慮して選択される。本問では、常に②の操作を選択する。

図3 ノードの削除

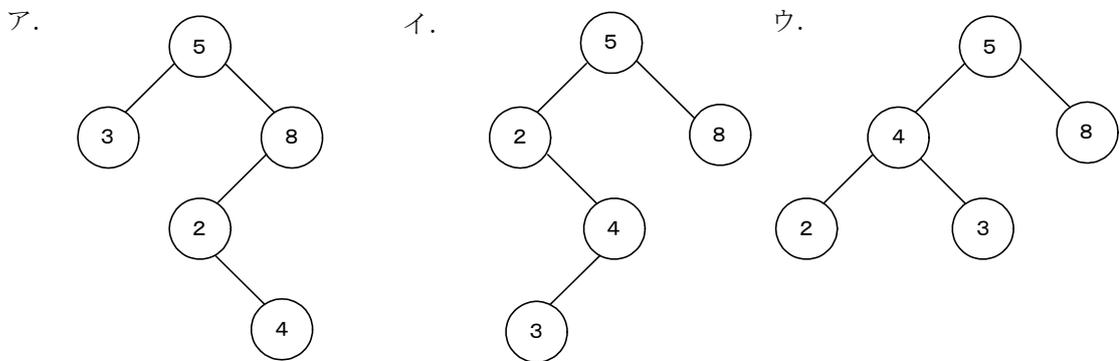
<設問> 次の (1)～(5) の順で、2分木に対する操作を行った。それぞれの操作が行われた時点での2分木の構造を解答群から選べ。

- (1) 最初のデータ5をルートとして、8、2、4の順に2分木に追加した。
- (2) 3を追加した。
- (3) 2を削除した。
- (4) 10を追加した。
- (5) 8を削除した。

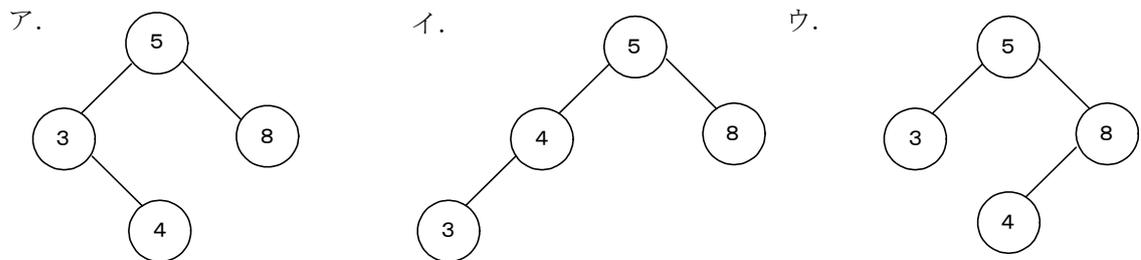
(1) の解答群



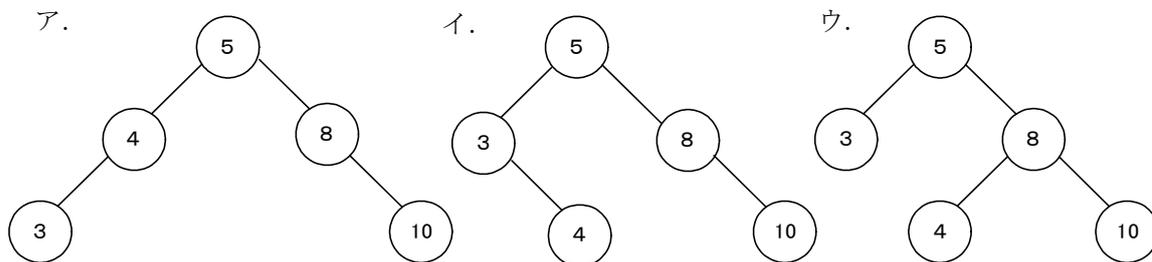
(2) の解答群



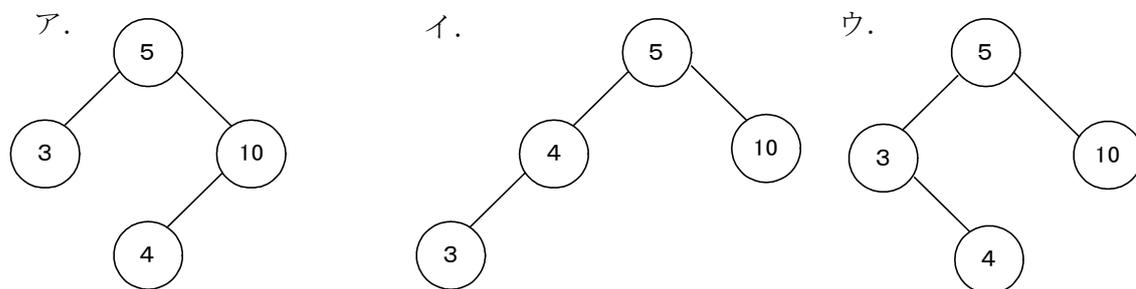
(3) の解答群



(4) の解答群



(5) の解答群



問題2 次のファイル更新に関する各設問に答えよ。

[ファイル更新の説明]

Jバイクショップで販売する商品の在庫量を更新する処理である。

Jバイクショップは 10 店舗を持っており，オートバイや自転車，関連用品を販売している。各店舗で在庫は持たず，店舗から毎日本社に送られる発注データをもとに本社の倉庫から店舗に発送する。

在庫マスタのレコード形式は次のとおりであり，商品コードの昇順に並んでいる順編成ファイルである。

(在庫マスタ)

商品コード	商品名	在庫数
-------	-----	-----

発注データのレコード形式は次のとおりであり，順編成ファイルである。各店舗で発生する発注データは発注日の昇順に並んでいる。

(発注データ)

