

# 令和3年度前期 情報検定

<実施 令和3年9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）、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34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. 試験後の合否結果（合否通知）、および合格者への「合格証・認定証」はすべて、Web認証で行います。
  - ①試験実施日の翌月より情報検定（J検）Webサイト合否検索ページ及びモバイル合否検索サイト上で、デジタル「合否通知」、デジタル「合格証・認定証」が交付されます。
  - ②団体宛には合否結果一覧ほか、試験結果資料一式を送付します。
  - 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～18 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	20 ページ～24 ページ
・ 表計算の問題	26 ページ～30 ページ
・ アセンブラの問題	31 ページ～34 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。

データ構造とはデータを効率的に扱うために、一定の決まりでデータを格納する仕組み、または形式のことである。適切なデータ構造を使用することでプログラムの実行速度を向上させることができる。代表的なものにスタックとキューがある。

<設問 1> 次のスタックとキューに関する記述中の  に入れるべき適切な字句を解答群から選べ。

[スタック構造]

スタックとは後入れ先出し法 (LIFO) によりデータなどを管理するメモリ領域のことである。主に  (1) で利用されている。スタックの操作は次の 2 つである。

- push(A) A をスタックに格納する
- pop() スタックからデータを取り出す

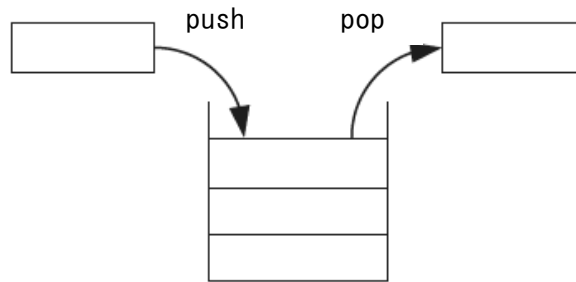


図 1 スタックへのデータの出し入れ

[キュー構造]

キューとは先入れ先出し法 (FIFO) によりデータなどを管理するメモリ領域のことである。主に  (2) に利用されている。キューの操作は次の 2 つである。

- enqueue(A) A をキューに格納する。
- dequeue() キューからデータを取り出す。



図 2 キューへのデータの出し入れ

(1) , (2) の解答群

- ア. OS の待ち行列管理など
- イ. キャッシュメモリのブロック入れ替えなど
- ウ. サブルーチンや関数の呼び出しなど
- エ. 磁気ディスク上の不要なファイルを削除など

<設問 2 > 次の前提条件を読み，処理の  に入れるべき適切な字句を解答群から選べ。

[前提条件]

- ・スタックは下記の状態から始める。なお，最後に格納されたデータは 50 である。

50
30
90
70

図 3 初期状態のスタック

- ・キューは下記の状態から始める。なお，最後に格納されたデータは 920 である。

920	420	860	390	120
-----	-----	-----	-----	-----

図 4 初期状態のキュー

[処理]

- ①初期状態のスタックに対して，次の順番で処理を実行した。

pop() → pop() → push(20) → push(10) → pop() → push(60)  
 → push(40) → pop() → pop()

実行が終了したときのスタックの状態は  (3) である。

- ②初期状態のキューに対して，次の順番で処理を実行した。

dequeue() → enqueue(220) → dequeue() → dequeue() →  
 enqueue(510) → enqueue(790) → dequeue()

実行が終了したときのキューの状態は  (4) である。

- ③初期状態のスタックとキューに対して，次の順番で処理を実行した。

enqueue(pop()) → push(dequeue()) → pop() → dequeue() → pop()  
 → push(dequeue()) → enqueue(pop())

実行が終了したときのスタックの状態は  (5) である。

実行が終了したときのキューの状態は  (6) である。

(3) の解答群

ア.

60
40
90
70

イ.

90
70
60
40

ウ.

20
90
70

エ.

40
90
70

(4) の解答群

ア.

920	510	790	220
-----	-----	-----	-----

イ.

