

令和2年度後期 情報検定

<実施 令和3年2月14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）、スマートフォン、タブレット、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、電卓付き腕時計、時計型ウェアラブル端末等
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32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～15 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 | 17 ページ～21 ページ |
| ・ 表計算の問題 | 22 ページ～28 ページ |
| ・ アセンブラの問題 | 29 ページ～32 ページ |

必須問題

問題 1 次のデータ構造に関する記述を読み、各設問に答えよ。

リストとは、データ構造の一つであり、データの格納場所を示すポインタによって複数のデータが連結されたものである。

図 1 は単方向リストといい、先頭から順番にポインタをたどってデータにアクセスし、逆方向にたどれない特徴がある。

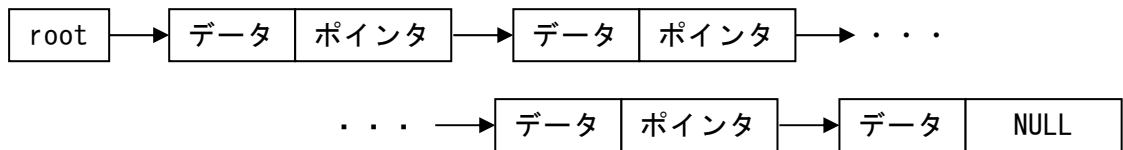


図 1 単方向リスト

root は最初のデータが格納されている場所を示すポインタである。

ポインタは、次のデータが格納されている場所を示すものであり、最後のデータのポインタには NULL が入る。

単方向リストをメモリ上に展開したものが図 2 である。データは全てが昇順に並んでいる。

| | 番地 | 内容 | |
|------|------|------|-------|
| root | 1000 | 1012 | …データ |
| | 1001 | 1012 | …ポインタ |
| | 1002 | 1004 | |
| | 1003 | 1010 | |
| | 1004 | 1024 | |
| | 1005 | NULL | |
| | 1006 | 1000 | |
| | 1007 | 1002 | |
| | 1008 | 1020 | |
| | 1009 | 1004 | |
| | 1010 | 1008 | |
| | 1011 | 1000 | |
| | 1012 | 1016 | |
| | 1013 | 1008 | |
| | 1014 | 1018 | |
| 1015 | | | |

図 2 メモリ上の単方向リスト

<設問 1 > 次の単方向リストの操作に関する記述中の [] に入れるべき適切な字句を解答群から選べ。

図 2 の単方向リストから 1000 番地にあるデータ「1012」を削除する場合を考える。その場合、1011 番地にあるポインタを [(1)] に変更する。

また、1014 番地にある「1018」を単方向リストに挿入する場合を考える。その場合、次の 2 つの手順で挿入することができる。

- ① 1013 番地にあるポインタを [(2)] に変更する。
- ② 1015 番地にあるポインタを [(3)] に変更する。

(1) ~ (3) の解答群

- ア. 1000 イ. 1004 ウ. 1008 エ. 1012
- オ. 1014 カ. 1018 キ. 1022 ク. 1026

<設問 2 > 次のリストと配列に関する記述中の [] に入れるべき適切な字句を解答群から選べ。

データ数が 100 個で、あらかじめ昇順に並べ替えられている配列にデータの挿入と削除を行う場合のデータ移動回数を考える。なお、配列要素は操作に必要な要素数を十分に確保している。

配列にデータを挿入する場合は、データを挿入したい位置以降のデータを、配列の最後尾のデータから一つずつ後ろに移動していく。また、配列からデータを削除する場合は、削除したデータ以降のデータを一つずつ前に移動する。

ここで、先頭から 30 番目にデータを挿入するとき、データの移動回数は [(4)] 回になる。また、先頭から 70 番目のデータを削除するとき、データの移動回数は [(5)] 回になる。

リスト構造でデータを挿入、削除するときはデータの移動は [(6)] 。

(4) , (5) の解答群

- ア. 30 イ. 31 ウ. 70 エ. 71

(6) の解答群

- ア. ない
- イ. 配列と同様の回数となる
- ウ. 配列より多くの回数となる

問題2 次の流れ図の説明を読み、各設問に答えよ。

[流れ図の説明]

1次元配列 $tbl[0] \sim tbl[4]$ までデータが格納済みである。このデータに対して挿入法によりデータを昇順に整列する流れ図である。

[挿入法]

昇順に整列されている配列に新たなデータを挿入していく。データの挿入位置を見つけるため、配列の最後尾から順に挿入データと比較し、挿入データより小さい値を見つけ、隣接する位置に挿入する。

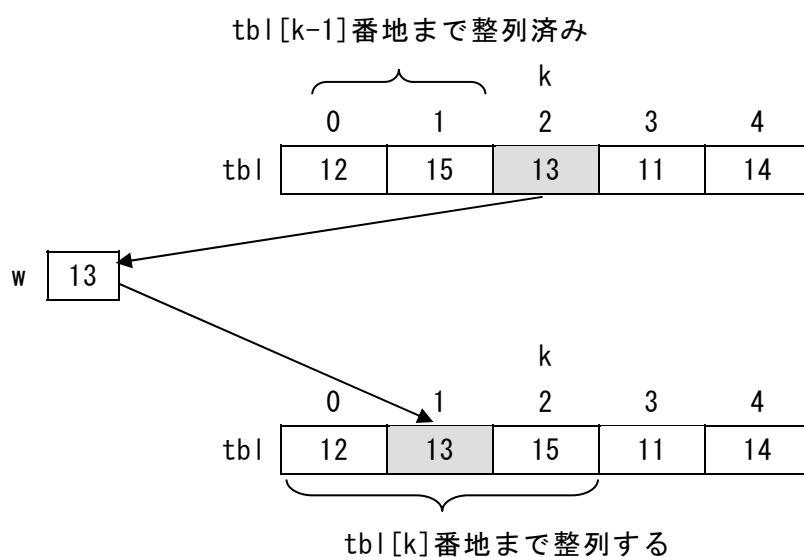


図1 挿入法の例(k=2の場合)