店舗コード	商品コード	発注日	発注数
-------	-------	-----	-----

在庫を更新するための流れ図は，下図のようになる。

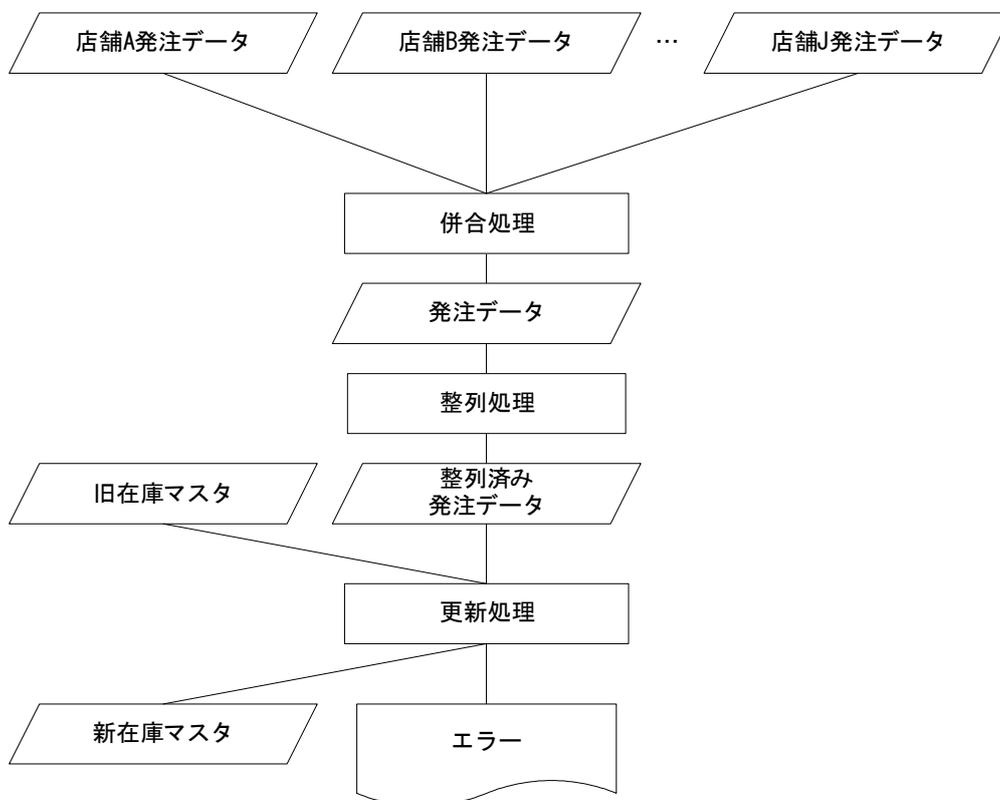


図1 在庫を更新する流れ

各処理の内容は、次のとおりである。

表 処理内容

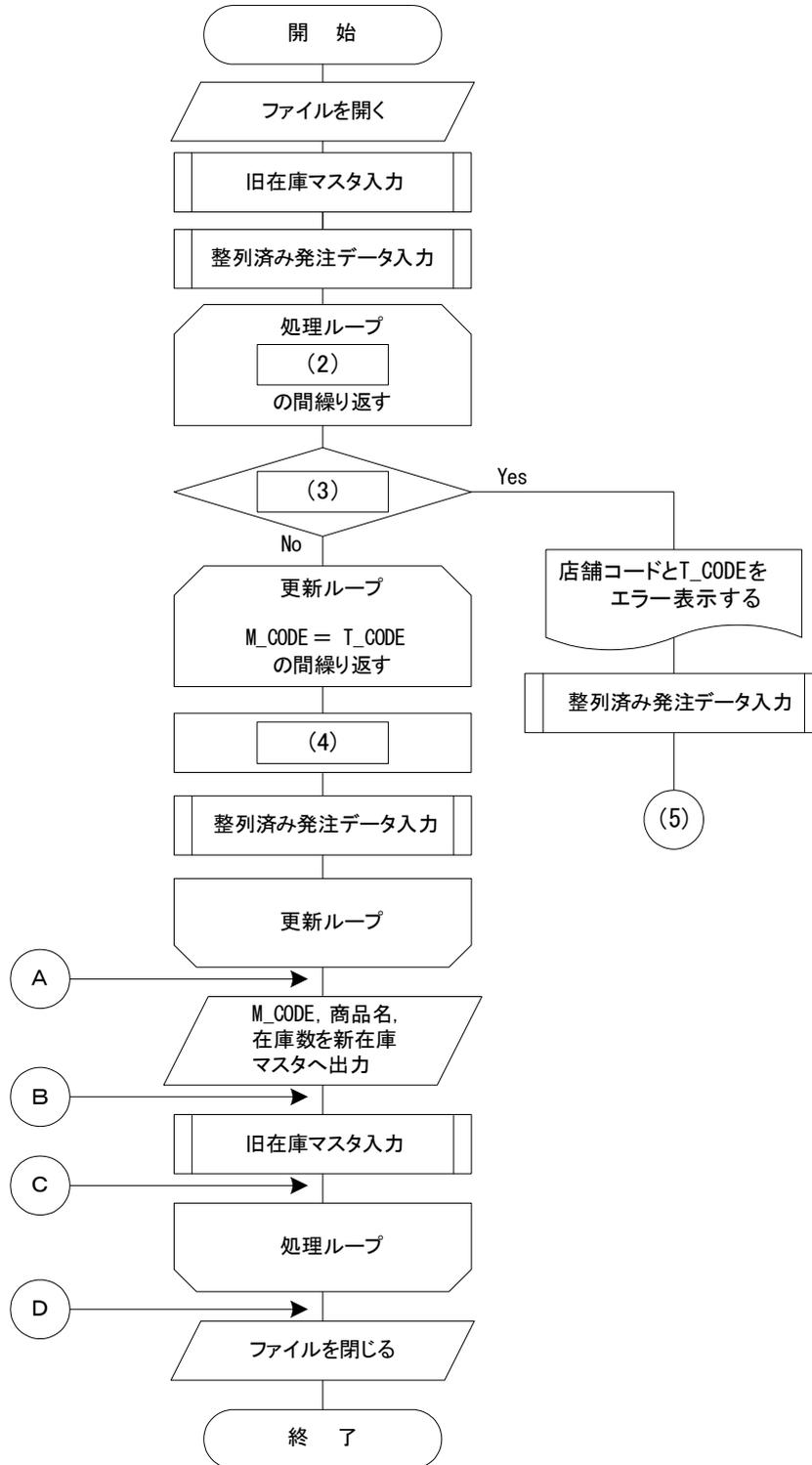
処 理	処 理 内 容
併合処理	各店舗のデータを1つのファイルにまとめ、発注データを作成する。発注データのレコード形式は各店舗から送られる発注データと同じである。
整列処理	発注データを並べ替えて整列済み発注データを作成する。発注データと整列済み発注データのレコード形式は同じである。
更新処理	旧在庫マスタと整列済み発注データを突き合わせ、商品コードが同じレコードの在庫量から発注量を引いて新しい在庫量とし、新在庫マスタを作成する。 ただし、旧在庫マスタの商品コードが整列済み発注データに存在しなければ、そのまま新在庫マスタへ出力し、整列済み発注データの商品コードが旧在庫マスタに存在しなければエラーとする。