790	510	220	920
-----	-----	-----	-----

ウ.

220	510	790
-----	-----	-----

エ.

790	510	220
-----	-----	-----

(5) の解答群

ア.

120
860

イ.

860
90

ウ.

70
90

エ.

90
70

(6) の解答群

ア.

860	50	920	420
-----	----	-----	-----

イ.

920	420	860	390
-----	-----	-----	-----

ウ.

50	920	420	860
----	-----	-----	-----

エ.

860	420	860	390
-----	-----	-----	-----

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]

ランダムに並んでいるアルファベットの文字を、アルファベットの昇順に並べ替える流れ図である。

アルファベットの文字数は10であり、1次元配列tbl[0]~tbl[9]の1要素に1文字ずつ格納済みである。ここで使用されるアルファベットはa, b, c, d, eの5種類のみであり、それ以外の文字は格納されていない。ただし、5種類すべての文字が使用されているとは限らず、また、同じ文字が複数回出現することがある。

図1に、並べ替え前後の1次元配列tbl[0]~tbl[9]を示す。

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
tbl	c	a	a	d	d	c	d	d	a	d
tbl	a	a	a	c	c	d	d	d	d	d

図1 並べ替え前（上）と並べ替え後（下）

[並べ替えの手順]

図2に示すように、使用可能な文字a~eは1次元配列moji[0]~moji[4]にあらかじめアルファベットの昇順に設定されており、各文字の1次元配列tblにおける出現回数を1次元配列cnt[0]~cnt[4]の対応位置に求める。なお、1次元配列cnt[0]~cnt[4]には初期値として0が格納済みである。