<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。

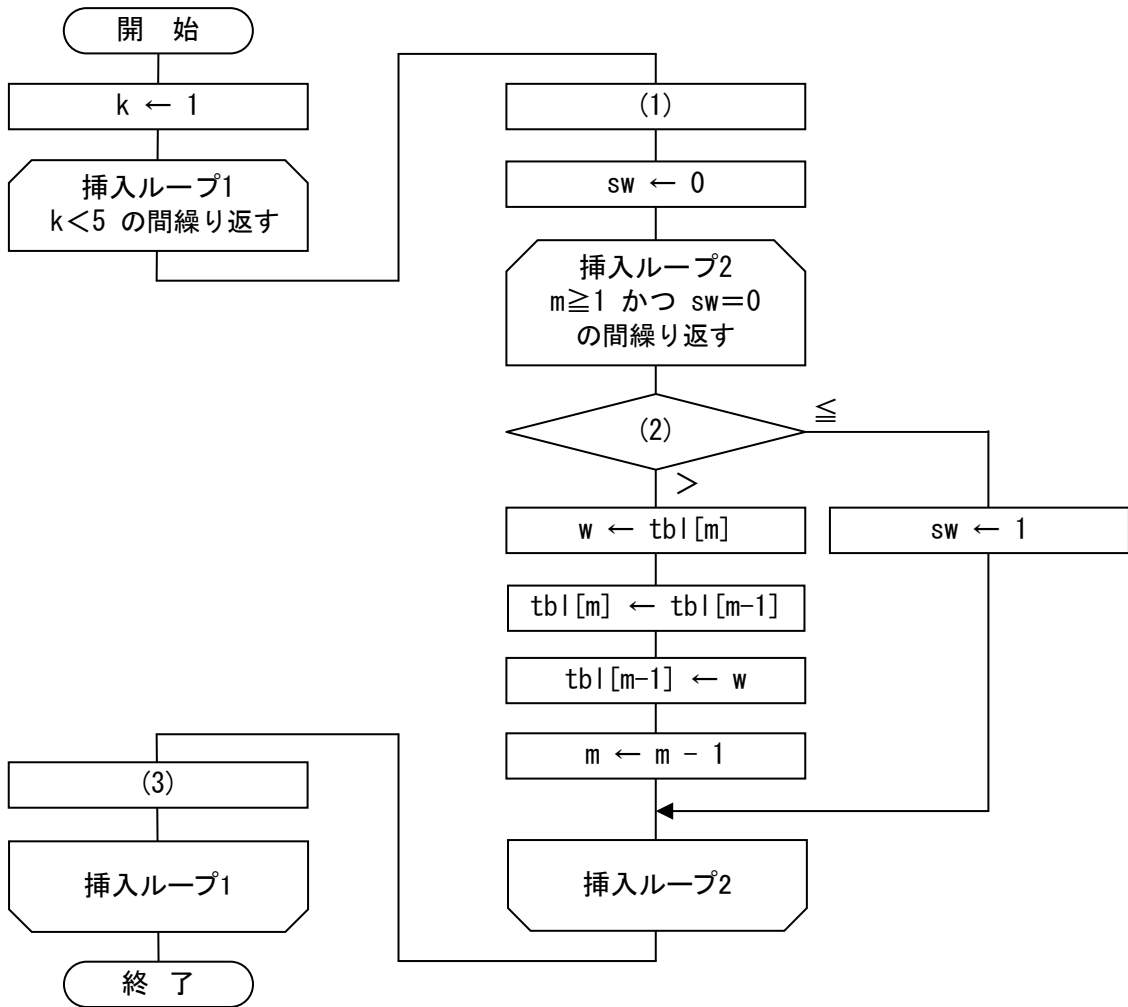


図 2 挿入法の流れ図

(1) の解答群

ア. $m \leftarrow 0$

イ. $m \leftarrow k-1$

ウ. $m \leftarrow k$

(2) の解答群

ア. $tbl[k-1] : tbl[k]$

イ. $tbl[m-1] : tbl[m]$

ウ. $tbl[m] : tbl[m-1]$

(3) の解答群

ア. $k \leftarrow k-1$

イ. $k \leftarrow k+1$

ウ. $k \leftarrow k+m$

<設問 2 > 次の整列が進んでいく過程中的の に入れるべき適切な配列を解答群から選べ。

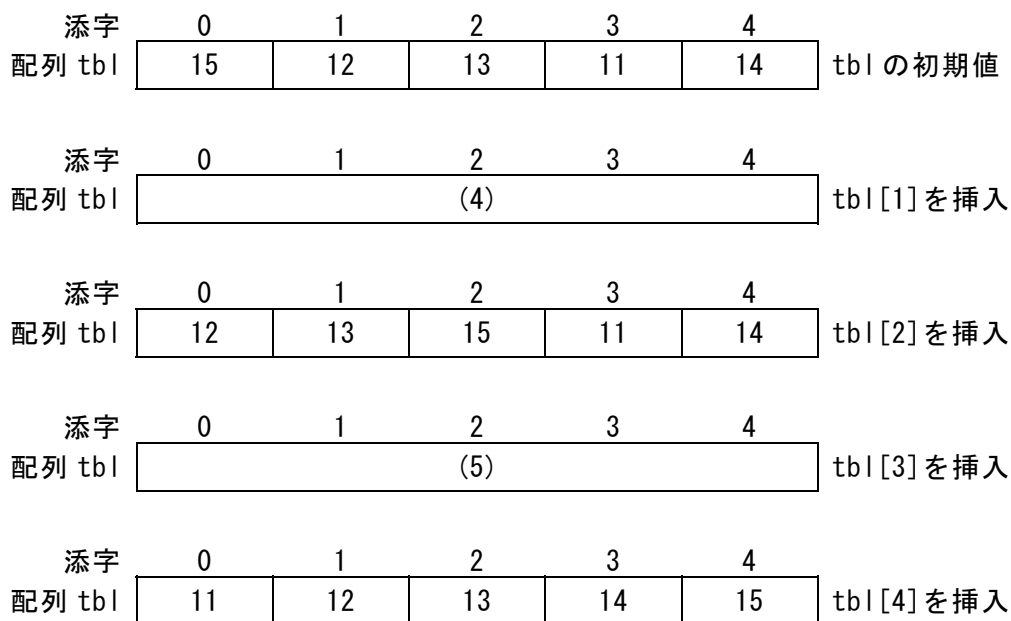


図 3 整列が進んでいく過程

(4) , (5) の解答群

- ア.

| | | | | |
|----|----|----|----|----|
| 11 | 12 | 13 | 15 | 14 |
|----|----|----|----|----|
- イ.

| | | | | |
|----|----|----|----|----|
| 11 | 15 | 12 | 14 | 13 |
|----|----|----|----|----|
- ウ.

| | | | | |
|----|----|----|----|----|
| 12 | 13 | 15 | 11 | 14 |
|----|----|----|----|----|
- エ.

| | | | | |
|----|----|----|----|----|
| 12 | 15 | 13 | 11 | 14 |
|----|----|----|----|----|
- オ.

| | | | | |
|----|----|----|----|----|
| 14 | 15 | 12 | 11 | 13 |
|----|----|----|----|----|

問題3 次のグループ集計に関する記述を読み、各設問に答えよ。

グループ集計とは、キー項目の値が同じデータを集計する処理のことである。例えば、日付、地域、売上数で構成するデータがあり、地域ごとに売上数の合計を求める場合は、キー項目となる値は地域になり、集計する対象は売上数となる。

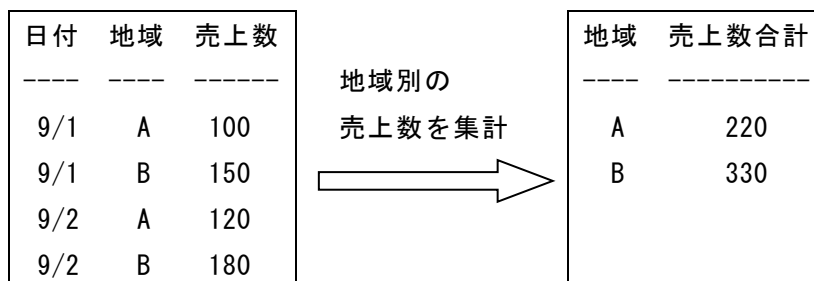


図1 地域別の売上数を集計

コンピュータ内部では、すべてのデータを一度に認識して集計することはできない。キー項目の並びがランダムである場合は、配列を用いた処理で対応できる。

<設問1> 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]