<設問 1 > 整列処理における並べ替えのキーとして最低限必要な項目を解答群から選べ。なお、複数の項目がある場合は、左から順に並べ替えのキーの優先順位を示す。

(1) の解答群

- ア. 商品コード
- イ. 店舗コード, 商品コード
- ウ. 店舗コード, 発注日, 商品コード
- エ. 発注日, 商品コード

<設問 2 > 次の更新処理の流れ図中の (2)～(4) に入れるべき適切な字句を解答群から選べ。なお、流れ図中で用いている最高値とは、システムで表現できる最大の値であり、どの入力データと比較しても大きいと判定される値である。



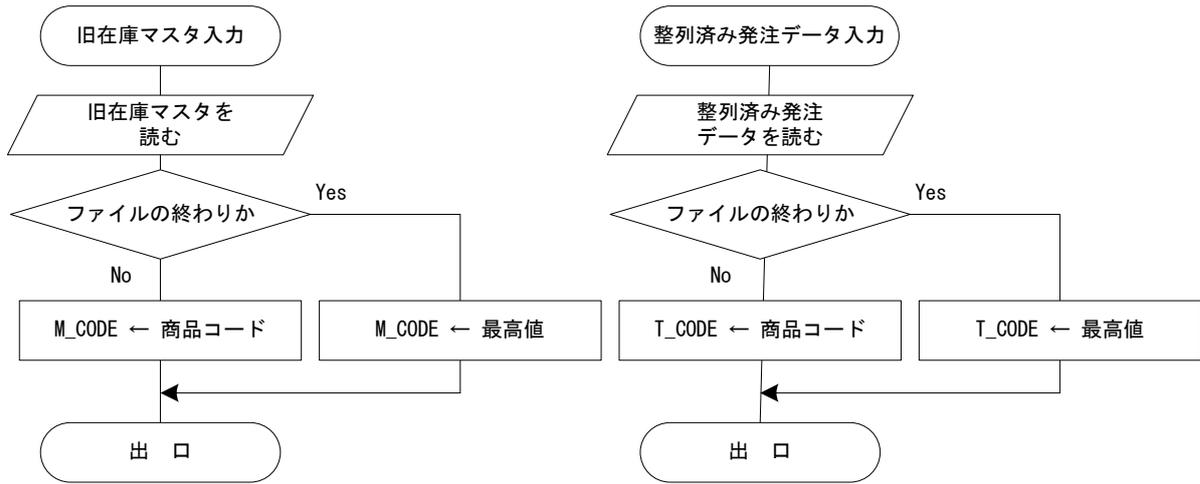


図2 更新処理の流れ図

(2) の解答群

- ア. $M_CODE \neq T_CODE$
- イ. $M_CODE \neq \text{最高値}$
- ウ. $T_CODE \neq \text{最高値}$
- エ. $M_CODE \neq \text{最高値}$ または $T_CODE \neq \text{最高値}$

(3) の解答群

- ア. $M_CODE = T_CODE$
- イ. $M_CODE \neq T_CODE$
- ウ. $M_CODE > T_CODE$
- エ. $M_CODE < T_CODE$

(4) の解答群

- ア. 在庫数 \leftarrow 在庫数 + 発注数
- イ. 在庫数 \leftarrow 在庫数 - 発注数
- ウ. 発注数 \leftarrow 在庫数 + 発注数
- エ. 発注数 \leftarrow 在庫数 - 発注数

<設問3> 更新処理の流れ図中の(5)の合流先として適切な場所を解答群から選べ。

(5) の解答群

- ア. A
- イ. B
- ウ. C
- エ. D

問題3 次の文字列処理に関する説明を読み、各設問に答えよ。

[文字列処理の説明]

配列 X 中に長さ r の文字列が格納してある。この文字列の中から配列 F に格納されている文字列を探索し、配列 E に格納されている文字列で置換する流れ図である。置換後の文字列は配列 Y に格納するが、配列 X, Y, F, E は処理に十分な大きさを持っているものとする。また、配列 F に格納されている文字列の長さは m 、配列 E に格納されている文字列の長さは n で与えられているものとする。なお、配列 F に格納されている文字列と同じ文字列が見つからないときは、そのまま配列 X を配列 Y に転送する。

配列 X

1	2	3	4	5	6	7	8	9	10	11	12	13	...	r	...
P	R	O	G	R	A	M	C	O	B	O	L	G	...	N	...

配列 F

1	2	3	4	5	...
C	O	B	O	L	...

 $m=5$

配列 E

1	2	3	4	...
J	A	V	A	...