	[0]	[1]	[2]	[3]	[4]
moji	a	b	c	d	e
cnt	3	0	2	5	0

図2 使用可能文字と出現回数の配列

次に、1次元配列cntの情報をもとに、1次元配列tblにアルファベット順に使用文字を格納し、並べ替えが終了する。

<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。

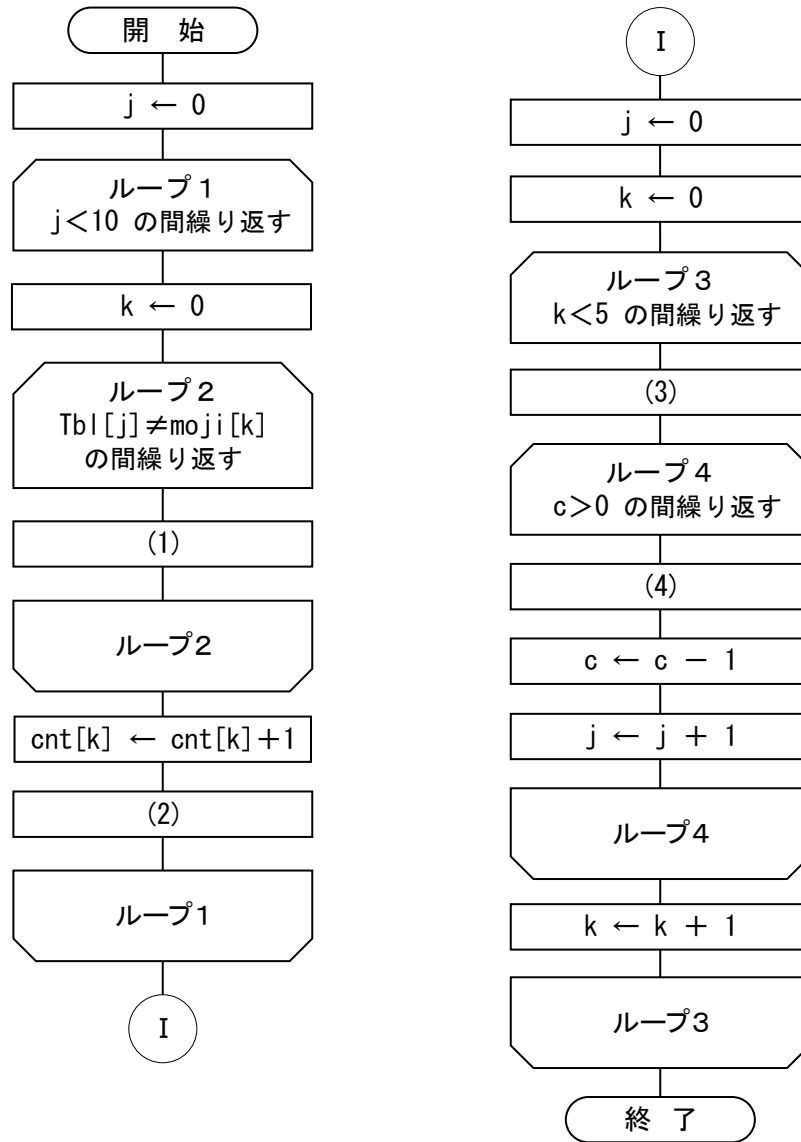


図3 並べ替えの流れ図

(1) , (2) の解答群

- ア.  $j \leftarrow j-1$       イ.  $j \leftarrow j+1$       ウ.  $k \leftarrow k-1$       エ.  $k \leftarrow k+1$

(3) の解答群

- ア.  $c \leftarrow 0$       イ.  $c \leftarrow 1$       ウ.  $c \leftarrow cnt[j]$       エ.  $c \leftarrow cnt[k]$

(4) の解答群

- ア.  $tbl[j] \leftarrow tbl[k]$       イ.  $tbl[j] \leftarrow moji[k]$   
 ウ.  $tbl[k] \leftarrow tbl[j]$       エ.  $tbl[k] \leftarrow moji[j]$



<設問 2 > 図 3 の流れ図に関する正しい記述を解答群から選べ。なお、流れ図は変更しないものとする。

(5) の解答群

- ア. 1次元配列 cnt の全要素は処理終了時に 0 になる
- イ. 1次元配列 tbl に a~e 以外の文字が紛れていた場合、流れ図ではループから抜け出すことができない
- ウ. 1次元配列 tbl を拡張し 20 文字を格納した場合でも正しく並べ替えが行える

問題3 次の文字列の検索に関する記述を読み、各設問に答えよ。

[文字列の検索について]

文字列 t から文字列 s を検索する。文字列は 1 文字ずつ配列に格納されており、文字列 t は配列 t、文字列 s は配列 s に格納される。なお、処理に必要な配列の領域は十分に確保されており、配列の添え字は 0 から始まる。

<設問 1> 次の文字列検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。

配列 s の先頭から連続する内容がすべて配列 t の一部と一致すれば、文字列が検索できたことになる(図 1)。

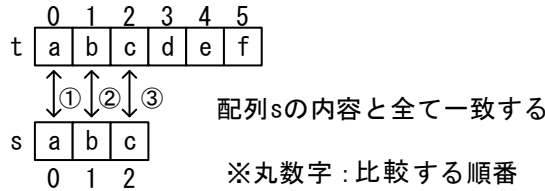


図 1 検索する文字列が一致している場合

しかし、先頭から順番に比較している途中で不一致となる場合がある。図 2 の場合、t[0]と s[0]から順番に比較を始めたが、t[2]と s[2] (③)で不一致となった。

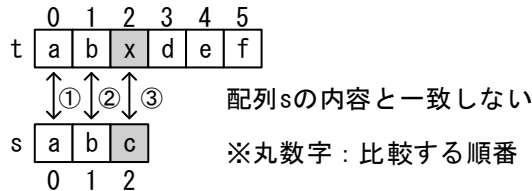


図 2 検索する文字列が不一致となる場合の例 1

次は t[1]と s[0]から順番に比較をする。図 3 の場合、t[1]と s[0] (①)で一致しない文字が出現するので、比較を中断する。次は t[2]と s[0]から順番に比較する。