ファイルから日付、地域、売上数で構成するレコードを入力し、地域ごとに売上数の合計を表示するものである。

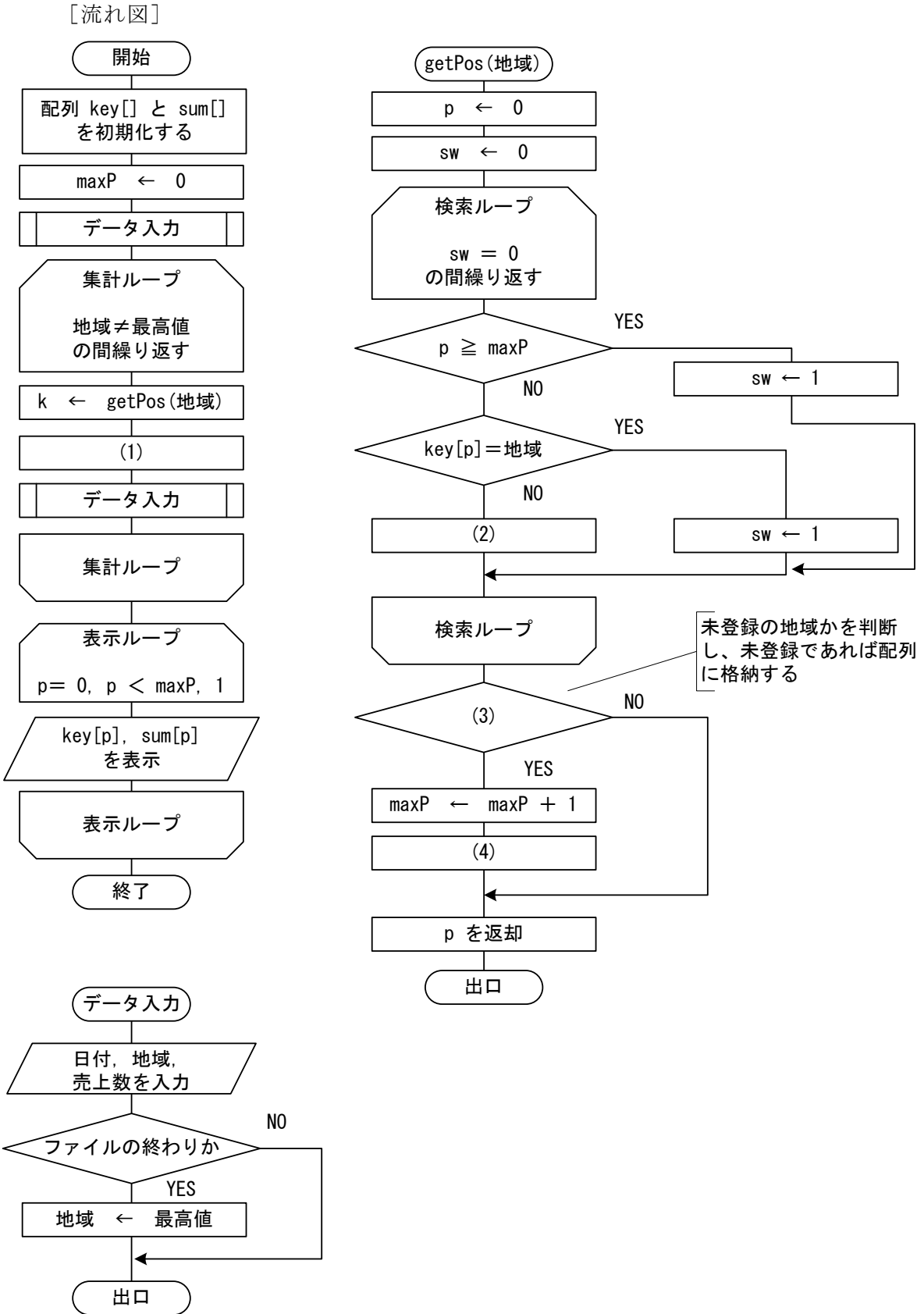
ファイルの内容はランダムに並んでおり、集計のために次の配列を利用する。

- ・配列 key[] … 地域を保存する配列
- ・配列 sum[] … 売上数の合計を求める配列

それぞれの配列は、集計に必要な大きさを持っており、地域を格納している位置と、その地域の売上数の合計を格納する位置は同じである。また、配列の要素位置は0から始まる。

流れ図中で用いているgetPosは、地域を引数に与えることにより、その地域が配列keyのどの位置に格納されているかを返す。引数に与えた地域が配列keyに存在しなければ、その地域を配列keyに追加する。

なお、流れ図中で用いている“最高値”は、コンピュータで扱える値の中で一番大きな値を意味し、データとして用いられることはない。



(1) の解答群

- ア. $\text{sum}[p] \leftarrow \text{sum}[p] + \text{売上数}$ イ. $\text{sum}[k] \leftarrow \text{sum}[k] + \text{売上数}$
- ウ. $\text{sum}[\text{maxP}] \leftarrow \text{sum}[\text{maxP}] + \text{売上数}$ エ. $\text{sum}[p] \leftarrow \text{sum}[\text{maxP}] + \text{売上数}$

(2) の解答群

- ア. $p \leftarrow p + 1$ イ. $p \leftarrow p + \text{maxP}$
- ウ. $\text{maxP} \leftarrow \text{maxP} + 1$ エ. $\text{maxP} \leftarrow \text{maxP} + p$

(3) の解答群

- ア. $p \geq \text{maxP}$ イ. $p \neq \text{maxP}$
- ウ. $\text{key}[p] = \text{地域}$ エ. $\text{key}[p] \neq \text{最高値}$

(4) の解答群

- ア. $\text{key}[p] \leftarrow \text{地域}$ イ. $\text{key}[\text{maxP}] \leftarrow \text{地域}$
- ウ. $\text{sum}[p] \leftarrow \text{売上数}$ エ. $\text{sum}[\text{maxP}] \leftarrow \text{売上数}$

<設問 2> 次の並べ替えたデータの集計に関する記述を読み、流れ図中の に入れるべき適切な字句を解答群から選べ。

グループ集計を行う方法として、キー項目で並べ替える方法がある。これは、キー項目で並び替えることで同じキー項目の値が連続することを利用して

この性質を利用し、1レコードずつ入力しながら以前のキー項目と比較し、同じ値であれば集計を、異なる値であれば集計結果を表示することで処理する。

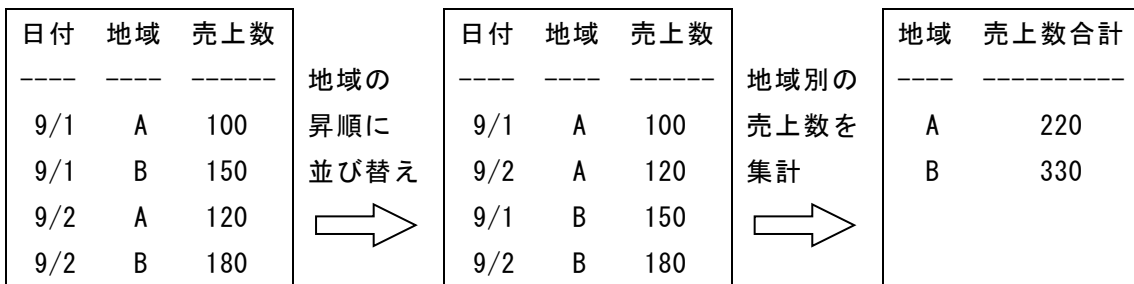


図 3 地域別に並べ替えて売上数を集計

次の流れ図は、ファイルから日付、地域、売上数で構成するレコードを入力し、地域ごとに売上数の合計を表示するものである。ただし、入力するレコードは地域の昇順に整列済みである。

[流れ図]