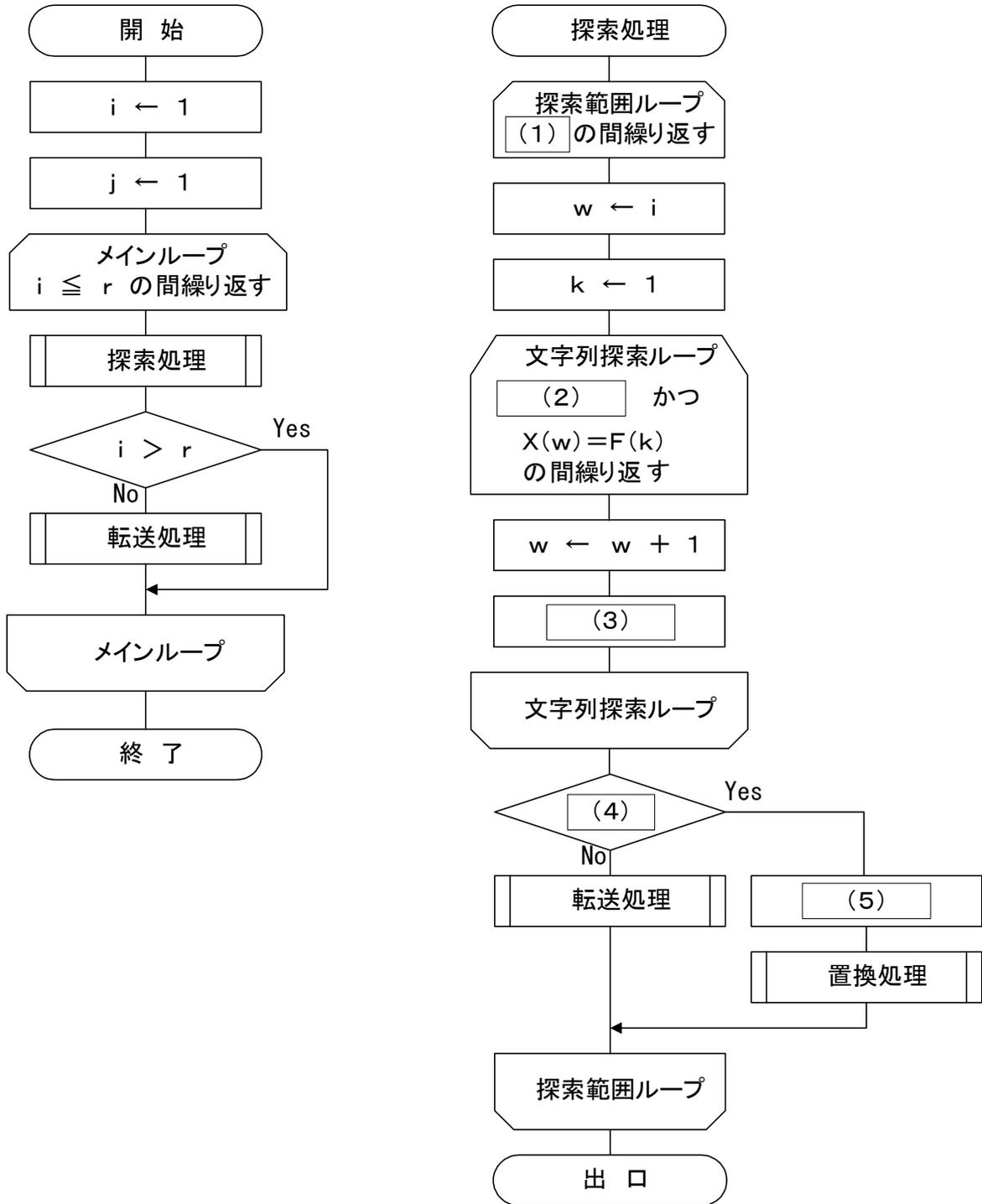
 $n=4$

配列 Y

1	2	3	4	5	6	7	8	9	10	11	12	...		
P	R	O	G	R	A	M	J	A	V	A	G	...	N	...

<設問 1> 次の流れ図中の に入れるべき適切な字句を解答群から選べ。

[流れ図]



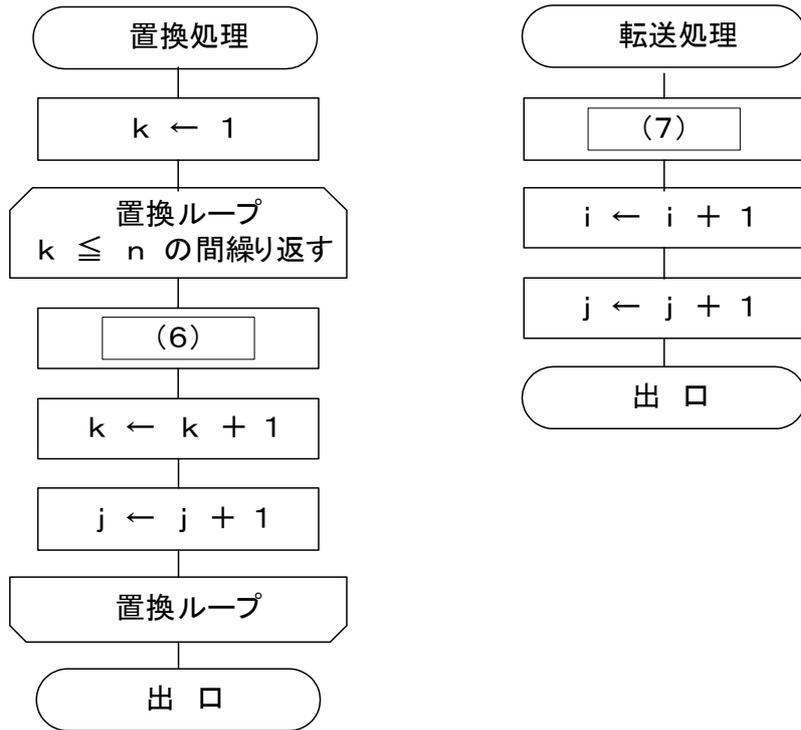


図 流れ図

(1) の解答群

- | | | |
|---------------|-------------------|-----------------------|
| ア. $i \leq n$ | イ. $i \leq n - m$ | ウ. $i \leq n - m + 1$ |
| エ. $i \leq r$ | オ. $i \leq r - m$ | カ. $i \leq r - m + 1$ |

(2) , (4) の解答群

- | | | |
|---------------|---------------|---------------|
| ア. $k > n$ | イ. $k \geq n$ | ウ. $k \leq n$ |
| エ. $k < n$ | オ. $k > m$ | カ. $k \geq m$ |
| キ. $k \leq m$ | ク. $k < m$ | |

(3) , (5) の解答群

- | | | |
|-------------------------|-------------------------|-------------------------|
| ア. $i \leftarrow i + 1$ | イ. $i \leftarrow i - 1$ | ウ. $i \leftarrow k$ |
| エ. $i \leftarrow w$ | オ. $k \leftarrow k + 1$ | カ. $k \leftarrow k - 1$ |
| キ. $j \leftarrow j + 1$ | ク. $j \leftarrow j - 1$ | ケ. $j \leftarrow k$ |
| コ. $j \leftarrow w$ | | |