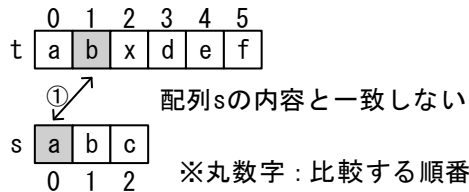


図 3 検索する文字列が不一致となる場合の例 2

このように、比較の途中で一致しない文字が出現した場合は比較を開始する配列 t の位置を 1 つずつずらす。

ここで、配列 t の文字数を tLen、配列 s の文字数を sLen、比較を開始する配列 t

の位置を  $pt$  として配列  $s[i]$  ( $i=0, 1, \dots$ ) と 1 文字ずつ順番に比較をすることを考える。比較をする要素は  $t[pt+i]$  と  $s[i]$  になるが、まだ検索対象の文字は十分残っている場合に途中で不一致になれば、次の比較は  $t[\text{(1)}]$  と  $s[0]$  から始める。なお、 $t[pt+i]$  と  $s[i]$  の比較において、 $i$  は最大で  $\text{(2)}$  である。また、 $pt+i$  が配列  $t$  の最後に格納されている位置を超えてはならないので、 $pt$  は最大で  $\text{(3)}$  である。

(1) の解答群

- |             |             |
|-------------|-------------|
| ア. $pt + 1$ | イ. $pt + 2$ |
| ウ. $pt + 3$ | エ. $pt + 4$ |

(2) , (3) の解答群

- |               |                  |
|---------------|------------------|
| ア. $sLen - 1$ | イ. $sLen - tLen$ |
| ウ. $tLen - 1$ | エ. $tLen - sLen$ |

<設問 2 > 次の流れ図の説明を読み、流れ図中の  $\text{[ ]}$  に入れるべき適切な字句を解答群から選べ。なお、流れ図中の空欄 (3) は設問 1 の正解と同じものが入る。

[流れ図の説明]

配列  $t$  の中に格納された文字列から配列  $s$  に格納された文字列を検索し、検索できた先頭位置を表示する流れ図である。ただし、検索できなかった場合は  $-1$  を表示する。なお、配列  $t$  と配列  $s$  にはあらかじめデータが入力されているものとし、配列  $t$  に格納された文字数は  $tLen$ 、配列  $s$  に格納された文字数は  $sLen$  に格納されている。

(4) の解答群

- |                       |                         |
|-----------------------|-------------------------|
| ア. $sw \leftarrow -1$ | イ. $sw \leftarrow 0$    |
| ウ. $sw \leftarrow 1$  | エ. $sw \leftarrow sLen$ |

(5) の解答群

- |                       |                    |
|-----------------------|--------------------|
| ア. $i \leq sLen - 1$  | イ. $i < sLen + 1$  |
| ウ. $pt \leq tLen - 1$ | エ. $pt < tLen + 1$ |

(6) の解答群

- |                       |                        |
|-----------------------|------------------------|
| ア. $pos \leftarrow 1$ | イ. $pos \leftarrow 0$  |
| ウ. $pos \leftarrow i$ | エ. $pos \leftarrow pt$ |

(7) の解答群

- |                           |                         |
|---------------------------|-------------------------|
| ア. $i \leftarrow i + 1$   | イ. $ps \leftarrow sLen$ |
| ウ. $pt \leftarrow pt + 1$ | エ. $pt \leftarrow tLen$ |

[流れ図]