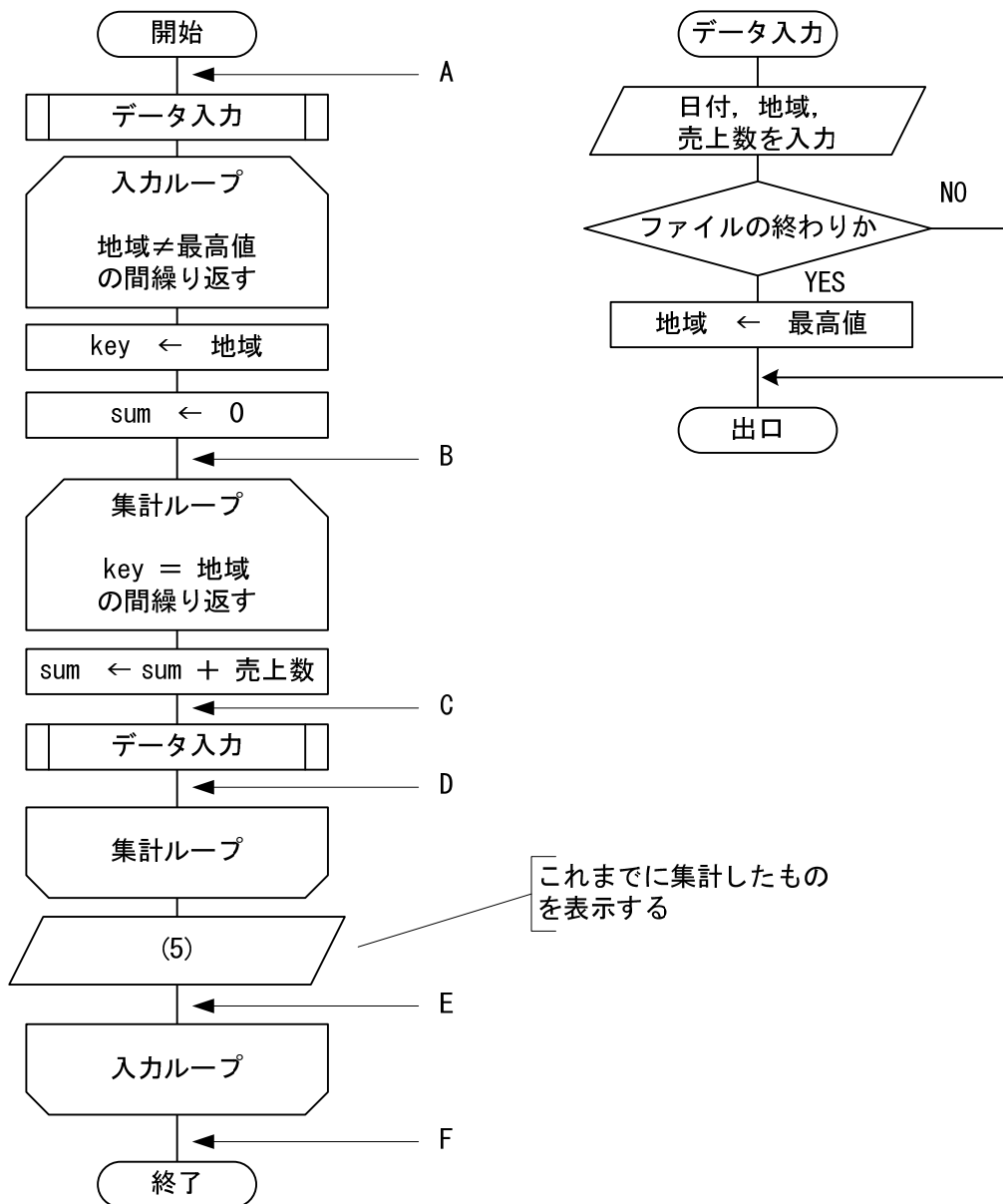


図4 地域別に並べ替えたデータの売上数を集計する流れ図

(5) の解答群

- ア. key, sum を表示
- ウ. 地域, sum を表示

- イ. key, 売上数を表示
- エ. 地域, 売上数を表示

<設問 3 > 次の集計処理の拡張に関する記述を読み、図 4 の流れ図に挿入する処理 (6) ~ (8) の位置を解答群から選べ。

各地域の集計値を表示した最後に、総合計を表示する処理を考える。

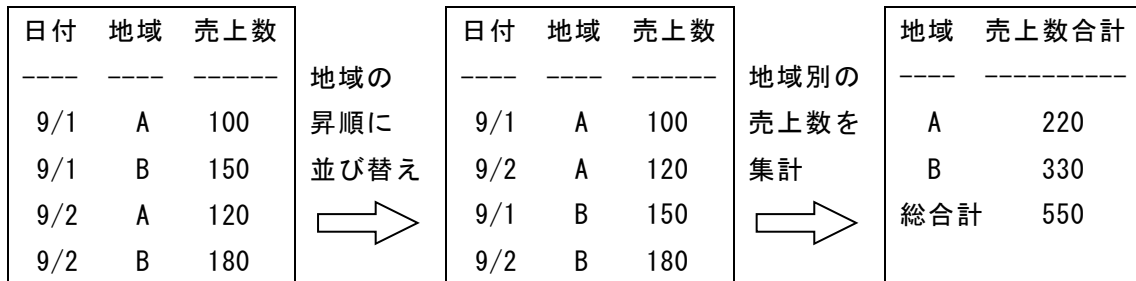


図 5 地域別の売上数と総合計を集計

総合計を求める変数を total とした場合、次のような処理が必要になる。それぞれの処理は図 4 の流れ図 A~F のいずれかに入る。

[挿入する処理]

- (6) total ← 0
- (7) total ← total + sum
- (8) "総合計", total を表示

(6) ~ (8) の解答群

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

問題4 次のプログラムの説明および擬似言語の記述を読み、各設問に答えよ。

探索は、記録されている多数のデータの中から条件を満たすデータを探し出す処理である。データが記録された配列内の特定のデータを探索する方法には様々なものがある。

[プログラム1の説明]

データが記録されている1次元配列 dat の先頭要素から順に目的のデータ X を探索し、先頭からの位置を返す関数 s_search である。1次元配列 dat の大きさは十分に確保されており、データ数はN個とし、添字は0から始まる。なお、要素中に存在する目的のデータには重複がなく、存在しない場合は-1を返す。

図1の探索の例の場合、目的のデータ X は 21 であり、配列 dat[5] に格納されているので、添え字 5 を返す。

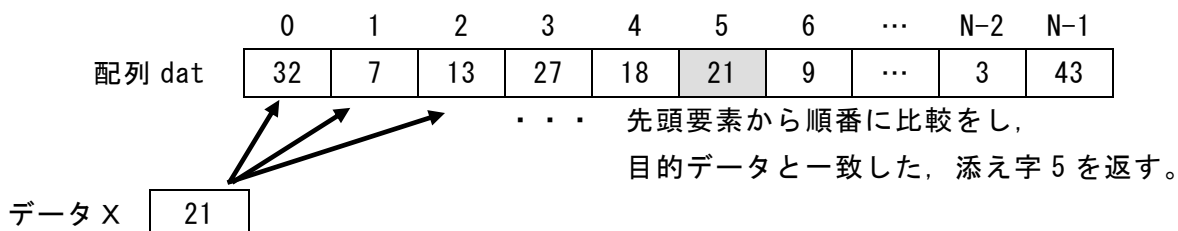


図1 プログラム1の探索例

[プログラム2の説明]

プログラム1と同様、データが記録されている1次元配列 dat の先頭要素から順に目的のデータ X を探索し、先頭からの位置を返す関数 b_search である。ただし、探索前に配列の最後に目的のデータ X を追加し、目的のデータが当初の配列 dat に存在しない場合でも必ず最後に一致することにする。

図2の探索の例の場合、目的のデータ X は 15 であり、配列 dat[0] ~ 配列 dat[N-1] に存在しない場合、探索の最後に配列 dat[N] と一致するが、探索前に追加したデータであるため -1 を返す。