(6) , (7) の解答群

- | | | |
|---------------------------|---------------------------|---------------------------|
| ア. $Y(j) \leftarrow X(i)$ | イ. $Y(j) \leftarrow X(k)$ | ウ. $Y(j) \leftarrow X(w)$ |
| エ. $Y(j) \leftarrow E(i)$ | オ. $Y(j) \leftarrow E(k)$ | カ. $Y(j) \leftarrow E(w)$ |

<設問 2 > 配列 X 中に配列 F と一致する文字列が複数存在する場合，この流れ図ではどのような置換を行うか，解答群から選べ。

(8) の解答群

- ア．最初に一致する文字列のみ，置換する。
- イ．最後に一致する文字列のみ，置換する。
- ウ．最初と最後に一致する文字列のみ，置換する。
- エ．一致する文字列全てを置換する。

問題4 次のプログラムの説明を読み、設問に答えよ。

[プログラムの説明]

エラトステネスのふるいにより、2~Nまでの素数を表示するプログラム SOSU である。素数とは、1と自分以外の約数を持たない自然数のことで、1は含まない。

エラトステネスのふるいにより素数を求める方法は、次のようになる。

- ① 2~Nの値を配列に順番に格納する。

2	3	4	5	6	7	8	9	10	11	12	13	14	...	N
---	---	---	---	---	---	---	---	----	----	----	----	----	-----	---

- ② 配列の先頭から順番に値を調べ、その値が0でなければ、その値の倍数を消去する。消去は、倍数を格納している配列の要素に0を代入することで行う。

最初の値は2なので、2の倍数を消去する。

2	3	0	5	0	7	0	9	0	11	0	13	0	...	N
---	---	---	---	---	---	---	---	---	----	---	----	---	-----	---

次の値は3なので、3の倍数を消去する。

2	3	0	5	0	7	0	0	0	11	0	13	0	...	N
---	---	---	---	---	---	---	---	---	----	---	----	---	-----	---

次の値は0なので消去は行わず、その次の値である5の倍数を消去する。

2	3	0	5	0	7	0	0	0	11	0	13	0	...	N
---	---	---	---	---	---	---	---	---	----	---	----	---	-----	---

倍数の基になる値が $2 \sim \sqrt{N}$ まで②の消去処理を繰り返し、消去されなかった（配列内の値が0でない）値が素数となる。

なお、配列の要素位置は0から始まるものとする。

表 SOSU の引数の仕様

変数名	入力／出力	意味
N	入力	素数を求める範囲の最大値

<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) ~ (4) の解答群

ア. $i \leftarrow 0$

イ. $i \leftarrow 2$

ウ. $i \leftarrow i + 1$

エ. $i \leftarrow i + a[i]$

オ. $j \leftarrow 0$

カ. $j \leftarrow 2$

キ. $j \leftarrow j + 1$

ク. $j \leftarrow i + a[i]$

ケ. $j \leftarrow j + a[i]$

<設問 2 > N に 100 を与えてプログラムを実行した場合、**Ⓐ** の処理は何回実行されるか、解答群から選べ。

(5) の解答群

ア. 2

イ. 4

ウ. 8

エ. 10

< 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 | 18 ページ～21 ページ |
| ・ 表計算の問題 | 22 ページ～26 ページ |
| ・ アセンブラの問題 | 28 ページ～33 ページ |

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、各設問に答えよ。

[プログラムの説明]

水泳競技の一種である飛込競技の採点をするプログラムである。

飛込競技の審判員は5名または7名であり、各審判員は10点を最高点として0.5点単位で採点する。審判員が5名の場合は、各審判員の点数から最高点と最低点を除いた3つの点数を合計する。また、審判員が7名の場合は、点数の高い2つと点数の低い2つを除いた3つの点数を合計する

[関数の説明]

saiten 関数

引 数：審判員数，点数の配列。

機 能：各審判員の点数を全て合計し，点数の中で最大の値と最小の値を引いて合計点を計算する。ただし，審判員数が7の場合は2番目に大きい値と2番目に小さい値も含めた4つの値を引く。

戻り値：合計点。

<設問1> 次のプログラム1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1]

```
#define MEMBER7      7
#define NOTHING      (-1)
#define DUMMY_MAX    (-1.0)
#define DUMMY_MIN    100.0
#define USED          0
#define UNUSED       1

float saiten(int n, float tensu[]) {
    int    i, max_idx, min_idx, chk_tbl[MEMBER7];
    float  gokei, max, min;
    /* チェック用配列の初期化 */
    for(i=0; i<n; i++) chk_tbl[i] = USED;
    /* 仮の最大値と最小値の設定 */
    max = DUMMY_MAX;
    min = DUMMY_MIN;
    /* 最大値と最小値の要素位置を初期化 */
    max_idx = NOTHING;
    min_idx = NOTHING;
    for(i=0; i<n; i++) {
```

```

gokei += tensu[i];
if ( (1) ) {
    if (max_idx != NOTHING) (2);
    (3);
    max = tensu[i];
    max_idx = i;
}
if ( (4) ) {
    if (min_idx != NOTHING) (5);
    (3);
    min = tensu[i];
    min_idx = i;
}
}
gokei -= (max + min);
/* 審判員が7人の場合2番目に大きい値と小さい値を引く */
if (n == MEMBER7) {
    max = DUMMY_MAX;
    min = DUMMY_MIN;
    for(i=0; i<n; i++) {
        if (chk_tbl[i] == USED) {
            if ( (1) ) {
                max = tensu[i];
            } else if ( (4) ) {
                min = tensu[i];
            }
        }
    }
    gokei -= (max + min);
}
return gokei;
}

```

(1) の解答群

- | | |
|---|---|
| ア. <code>tensu[i] > max</code> | イ. <code>tensu[i] < max</code> |
| ウ. <code>tensu[max_idx] > max</code> | エ. <code>tensu[min_idx] < max</code> |

(2) , (3) , (5) の解答群

- | | |
|---|---|
| ア. <code>chk_tbl[i] = USED</code> | イ. <code>chk_tbl[i] = UNUSED</code> |
| ウ. <code>chk_tbl[max_idx] = USED</code> | エ. <code>chk_tbl[max_idx] = UNUSED</code> |
| オ. <code>chk_tbl[min_idx] = USED</code> | カ. <code>chk_tbl[min_idx] = UNUSED</code> |