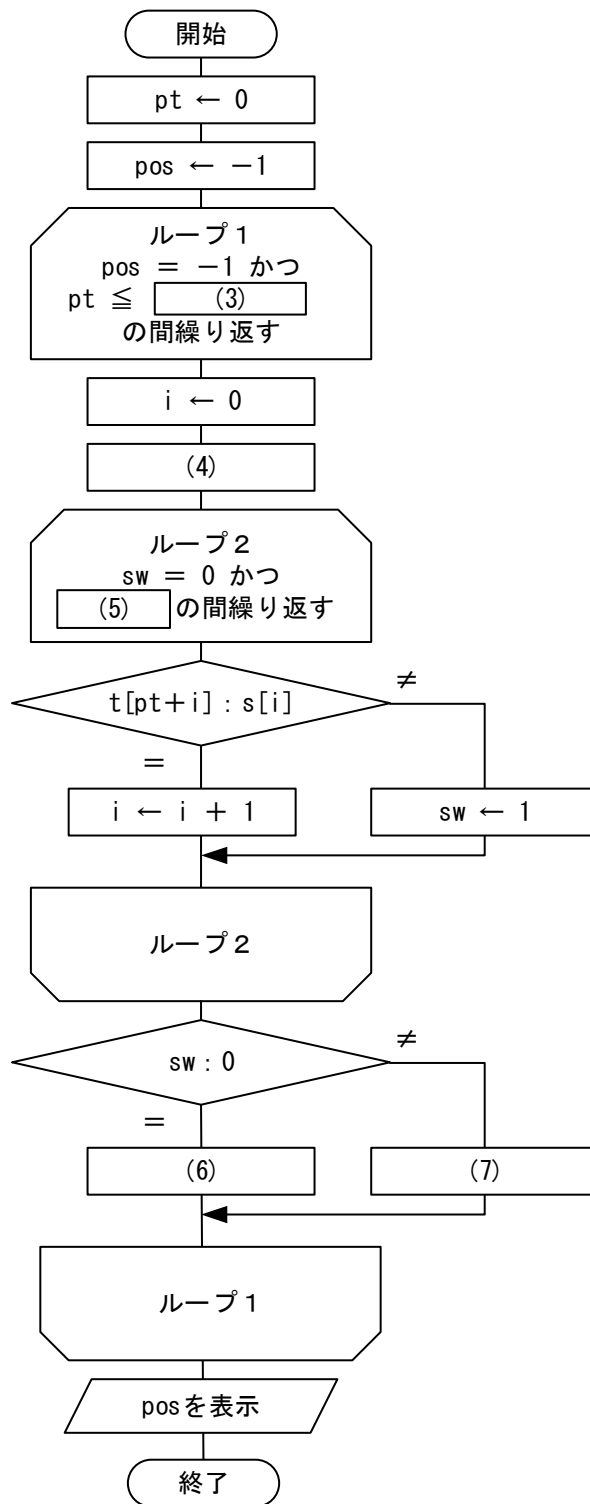


図 4 配列 t から配列 s の文字列を検索する流れ図

[ボイヤー・ムーア法について]

ボイヤー・ムーア法とは、比較する文字の順序を後方から前方にすることで無駄になる比較を行わないようにするものである。なお、ここでは配列 t の文字列から配列 s の文字列を検索するものとする。

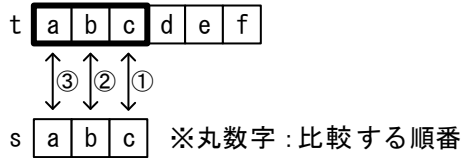


図5 ボイヤー・ムーア法による検索

比較の途中で不一致になった場合は、それ以上の比較を止めて次の比較開始位置へ進むが、次の比較開始位置は比較を開始した時の配列 t の文字が、配列 s に存在するかどうかにより次のように考える。