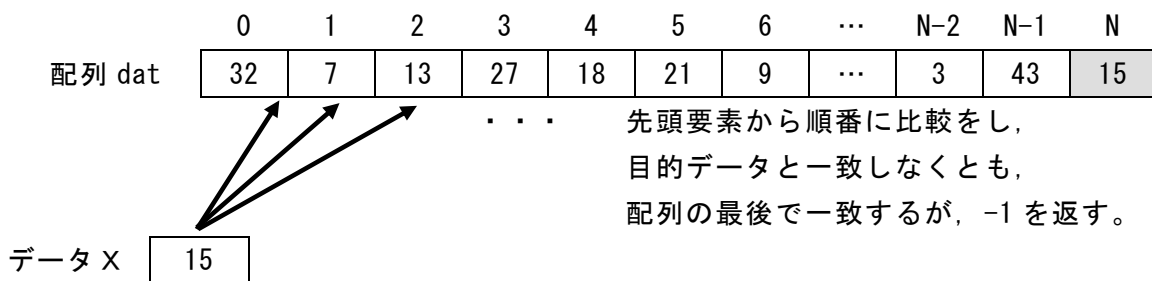


図2 プログラム2の探索例

[擬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 ─── ■ | 前判定繰返し処理を示す。 条件式が真の間, 処理を実行する。 |

[演算子と優先順位]

| 演算の種類 | 演算子 | 優先順位 |
|-------|------------------|----------------------|
| 単項演算 | +, -, not | 高 ↑ ↓ 低 |
| 乗除演算 | *, /, % | |
| 加減演算 | +, - | |
| 関係演算 | >, <, ≥, ≤, =, ≠ | |
| 論理積 | and | |
| 論理和 | or | |

注記 整数同士の除算では, 整数の商を結果として返す。%演算子は剰余算を表す。

[プログラム 1]

```

○s_search (整数型 : x, 整数型 : N, 整数型 : dat[])
○整数型 : k, m
・ (1)
/* 配列の中から x を探索する */
・ k ← 0
■ k < N                ← α
  ▲ dat[k] = x         ← β
  │  ・ m ← k
  │  ・ break          /* 繰返し処理から抜け出す */
  ───
  ▼  ・ k ← k + 1
  ▼
■
・ return m
  
```

[プログラム 2]

○ b_search (整数型 : x, 整数型 : N, 整数型 : dat[])

○ 整数型 : k, m

•

/* x の保存 */

•

/* 配列の中から x を探索する */

• k ← 0

■ dat[k] ≠ x ← γ

• k ← k + 1

■

▲

• m ← k

▼

• return m

<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. m ← -1

イ. m ← 0

ウ. m ← k

エ. m ← N

(2) の解答群

ア. dat[0] ← x

イ. dat[N-1] ← x

ウ. dat[N] ← x

エ. dat[N-2] ← x

(3) の解答群

ア. k < N

イ. k < N + 1

ウ. k = N

エ. k > N

<設問 2 > 次の比較回数に関する記述中の に入れるべき適切な字句を解答群から選べ。

プログラム 1 の s_search では、1 回の繰り返しごとに α と β の条件比較を行っている。ここで、配列の要素数を 10 とし、配列のデータは、dat[0]~dat[9]に格納済みとする。目的のデータが配列 dat に存在しない場合も含め、 α と β の条件比較の合計回数は最大 (4) 回である。また、プログラム 2 の b_search では、目的のデータが配列 dat に存在しない場合も含め、 γ の条件比較は最大 (5) 回である。

(4) , (5) の解答群

ア. 9

イ. 10

ウ. 11

エ. 12

オ. 21

カ. 22

< 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 | 17 ページ～21 ページ |
| ・ 表計算の問題 | 22 ページ～28 ページ |
| ・ アセンブラの問題 | 29 ページ～32 ページ |

次のヒープに関する記述を読み、各設問に答えよ。

[ヒープについて]

二分木構造において、親の節の値が子の節の値より常に大きいか等しい（または小さいか等しい）という制約を満たすものをヒープと呼ぶ。この問題では、親の値が子の値より常に大きいか等しいものを扱う。

図1にヒープの例を示す。図は「○」で節を表しており、「○」の中の数値が節の値である。また、子を持つ節を親と呼び、親を持たない節を根と呼ぶ。ヒープの制約に従えば、根の値は節の中で一番大きな値となる。なお、子の節が1つの場合は左の子とし、子の節が2つある場合の並びは節の数値と関係ない。

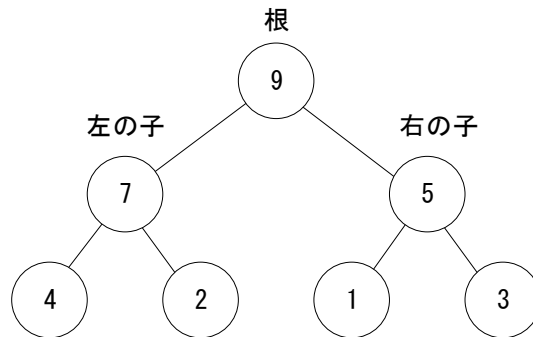


図1 ヒープの例

ヒープは一次元配列で実装できる。図1のヒープを一次元配列hに展開したものが図2である。

| 位置 | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|----|---|---|---|---|---|---|---|
| h | 9 | 7 | 5 | 4 | 2 | 1 | 3 |

図2 ヒープを一次元配列に格納した例

根はh[0]であり、親の節の位置をpとすれば子の節の位置は次の計算で求められる。

- ・左の子 $p \times 2 + 1$
- ・右の子 $p \times 2 + 2$

<設問 1> 次の図 3 のヒープを配列 h に展開したものを解答群から選べ。なお、配列の要素位置は 0 から始まるものとする。

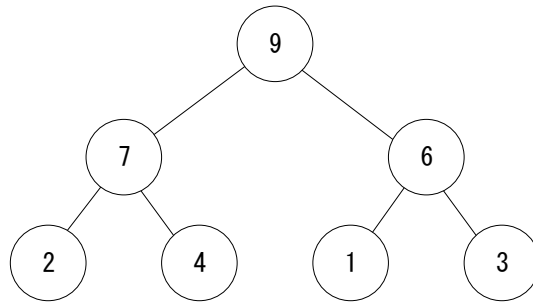


図 3 ヒープ

(1) の解答群

- ア.

| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 6 | 7 | 9 |
|---|---|---|---|---|---|---|
- イ.

| | | | | | | |
|---|---|---|---|---|---|---|
| 9 | 6 | 7 | 2 | 4 | 1 | 3 |
|---|---|---|---|---|---|---|
- ウ.

| | | | | | | |
|---|---|---|---|---|---|---|
| 9 | 7 | 6 | 2 | 4 | 1 | 3 |
|---|---|---|---|---|---|---|
- エ.

| | | | | | | |
|---|---|---|---|---|---|---|
| 9 | 7 | 6 | 4 | 3 | 2 | 1 |
|---|---|---|---|---|---|---|

[ヒープの構築について]

配列 h に格納されたランダムな値に対して次の手順によりヒープを構築する。

1. 配列 h 中の末端にある親の位置を求め、これを parent とする。これは、配列の要素数を size とすれば、 $(size - 2) \div 2$ で求めることができる。ただし、小数点以下は切り捨てる。
2. parent が 0 以下になるまで 1 ずつ小さくしながら「3」の処理を繰り返す。
3. 親の位置を parent, 左の子の位置を child, 右の子の位置を child+1 とすれば、h[parent], h[child], h[child+1] の最大値を h[parent] に格納する。
 - ・親の値が最大値である場合
繰り返しを終える。
 - ・どちらかの子の値が最大値である場合
h[parent] と最大値であるどちらかの子の値を交換し、子の値が格納されていた位置を新しい親の位置 (parent) に設定し、左の子の位置 (child) を計算する。この時、child が示す位置がヒープを構築する範囲を超えるようであれば繰り返しを終える。

<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]

一次元配列 `h` に格納されたランダムな値をヒープの構築に従って再配置する `makeHeap` 関数である。`makeHeap` 関数からは `downHeap` 関数を呼び出し、`downHeap` 関数からは `swap` 関数を呼び出す。

[関数の説明]

`makeHeap` 関数

引 数：`h`(ランダムな値が格納される配列)、`size`(配列 `h` の要素数)