(4) の解答群

ア. `tensu[i] > min`

イ. `tensu[i] < min`

ウ. `tensu[min_idx] > min`

エ. `tensu[min_idx] < min`

<設問 2> 次のプログラム 2 の説明を読み、プログラム 2 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 2 の説明]

審判員数が 5 名でも 7 名でも、点数の順に並べた場合、中央の 3 つの点数を集計することに注目してプログラム 1 を変更した。新しい集計方法は、審判員の点数を昇順に並べ替え、配列の中央に位置する値と、その前後に位置する値を加えることで求める。

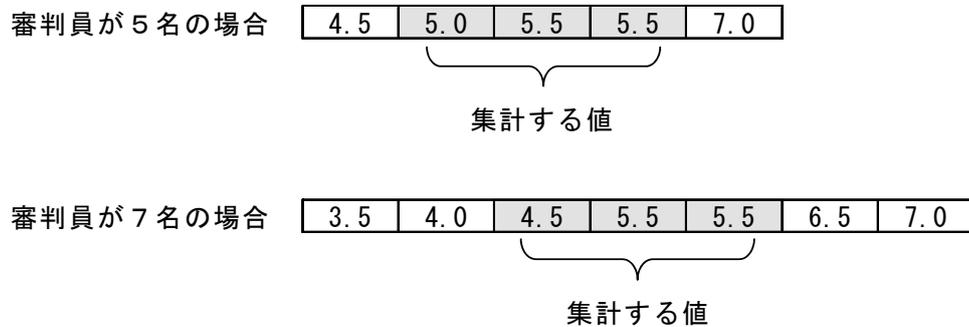


図 1 昇順に並べ替えた場合に集計する位置

プログラム 2 では、配列 `tensu` に受取った点数を配列 `data` へ昇順に並ぶように挿入法で転送する。ここでは、挿入法を以下の考え方で行う。

- ① 配列の p 番目まで昇順に並んでいるとする。
- ② 以下の動作を p が 0 以上であり、配列の p 番目の値が配列に転送しようとする値より大きい間繰り返す。
 - ・配列の p 番目の値を $p+1$ 番目に移動する。
 - ・ p から 1 を引く。
- ③ ② の繰り返しで空けた場所に値を転送する。

[プログラム 2]

```
#define MEMBER7 7

float saiten(int n, float tensu[]) {
    int    i, j;
    float  data[MEMBER7];
    /* 配列 tensu のデータを配列 data へ並べ替えて転送する */
    for(i=0; i<n; i++) {
        j = i - 1;
        while( j >= 0 && (6) ) {
            data[j+1] = data[j];
            j--;
        }
        (7);
    }
    (8);
    return data[i-1] + data[i] + data[i+1];
}
```

(6) の解答群

- ア. `data[i] > tensu[j]`
- ウ. `data[i] > data[j]`

- イ. `data[j] > tensu[i]`
- エ. `tensu[j] > tensu[i]`

(7) の解答群

- ア. `data[i] = tensu[j]`
- ウ. `data[j] = tensu[i]`

- イ. `data[i+1] = tensu[j]`
- エ. `data[j+1] = tensu[i]`

(8) の解答群

- ア. `i = n / 2`
- ウ. `i = n / 2 + 1`

- イ. `i = n / 2 - 1`
- エ. `i = (n + 1) / 2`

次の表計算ソフトの記述を読み、各設問に答えよ。

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。

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。

IF 関数の中にIF 関数を入れることができる。

書式：IF (条件式, 真の場合, 偽の場合)

ROUNDUP関数

指定した桁で値を切り上げる。桁数が正の数であれば小数点以下、負の数であれば小数点以上の桁になる。例えば、1 にすると小数点以下第2位以下の桁を切り上げて小数点以下第1位までを表示し、-1 にすると1の位以下の桁を切り上げる。

書式：ROUNDUP (式または値, 桁数)

ROUNDDOWN 関数

指定した桁で値を切り捨てる。桁数が正の数であれば小数点以下、負の数であれば小数点以上の桁になる。例えば、1 にすると小数点以下第2位以下の桁を切り捨てて小数点以下第1位までを表示し、-1 にすると1の位以下の桁を切り捨てる。

書式：ROUNDDOWN (式または値, 桁数)

SUM 関数

指定した範囲の合計値を求める。

書式：SUM (開始セル番地：終了セル番地)

HOUR関数

時間を取り出す。0～23(時)の範囲の整数になる。

書式：HOUR(シリアル値)

MINUTE関数

分を取り出す。0～59(分)の範囲の整数になる。

書式：MINUTE(シリアル値)

TODAY関数：

現在の日付に対応するシリアル値を返し、日付形式で表示される。

書式：TODAY ()

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。

他のワークシートの参照

「ワークシートの名前!セル番地」とすることで他のワークシートのセルを参照することができる。

Jカラオケ店では、表計算ソフトを使用して料金の計算と年間集計表を作成している。

<設問1> 次のレシート作成に関する記述中の に入れるべき適切な式を解答群から選べ。

表1 料金表

	A	B	C	D	E
1	料金表				
2					
3		30分ごと		フリータイム料金	
4		料金	会員料金	料金	会員料金
5	一般	150	100	1000	800
6	学生	80	50	600	400

※ フリータイムとは、11:00~18:00の間であり、
料金は定額である。

表2 レシート

	A	B	C	D	E	F	G	H
1	レシート							
2								
3	会員番号	30246				日付	2012/4/4	
4								
5	フリータイム利用		0					
6								
7	開始時刻	11時02分						
8	終了時刻	15時54分				利用時間	4時52分	
9								
10	部屋番号		人数	30分ごとの単価	フリータイムの単価	料金		
11	205	一般	2	100	800	2,000		
12		学生	3	50	400	1,500		
13								
14					合計料金	3,500		
15								

受付で、ワークシート「レシート」にカラオケ利用に必要な項目を入力すると料金計算が行われるように作成する。料金表の料金は、入力済みである。

- ・会員番号は、会員であるお客から会員カードを借りて入力する。会員でないお客の場合は 0 と入力する。
- ・日付は、システム日付が表示される。
- ・フリータイムを利用するかをお客に確認後、利用の場合は 1 を、利用しない場合は 0 を入力する。1 部屋にフリータイムと時間利用が混在することはない。
- ・開始時刻、終了時刻を入力すると、利用時間が表示される。
- ・利用部屋番号と利用人数を入力する。学生の場合は、学生証の提示で確認を行う。
- ・時間利用の場合は、利用時間を 30 分単位に切り上げた金額になる。