(ア) 配列 t で比較を開始した位置の文字が配列 s に存在しない

図6では不一致となった時の配列 t の文字は"x"である。"x"は配列 s に存在しないので、次の比較開始位置を配列 s の文字数だけ移動した位置とする。

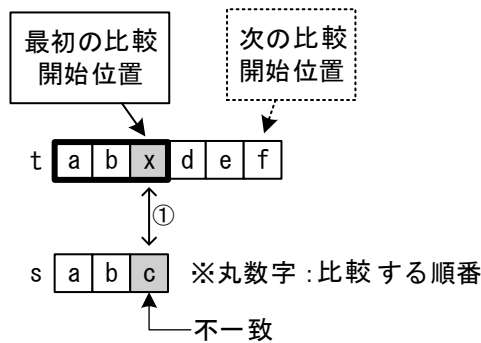


図6 不一致となった文字が配列 s に存在しない場合

(イ) 配列 t で比較を開始した位置の文字が配列 s に存在する

図7では不一致となった時の配列 t の文字は"a"である。"a"は配列 s に存在するので、配列 s と配列 t の文字位置を合わせるように移動量を調整する。

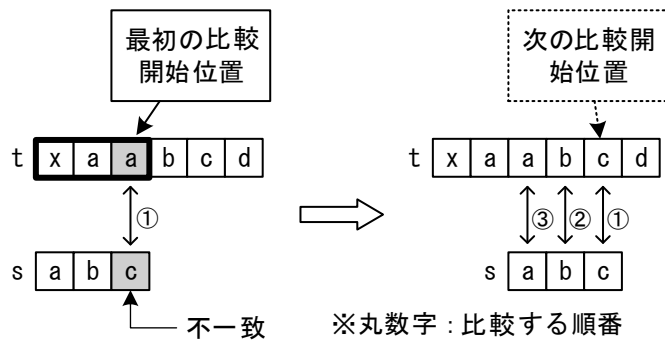


図7 不一致となった文字が配列 s に存在する場合

(イ) の移動量は、配列  $s$  の中での文字の位置によって異なる。ここでは、「配列  $s$  の文字数 - 文字の位置 - 1」として計算する。ただし、文字の位置は 0 から始まる。また、配列  $s$  の末尾の文字は移動量を配列  $s$  の文字数とし、同じ文字が複数存在した場合は一番小さい値とする。

ここで、配列  $s$  の文字の移動量を配列  $m$  の対応する位置に格納するとすれば図 8 のようになる。

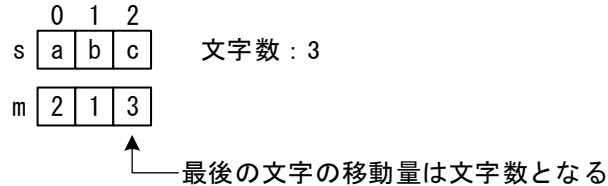


図 8 配列  $s$  と移動量を格納する配列  $m$

<設問 3> 図 9 の  に入れるべき適切な字句を解答群から選べ。

配列  $s$  が "aybabx" とすれば、配列  $m$  に格納される値は次の図 9 のようになる。

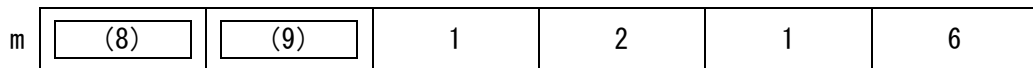


図 9 配列  $m$  の内容

(8) , (9) の解答群

- ア. 2                      イ. 3                      ウ. 4                      エ. 5

<設問 4> 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]

配列  $s$  の文字に対する移動量を配列  $m$  へ求めて返却する移動量計算である。

流れ図中で用いる配列  $s$  には文字列の 1 文字ずつが先頭から順番に設定されており、その文字数は変数  $sLen$  に設定されているものとする。また、配列  $m$  の大きさは配列  $s$  と同じとする。

(10) の解答群

- ア.  $m[sLen - 1] \leftarrow sLen$                       イ.  $m[sLen - 1] \leftarrow sLen - 1$   
 ウ.  $m[sLen] \leftarrow sLen$                               エ.  $m[sLen] \leftarrow sLen - 1$

[流れ図]