機 能：ヒープを構築するため、末端の親から前の親へ遷移しながら `downHeap` 関数を呼び出す

戻り値：なし

`downHeap` 関数

引 数：`h`(ランダムな値が格納される配列)、`parent`(親の位置)、`last` (ヒープを構築する範囲の最後の位置)

機 能：親と子要素の中から最大値を見つけ親に設定する。

戻り値：なし

`swap` 関数

引 数：`data1`(整数値)、`data2`(整数値)

機 能：`data1` と `data2` の値を入れ替える

戻り値：なし

[プログラム]

```
void makeHeap(int *h, int size) {
    int parent;
    /* 末端の親の位置から構築する */
    for(parent = (2); parent >= 0; parent--) {
        downHeap(h, parent, size - 1);
    }
}

void downHeap(int *h, int parent, int last) {
    int sw, child;
    sw = 0;
    child = (3); /* 左の子の位置を求める */
    while(child <= last && (4)) {
        if (child + 1 <= last) { /* 大きい方の位置を child に設定 */
            if ((5)) {
                child++;
            }
        }
    }
}
```

```

        if (h[parent] < h[child]) { /* 親要素と子要素の比較 */
            swap(&h[parent], &h[child]);
            parent = ;
            child = ;
        } else {
            sw = 1;
        }
    }
}

void swap(int *data1, int *data2) {
    int work;
    work = *data1;
    *data1 = *data2;
    *data2 = work;
}

```

(2) の解答群

- | | |
|------------------------------|--------------------------------|
| ア. <code>zize - 1</code> | イ. <code>size * 2 - 1</code> |
| ウ. <code>size * 2 + 1</code> | エ. <code>(size - 2) / 2</code> |

(3) の解答群

- | | |
|--------------------------------|--------------------------------|
| ア. <code>parent * 2 - 1</code> | イ. <code>parent * 2</code> |
| ウ. <code>parent * 2 + 1</code> | エ. <code>parent * 2 + 2</code> |

(4) の解答群

- | | |
|-----------------------------------|-----------------------------------|
| ア. <code>parent < child</code> | イ. <code>parent > child</code> |
| ウ. <code>sw == 0</code> | エ. <code>sw == 1</code> |

(5) の解答群

- | | |
|--|--|
| ア. <code>h[child] > h[child - 1]</code> | イ. <code>h[child] < h[child - 1]</code> |
| ウ. <code>h[child] > h[child + 1]</code> | エ. <code>h[child] < h[child + 1]</code> |

(6) の解答群

- | | |
|---------------------------------|----------------------------|
| ア. <code>(child - 1) * 2</code> | イ. <code>child</code> |
| ウ. <code>parent - 1</code> | エ. <code>parent + 1</code> |

(7) の解答群

- | | |
|--------------------------------|--------------------------------|
| ア. <code>child * 2 - 1</code> | イ. <code>child * 2</code> |
| ウ. <code>parent * 2 - 1</code> | エ. <code>parent * 2 + 1</code> |

[ヒープソートについて]

ここでは、ヒープが構築された配列の先頭は必ず一番大きな値になる。これを利用して配列内を並び替えるものがヒープソートである。

ヒープソートにより昇順に並び替えるには、次の手順で行う。

1. ヒープを構築する。
2. 配列の最後の位置を last とすれば、last を 1 ずつ減らしながら「3」の処理を繰り返す。
3. 配列の先頭と last で示す位置にある値を入れ替え、配列の先頭から last - 1 までの範囲でヒープを構築する。

<設問 3> 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]

ランダムな値が格納された配列 h をヒープソートにより並べ替える heapSort 関数である。プログラム中で呼び出す makeHeap, downHeap, swap は、設問 2 で用いたものである。

[関数の説明]

heapSort 関数

引 数 : h(ランダムな値が格納される配列), size(配列の要素数)

機 能 : ヒープを構築しながら配列の要素を入れ替えて整列する

戻り値 : なし

[プログラム]

```
void heapSort(int *h, int size) {  
    int i, last;  
    makeHeap(h, size);  
    last = size - 1;  
    for(i=0; i<size - 1; i++) {  
         (8);  
        last--;  
        downHeap(h, 0, last);  
    }  
}
```

(8) の解答群

- | | |
|------------------------------|--------------------------|
| ア. swap(&h[0], &h[last - 1]) | イ. swap(&h[0], &h[last]) |
| ウ. swap(&h[1], &h[last - 1]) | エ. swap(&h[1], &h[last]) |

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。

COUNTIF 関数

範囲に含まれるセルのうち、条件に一致するセルの個数を返す。

書式：COUNTIF(範囲, 条件)

IF 関数

条件が真のときに真の場合、偽のときに偽の場合の計算結果や値を返す。

書式：IF(条件, 真の場合, 偽の場合)

INDEX 関数

範囲の中から行位置と列位置で指定したセルの値を返す。位置は 1 から始まる相対値である。

書式：INDEX(範囲, 行位置, 列位置)

LEFT 関数

文字列の左端から文字数で指定した位置までの文字列を返す。

書式：LEFT(文字列, 文字数)

MATCH 関数

検査範囲から検査値が存在するセルの相対的な位置を返す。位置は 1 から始まる相対的な値である。検査範囲は 1 行または 1 列である。検査の型は、検査値と等しい最初の値を検索する場合は 0、検査値以下の最大の値を検索する場合は 1、検査値以上の最小の値を検索する場合は -1 を指定する

書式：MATCH(検査値, 検査範囲, 検査の型)

RANK 関数

範囲内の数値を並べたときに何番目になるか(順位)を返す。順序は、降順の場合は 0、昇順の場合は 1 を設定する。なお、範囲内の検査値に同じものがあれば同じ順位を返し、以降の順位に欠番が生じる。

書式：RANK(検査値, 範囲, 順序)

SUMIF 関数

指定した検索範囲の中で、条件に一致するセルの合計範囲に対応するセルの値の合計値を返す。

書式：SUMIF(検索範囲, 条件, 合計範囲)

TODAY関数

現在の日付に対応するシリアル値を返す。

書式：TODAY()

VLOOKUP 関数

検索範囲から検索値を探し、位置で指定した列の値を返す。位置は1から始まる相対的な値であり、検索範囲中に見つけた行の中で、左から何番目の列かを示す。検索方法は0または1を指定し、0の場合は完全に一致する値を、1の場合は検索値以下の最大値を探す。

書式：VLOOKUP(検索値, 検索範囲, 位置, 検索方法)

式

=に続いて計算式や関数などを入力する。

セル番地の絶対参照

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、別のワークシートを参照できる。

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

J図書館では、表計算ソフトを使用して図書管理情報を作成している。会員は300名おり、「会員表」ワークシートにまとめられている。2行目からA列に会員番号、B列に氏名、C列に性別、D列に生年月日が301行まで入力されている。図書分類情報は「分類表」ワークシートにまとめており、2行目からA列にカテゴリ、B列に分類コード、C列に分類名が27行まで入力されている。現在貸出可能な書籍は2000冊あり、「本情報」ワークシートにまとめている。2行目からA列に図書コード、B列に図書名、C列に著者、D列に出版社、E列に出版年が2001行まで入力されている。なお、図書コードの先頭から2文字が分類コードになっている。

| | A | B | C | D |
|-----|-------|--------|----|------------|
| 1 | 会員番号 | 氏名 | 性別 | 生年月日 |
| 2 | 13589 | 関根 三郎 | 男 | 1960/12/6 |
| 3 | 16372 | 中村 光良 | 男 | 1986/2/1 |
| 4 | 17794 | 秋元 美佐 | 女 | 1990/7/13 |
| 5 | 18575 | 徳永 三省 | 男 | 1978/1/7 |
| 6 | 21594 | 大原 そら | 女 | 1967/8/26 |
| : | : | : | : | : |
| 300 | 59305 | 伊藤 奈央 | 女 | 1971/3/23 |
| 301 | 64422 | 久野 裕次郎 | 男 | 1970/11/11 |