セル E11 に ‘30 分ごとの単価’ を求めセル E12 へ、セル F11 に ‘フリータイムの単価’ を求めセル F12 へ、セル G11 に人数分の料金を求めセル G12 へそれぞれ複写する。セル E11 に入力する式は となり、セル F11 に入力する式は となり、セル G11 に入力する式は となる。合計料金を求める式をセル G14 に入力する。

(1), (2) の解答群

- | | |
|--------------------------------|--------------------------------|
| ア. =IF(C3=0, 料金表!B5, 料金表!C5) | イ. =IF(C\$3=0, 料金表!B5, 料金表!C5) |
| ウ. =IF(\$C3=0, 料金表!B5, 料金表!C5) | エ. =IF(C3=0, 料金表!D5, 料金表!E5) |
| オ. =IF(C\$3=0, 料金表!D5, 料金表!E5) | カ. =IF(\$C3=0, 料金表!D5, 料金表!E5) |

(3) の解答群

- ア. =IF(D5=0, (HOUR(G\$8)/2+ROUNDUP(MINUTE(G\$8)/30, 0))*D11*E11, D11*F11)
- イ. =IF(D\$5=0, (HOUR(G\$8)/2+ROUNDUP(MINUTE(G\$8)/30, 1))*D11*E11, D11*F11)
- ウ. =IF(D\$5=0, (HOUR(G\$8)*2+ROUNDDOWN(MINUTE(G\$8)/30, 0))*D11*E11, D11*F11)
- エ. =IF(D5=0, (HOUR(G\$8)*2+ROUNDUP(MINUTE(G\$8)/30, 1))*D11*E11, D11*F11)
- オ. =IF(D\$5=0, (HOUR(G\$8)*2+ROUNDUP(MINUTE(G\$8)/30, 0))*D11*E11, D11*F11)
- カ. =IF(\$D5=0, (HOUR(G\$8)*2+ROUNDDOWN(MINUTE(G\$8)/30, 0))*D11*E11, D11*F11)

<設問 2> 次の年間売上一覧表とグラフ作成に関する記述中の□に入れるべき適切な式を解答群から選べ。

表 3 年間売上一覧表

	A	B	C	D	E
1	年間売上一覧表				
2					単位:千円
3	月	2010年	2011年	2011年累計	移動年計
4	1	1,350	2,263	2,263	23,022
5	2	1,140	1,281	3,544	23,163
6	3	2,458	2,321	5,865	23,026
7	4	2,375	1,327	7,192	21,978
8	5	1,844	1,867	9,059	22,001
9	6	1,695	1,327	10,386	21,633
10	7	1,376	1,156	11,542	21,413
11	8	2,083	1,482	13,024	20,812
12	9	1,942	1,436	14,460	20,306
13	10	1,753	1,657	16,117	20,210
14	11	1,564	1,342	17,459	19,988
15	12	2,529	1,984	19,443	19,443
16	合計	22,109	19,443		

ワークシート「レシート」に入力された売上高は、ワークシート「年間売上一覧表」の年と月の一致するセルに集計される。

次に、グラフを作成するため、2011年累計と移動年計（その月を含めた過去12カ月分の合計）を表示する。2011年累計を求めるため、セルD4に入力する式は□(4)となり、セルD5に入力する式は□(5)となる。ただし、セルD5に入力した式は、セルD6～D15まで複写する。

移動年計を求めるため、セルE4に入力する式は□(6)となり、セルE5に入力する式は□(7)となる。ただし、セルE5に入力した式は、セルE6～E15まで複写する。

次に、図のようなグラフを作成する。このグラフは、売上の季節変動など含めた長期的な傾向を見ることができ、□(8)と呼ばれている。

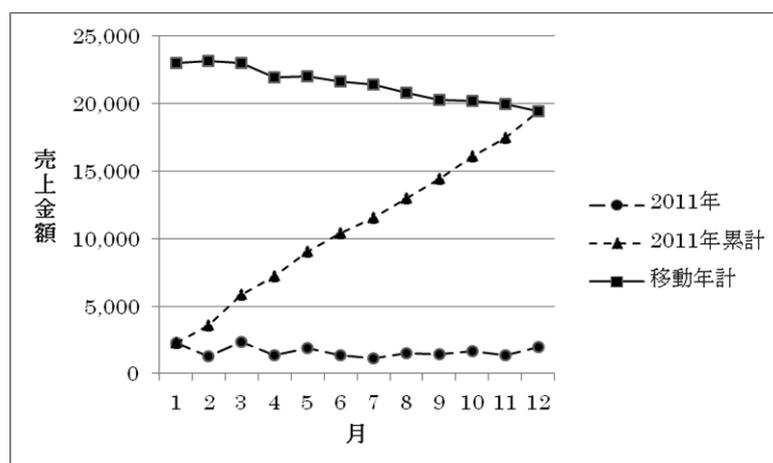


図 年間売上累計グラフ

(4) , (5) の解答群

ア. =B4

ウ. =D4+B5

イ. =C4

エ. =D4+C5

(6) の解答群

ア. =SUM(A5:B15)+C4

ウ. =SUM(B5:B15)+C4

イ. =SUM(A5:B15)+D4

エ. =SUM(B5:B15)+D4

(7) の解答群

ア. =E4+B5+C5

ウ. =E4-B5+C5

イ. =E4+B5-C5

エ. =E4-B5-C5

(8) の解答群

ア. SD チャート

ウ. Z チャート

イ. J チャート

エ. 積み上げ面グラフ

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]

給与など現金支払いする場合の金種計算をする副プログラム SUBK である。

支払金額は 100 万円未満であり、MAN 番地から始まる連続した 2 語の領域に、1 万円以上の値と 1 万円未満の値に分けて格納されている(図 1)。

MAN 番地+0	45
+1	2736

図 1 支払金額が 452,736 円の場合

金種は、KINSHU 番地以降の連続した領域に格納されており、副プログラム SUBK により計算された金種別枚数を MAISU 番地以降に格納する(図 2)。

なお、2000 円札は含まないものとする。

KINSHU 番地+0	10000	MAISU 番地+0		…10000 円札の枚数
+1	5000	+1		…5000 円札の枚数
+2	1000	+2		…1000 円札の枚数
+3	500	+3		…500 円硬貨の枚数
+4	100	+4		…100 円硬貨の枚数
+5	50	+5		…50 円硬貨の枚数
+6	10	+6		…10 円硬貨の枚数
+7	5	+7		…5 円硬貨の枚数
+8	1	+8		…1 円硬貨の枚数