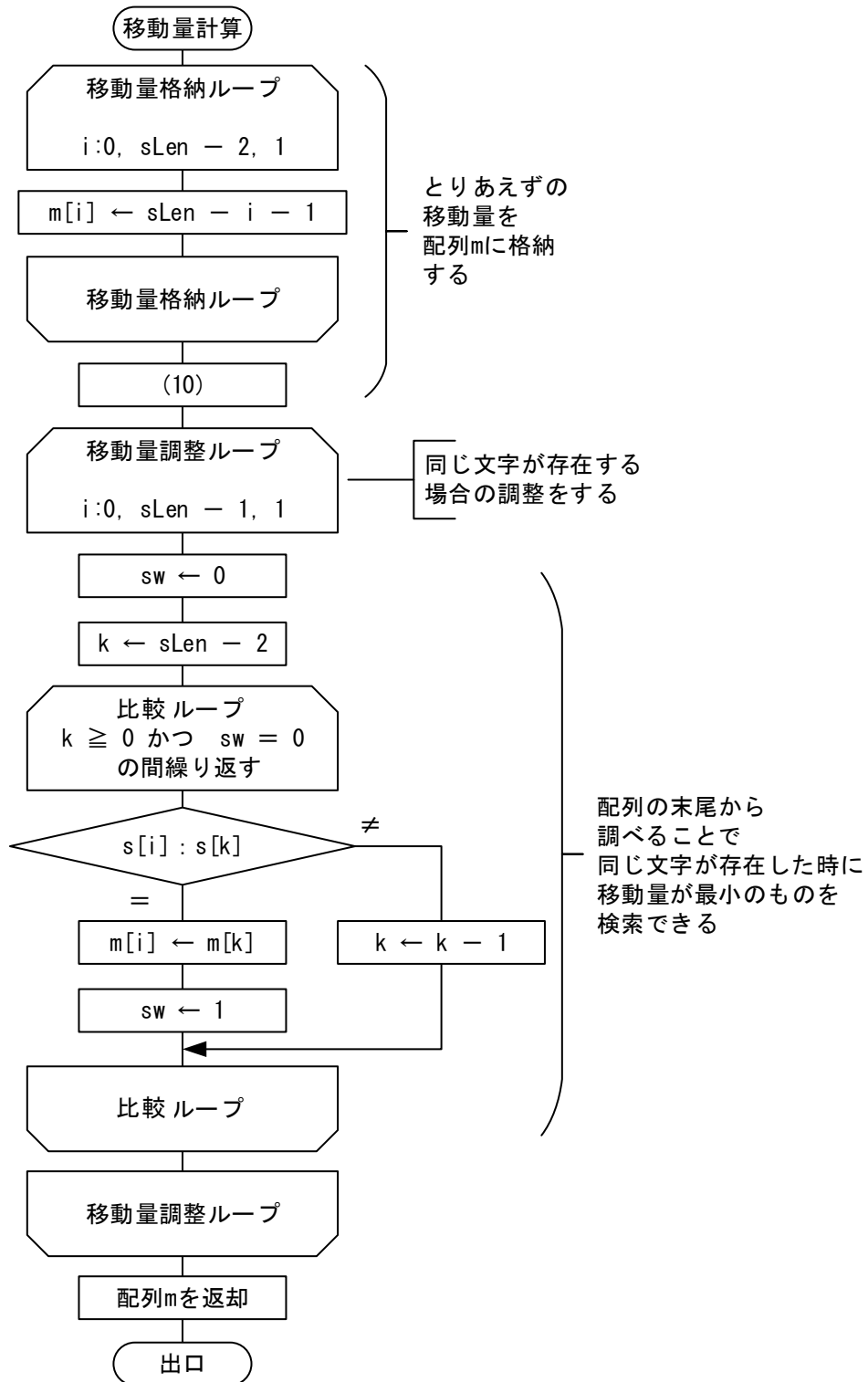


図 10 移動量計算の流れ図

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。

情報の圧縮は、その情報の性質を保ったままデータ量を減らすために、別のデータ形式に変換する処理である。圧縮の方法には様々なものがある。

[プログラムの説明]