図1 「会員表」ワークシート

| | A | B | C |
|----|--------|-------|-------|
| 1 | カテゴリ | 分類コード | 分類名 |
| 2 | 一般書 | A1 | 日本の文学 |
| 3 | | A2 | 日本の小説 |
| 4 | | A3 | 世界の文学 |
| 5 | | A4 | 郷土資料 |
| 6 | | : | : |
| : | : | : | : |
| 26 | バリアフリー | F1 | 視覚 |
| 27 | | F2 | 聴覚 |

図2 「分類表」ワークシート

| | A | B | C | D | E |
|------|----------|---------------|--------|--------|---------|
| 1 | 図書コード | 図書名 | 著者 | 出版社 | 出版年 |
| 2 | A1003487 | 宇津の巫女 | 富士川 佳祐 | 角丸出版 | 1999.3 |
| 3 | A1005626 | 海の闇と月の影 | 八武崎 知恵 | 秀英社 | 1999.5 |
| 4 | A1018574 | わたしたちの犬 | 和田 真一 | 秀英社 | 2009.2 |
| 5 | A1092736 | ニッポンの好感度 | 細川 美古 | 秋桑社 | 2010.12 |
| 6 | A1203876 | 一度は訪ねたい徒然ふるさと | 文月 絢 | 川出書房新社 | 2018.8 |
| : | : | : | : | : | : |
| 2000 | F2000231 | はじめての手話 | 安西 絵美 | 角丸出版 | 2019.8 |
| 2001 | F2000357 | 思い出の町並み | 高橋 敬 | 秀英社 | 2020.8 |

図3 「本情報」ワークシート

<設問1> 次の「貸出表」ワークシートの作成に関する記述中の に入れるべき適切な字句を解答群から選べ。

| | A | B | C | D | E | F | G |
|----|-------------------|---------------|-------------------|-----------|---|---|---|
| 1 | 貸出表 | | | | | | |
| 2 | | | | | | | |
| 3 | 会員番号 <u>18575</u> | | 会員氏名 <u>徳永 三省</u> | | | | |
| 4 | | | | | | | |
| 5 | 図書コード | 図書名 | 貸出日 | 返却期日 | | | |
| 6 | A1203876 | 一度は訪ねたい徒然ふるさと | 20XX/9/3 | 20XX/9/10 | | | |
| 7 | C2033071 | おほしさまとおつきさま | 20XX/9/3 | 20XX/9/10 | | | |
| 8 | C2053360 | がたんごとん | 20XX/9/3 | 20XX/9/10 | | | |
| 9 | | | | | | | |
| 10 | | | | | | | |
| 11 | | | | | | | |

図4 「貸出表」ワークシート

- セルD1とB3, D3, B5~F5の項目名を入力し、セルの結合などを行って書式を整えた。
- 3行目の会員氏名は、「会員表」ワークシートから会員番号で検索をし、表示する。セルE3に次の式を入力した。

=

- ・ 6～10行目の図書コードには、貸出希望の本の図書コードを入力した。なお、1回の貸出で上限5冊までであり、5冊未満の場合は空白とする。

- ・ C列は図書名を表示する。「本情報」ワークシートから図書番号で検索をして表示する。なお、図書コードが入力されなければ、図書名は表示しない。セルC6に次の式を入力し、セルC7～C10まで複写した。

= IF(B6="", "",

- ・ E列は貸出日(処理当日)を表示する。なお、図書コードが入力されなければ、貸出日は表示しない。セルE6に次の式を入力し、セルE7～E10まで複写した。

=

- ・ F列は返却期日を表示する。返却期日は貸出日の1週間後とする。なお、図書コードが入力されなければ、返却期日は表示しない。セルF6に次の式を入力し、セルF7～F10まで複写した。

=

(1) の解答群

- ア. VLOOKUP(B3, 会員表!A2:B301, 2, 0)
- イ. VLOOKUP(B3, 会員表!B2:B301, 1, 0)
- ウ. VLOOKUP(C3, 会員表!A2:B301, 2, 0)
- エ. VLOOKUP(C3, 会員表!B2:B301, 1, 0)

(2) の解答群

- ア. INDEX(本情報!\$A2:\$B2001, MATCH(B\$6, 本情報!\$A2:\$A2001, 0), 2)
- イ. INDEX(本情報!A\$2:B\$2001, MATCH(B6, 本情報!A\$2:A\$2001, 0), 2)
- ウ. INDEX(本情報!A\$2:B\$2001, MATCH(B6, 本情報!A2:A2001, 0), 2)
- エ. INDEX(本情報!A2:B2001, MATCH(B6, 本情報!A2:A2001, 0), 2)

(3) の解答群

- ア. IF(B6<>"", TODAY(), "")
- イ. IF(B\$6<>"", TODAY(), "")
- ウ. IF(B6<>"", "", TODAY())
- エ. TODAY()

(4) の解答群

- ア. E6 + 1
- イ. E6 + 7
- ウ. IF(B6<>"", E6+1, "")
- エ. IF(B6<>"", E6+7, "")

<設問2> 次の「貸出集計表」ワークシートの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。

「貸出表」ワークシートに貸出情報が入力されると「貸出情報」ワークシートにもまとめられる。2行目からA列に図書コード、B列に貸出日、C列に返却日、D列に貸出日数、E列に延滞日数が5001行まで入力されている。なお、返却日は返却された日が入力され、貸出日数は貸出した日と返却した日も含めて求められる。延滞日数は、返却期日から返却が遅れた日数である。

| | A | B | C | D | E |
|------|----------|-----------|-----------|------|------|
| 1 | 図書コード | 貸出日 | 返却日 | 貸出日数 | 延滞日数 |
| 2 | C2031583 | 20XX/9/1 | 20XX/9/3 | 3 | 0 |
| 3 | C2033071 | 20XX/9/1 | 20XX/9/3 | 3 | 0 |
| 4 | A1003487 | 20XX/9/1 | 20XX/9/16 | 16 | 8 |
| 5 | A1005626 | 20XX/9/1 | 20XX/9/5 | 5 | 0 |
| 6 | A1018574 | 20XX/9/1 | 20XX/9/6 | 6 | 0 |
| : | : | : | : | : | : |
| 5000 | G1001276 | 20XX/9/30 | 20XX/10/6 | 7 | 0 |
| 5001 | F2000231 | 20XX/9/30 | 20XX/10/7 | 8 | 0 |

図5 「貸出情報」ワークシート

「貸出情報」ワークシートから本ごとの貸出状況を確認できる「貸出集計表」ワークシートを作成した。

| | A | B | C | D | E | F |
|------|----------|------|--------|--------|-------|--------|
| 1 | 図書コード | 貸出回数 | 合計貸出日数 | 平均貸出日数 | 分類名 | 貸出回数順位 |
| 2 | A1003487 | 6 | 19 | 3.2 | 日本の文学 | 13 |
| 3 | A1005626 | 9 | 23 | 2.6 | 日本の文学 | 4 |
| 4 | A1018574 | 3 | 17 | 5.7 | 日本の文学 | 16 |
| 5 | A1092736 | 0 | 0 | 0.0 | 日本の文学 | 1973 |
| 6 | A1203876 | 10 | 23 | 2.3 | 日本の文学 | 1 |
| : | : | : | : | : | : | : |
| 2000 | F2000231 | 1 | 15 | 15.0 | 聴覚 | 1534 |
| 2001 | F2000357 | 4 | 16 | 4.0 | 聴覚 | 983 |