図 2 金種および枚数の格納領域

副プログラム SUBK は図 3 の形式で、パラメタが格納された先頭番地を GR1 に設定して呼び出される。

(GR1)+0	MAN 番地
+1	KINSHU 番地
+2	MAISU 番地

図 3 SUBK に渡されるパラメタ

<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。

[プログラム]

行番号	ラベル	命令	オペランド	コメント
100	SUBK	START		
110		RPUSH		
120		LD	GR2, 0, GR1	;MAN 番地のアドレス
130		LD	GR3, 1, GR1	;KINSHU 番地のアドレス
140		LD	GR4, 2, GR1	;MAISU 番地のアドレス
150		LAD	GR5, 8, GR4	;終了アドレス
160		ST	GR5, LASTAD	
170		LD	GR0, 0, GR2	;1万円以上のデータ
180		ST	GR0, 0, GR4	
190		LAD	GR2, 1, GR2	
200			<input type="text"/> (1)	;1万円未満のデータ
210	LOOP1	LD	GR5, =0	;枚数の初期化
220		LAD	GR3, 1, GR3	;金種の変更
230			<input type="text"/> (2)	
240		CPA	GR4, LASTAD	
250		JZE	L2	
260	LOOP2	SUBA	GR0, 0, GR3	;データ÷金種
270		JMI	L1	
280		LAD	GR5, 1, GR5	;枚数のカウント
290		JUMP	LOOP2	
300	L1		<input type="text"/> (3)	
310		ST	GR5, 0, GR4	
320		JUMP	LOOP1	
330	L2		<input type="text"/> (4)	
340		RPOP		
350		RET		
360	LASTAD	DS	1	
370		END		

(1) の解答群

ア. LD GR0,0,GR2

ウ. LD GR0,0,GR4

イ. LD GR0,1,GR2

エ. LD GR0,1,GR4

(2) の解答群

ア. LD GR2,GR3

ウ. LAD GR2,1,GR2

イ. LD GR4,GR3

エ. LAD GR4,1,GR4

(3) の解答群

ア. ADDA GR0,0,GR3

ウ. LAD GR3,1,GR3

イ. ADDA GR0,0,GR4

エ. LAD GR4,1,GR4

(4) の解答群

ア. ST GR0,0,GR3

ウ. ST GR3,0,GR2

イ. ST GR0,0,GR4

エ. ST GR3,0,GR4

<設問 2> 次の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字句を解答群から選べ。

[プログラムの説明]

設問 1 の副プログラム SUBK を利用して、複数件のデータを処理するプログラムである。

データは DAT+1 番地以降の連続した領域に、1 件につき 2 語を使用して格納されている。また、DAT 番地にはデータ件数が格納されており(図 4)、1 件以上 30 件以内とする。

DAT 番地+0	3	…データ件数
+1	45	… 1 件目 2 語で 452,736 円
+2	2736	
+3	28	… 2 件目 2 語で 289,461 円
+4	9461	
+5	32	… 3 件目 2 語で 324,917 円
+6	4917	

図 4 DAT 番地以降のデータ例

データを 1 件ずつ、MAN 番地以降の 2 語に設定して副プログラム SUBK を呼び出す。副プログラムにより求められた金種の枚数を、GOKEI 番地以降に集計し、データ全体での各金種枚数を求める(図 5)。

GOKEI 番地+0	105	…10000 円札の枚数
+1	1	…5000 円札の枚数
+2	10	…1000 円札の枚数
+3	2	…500 円硬貨の枚数
+4	10	…100 円硬貨の枚数
+5	1	…50 円硬貨の枚数
+6	5	…10 円硬貨の枚数
+7	2	…5 円硬貨の枚数
+8	4	…1 円硬貨の枚数

図 5 GOKEI 番地以降の例(図 4 のデータの場合)

[プログラム]

行番号	ラベル	命令	オペランド	コメント
500	PROG	START		
510		LD	GR0,=0	
520		LD	GR1,=8	
530	CLEAR	ST	GR0,GOKEI,GR1	;GOKEI 領域でクリア
540		SUBA	GR1,=1	
550		JPL	CLEAR	
560		JZE	CLEAR	
570		LAD	GR3,MAN	
580		LAD	GR4,1,GR3	
590		LD	GR2,DAT	
600			(5)	
610	LOOP	LD	GR0,DAT,GR2	
620		ST	GR0,0,GR4	
630		SUBA	GR2,=1	
640		LD	GR0,DAT,GR2	
650			(6)	
660		SUBA	GR2,=1	
670		LAD	GR1,PARM	
680		CALL	SUBK	;SUBK 呼び出し
690		LD	GR5,=8	
700	SHUKEI	LD	GR0,GOKEI,GR5	
710			(7)	
720		ST	GR0,GOKEI,GR5	
730		SUBA	GR5,=1	
740		JPL	SHUKEI	
750		JZE	SHUKEI	
760		CPA	GR2,=0	
770		JPL	LOOP	
780		RET		
790	PARM	DC	MAN	;SUBK 用パラメタ
800		DC	KINSHU	
810		DC	MAISU	
820	MAN	DS	2	
830	KINSHU	DS	9	;金種のデータ格納済
840	MAISU	DS	9	
850	GOKEI	DS	9	
860	DAT	DS	61	;30 件以内
870		END		

(5) の解答群

ア. LAD GR2,1,GR2

ウ. SLA GR2,1

イ. LAD GR2,2,GR2

エ. SRA GR2,1

(6) の解答群

ア. ST GR0,0,GR3

ウ. ST GR0,0,GR4

イ. ST GR0,1,GR3

エ. ST GR0,1,GR4

(7) の解答群

ア. ADDA GR0,MAISU,GR2

ウ. ADDA GR0,MAISU,GR4

イ. ADDA GR0,MAISU,GR3

エ. ADDA GR0,MAISU,GR5