図6 「貸出集計表」ワークシート

- セルA1～F1に項目名を入力し、A列に「本情報」ワークシートから図書コードを複写した。
- B列は貸出回数を表示する。セルB2に次の式を入力し、セルB3～B2001まで複写した。

=

- ・ C 列は合計貸出日数を表示する。セル C2 に次の式を入力し、セル C3～C2001 まで複写した。

=

- ・ D 列は平均貸出日数を表示する。セル D2 に次の式を入力し、セル D3～D2001 まで複写した。

=

- ・ E 列は分類名を表示する。セル E2 に次の式を入力し、セル E3～E2001 まで複写した。

=

- ・ F 列は貸出回数の多い順位を表示する。セル F2 に次の式を入力し、セル F3～F2001 まで複写した。

=

(5) の解答群

- ア. COUNTIF(貸出情報!A\$2:A\$5001, A2)
- イ. COUNTIF(貸出情報!A2:A5001, A2)
- ウ. SUMIF(貸出情報!A\$2:A\$5001, A2)
- エ. SUMIF(貸出情報!A2:A5001, A2)

(6) の解答群

- ア. COUNTIF(貸出情報!A\$2:A\$5001, A2, 貸出情報!D\$2:D\$5001)
- イ. COUNTIF(貸出情報!A2:A5001, A2, 貸出情報!D2:D5001)
- ウ. SUMIF(貸出情報!A\$2:A\$5001, A2, 貸出情報!D\$2:D\$5001)
- エ. SUMIF(貸出情報!A2:A5001, A2, 貸出情報!D2:D5001)

(7) の解答群

- ア. B2 / C2
- イ. C2 / B2
- ウ. IF(B2<>0, 0, C2/B2)
- エ. IF(B2<>0, C2/B2, 0)

(8) の解答群

- ア. VLOOKUP(LEFT(A\$2, 2), 分類表!B\$2:C\$27, 2, 0)
- イ. VLOOKUP(LEFT(A\$2, 2), 分類表!B2:C27, 2, 0)
- ウ. VLOOKUP(LEFT(A2, 2), 分類表!B\$2:C\$27, 2, 0)
- エ. VLOOKUP(LEFT(A2, 2), 分類表!B2:C27, 2, 0)

(9) の解答群

ア. RANK(B\$2, B\$2:B\$2001, 0)

ウ. RANK(B\$2, B\$2:B\$2001, 1)

イ. RANK(B2, B\$2:B\$2001, 0)

エ. RANK(B2, B\$2:B\$2001, 1)

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]

PAC番地に格納されている4桁のパック10進数形式(ただし符号部は省略してある)を10進数に変換してDEC番地に格納するプログラムNUMである。

なお、行番号380のFUGO番地は設問3で使用するためのものである。

[パック10進数形式の説明]

パック10進数形式は、10進数の1桁を4ビットで表し、最下位バイトの下位4ビットで符号を表す。符号は正数なら $(1100)_2$ 負数なら $(1101)_2$ である。

| | | | | |
|------|------|------|------|--------|
| 0100 | 0000 | 1001 | 0110 | 1101 |
| 4 | 0 | 9 | 6 | 符号部(-) |

図1 パック10進数の例

[変換の手順]

- ① 図1に示すように、PAC番地の1桁ずつを取り出して、KETA番地から始まる4語に格納する。なお、PAC番地には必ず4桁で格納する。例えば $(56)_{10}$ の場合PAC番地は $(0056)_{16}$ である。



図2 PAC番地から1桁ずつKETA番地以降に取り出す

- ② KETA番地以降から1桁ずつ取り出して以下の計算を行う。
 $(\text{千の位}) \times 1000 + (\text{百の位}) \times 100 + (\text{十の位}) \times 10 + (\text{一の位})$
 ただし、1000倍、100倍は10倍の繰り返しで求める。

[プログラム]

| 行番号 | ラベル | 命令 | オペランド | コメント |
|-----|-------|-------|----------------|----------------|
| 100 | NUM | START | | |
| 110 | | RPUSH | | |
| 120 | | LD | GR1, PAC | |
| 130 | | LD | GR3, =0 | |
| 140 | | LD | GR2, =0 | |
| 150 | LOOP1 | | (1) | |
| 160 | | SRL | GR0, 0, GR2 | |
| 170 | | AND | GR0, =#000F | |
| 180 | | ST | GR0, KETA, GR3 | |
| 190 | | | (2) | ; 次のシフト数の準備 |
| 200 | | LAD | GR3, 1, GR3 | |
| 210 | | CPA | GR3, =4 | |
| 220 | | JMI | LOOP1 | |
| 230 | | LD | GR0, =0 | |
| 240 | LOOP2 | | (3) | ; KETA 番地の指標更新 |
| 250 | | JMI | OWARI | |
| 260 | | SLA | GR0, 1 | ; 10 倍処理の開始 |
| 270 | | ST | GR0, WK | |
| 280 | | | (4) | |
| 290 | | ADDA | GR0, WK | ; 10 倍処理の終了 |
| 300 | | ADDA | GR0, KETA, GR3 | |
| 310 | | JUMP | LOOP2 | |
| 320 | OWARI | ST | GR0, DEC | |
| 330 | | RPOP | | |
| 340 | | RET | | |
| 350 | PAC | DS | 1 | |
| 360 | DEC | DS | 1 | |
| 370 | KETA | DS | 4 | |
| 380 | FUGO | DS | 1 | ; 設問 3 で使用 |
| 390 | WK | DS | 1 | |
| 400 | | END | | |

<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. LD GR0,=#000F

イ. LD GR0,GR1

ウ. LD GR0,GR2

エ. LD GR0,GR3

(2) の解答群

ア. ADDA GR2,=4

イ. ADDA GR2,=8

ウ. SLL GR2,4

エ. SLL GR2,8

(3) の解答群

ア. SUBA GR0,=1

イ. SUBA GR1,=1

ウ. SUBA GR2,=1

エ. SUBA GR3,=1

(4) の解答群

ア. SLA GR0,1

イ. SLA GR0,2

ウ. SLA GR0,3

エ. SLA GR0,4

<設問 2> 次のプログラム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。

PAC 番地の内容を $(4096)_{16}$ としてプログラムを実行すると、DEC 番地の値は 16 進数で (5) になる。

(5) の解答群

ア. 0800

イ. 1000

ウ. 2000

エ. 4000

<設問 3> 次のプログラム機能の追加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。

プログラムではパック 10 進数形式の符号を考慮していないが、符号部だけを FUGO 番地の下位 4 ビットに格納して符号の処理を追加することにした。追加する命令は次の行番号 311~315 の 5 命令であり、行番号 250 の分岐先を追加する命令の行番号 311 に分岐するように変更する。なお、FUGO 番地の上位 12 ビットには 0 が、下位 4 ビットの符号も正しいビット列が格納されている。

[変更する命令]

| 行番号 | ラベル | 命令 | オペランド | コメント |
|-----|-----|------------|-------------|------|
| 250 | | JMI | NEXT | |

[追加する命令群]

| 行番号 | ラベル | 命令 | オペランド | コメント |
|-----|-------------|-------------|----------------------------------|------------|
| 311 | NEXT | | <input type="text" value="(6)"/> | |
| 312 | | CPL | GR2, FUGO | |
| 313 | | JNZ | OWARI | ;負数でないなら分岐 |
| 314 | | | <input type="text" value="(7)"/> | |
| 315 | | ADDA | GR0,=#0001 | |

(6) の解答群

ア. **LD GR2,=#000A**

イ. **LD GR2,=#000B**

ウ. **LD GR2,=#000C**

エ. **LD GR2,=#000D**

(7) の解答群

ア. **AND GR0,=#000F**

イ. **AND GR0,=#FFFF**

ウ. **XOR GR0,=#000F**

エ. **XOR GR0,=#FFFF**

<メモ欄>