英字で構成される文字列の圧縮・展開をランレングス符号化で行うプログラムである。ランレングス符号化とは同じデータ(英字)が連続して出現することに着目して圧縮する方法であり、4文字以上連続している場合に圧縮対象文字と、制御文字(\$), その個数の組み合わせで次のような形式で符号化し、3文字以下の場合、圧縮は行わない。なお、連続する同一データの個数は最大26とし、個数4~26はD~Zで表す。

圧縮対象文字	制御文字(\$)	個数
--------	----------	----

例

- ① v\$Hww : データ v が 8 個連続し、データ w が 2 個連続する
- ② x\$Dy\$G : データ x が 4 個連続し、データ y が 7 個連続する

ここで、圧縮前の文字列は255文字以内であり、圧縮は関数Comp, 展開は関数DeCompで行う。

関数 Comp では、配列 In\_data に格納されている圧縮前の文字列を受け取り、圧縮後の文字列を配列 Out\_data に格納する。なお、In\_len, Out\_len は、それぞれ配列 In\_data と配列 Out\_data に格納されている文字列の長さである。

なお、ここで扱う配列は十分な大きさがあり、未使用の領域には NULL が設定されている。

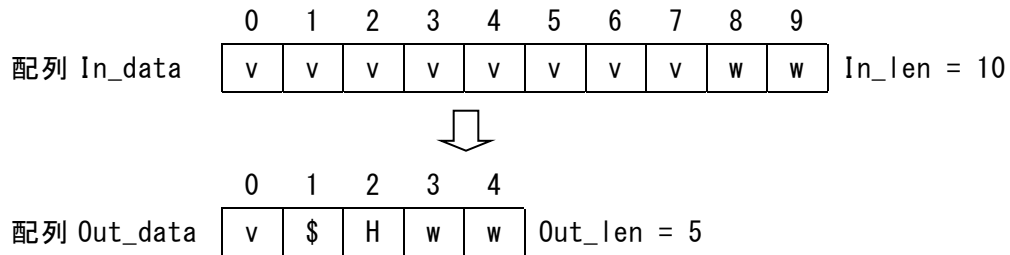


図1 圧縮処理による文字列の変化



関数 DeComp では、配列 Out\_data に格納されている圧縮されている文字列を受け取り、展開後の文字列を配列 In\_data に格納する。

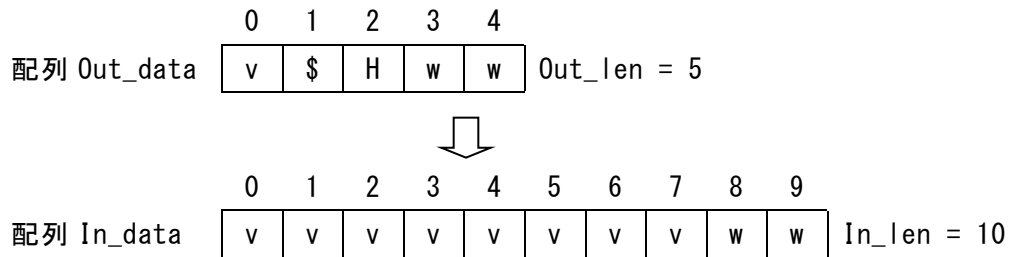


図2 展開処理による文字列の変化

関数 Comp と関数 DeComp で使用している関数 ToAlpha と関数 ToInt の仕様は次の通りである。

[関数 ToAlpha の仕様]

整数 1, 2, ..., 26 を順に英字 A, B, ..., Z に変換する。

引数/返却値	データ型	意味
引数	整数型	整数 1, 2, ..., 26 の値
返却値	文字型	引数に対応した英字

[関数 ToInt の仕様]

英字 A, B, ..., Z を順に整数 1, 2, ..., 26 に変換する。

引数/返却値	データ型	意味
引数	文字型	英字 A, B, ..., Z の文字
返却値	整数型	引数に対応した整数の値

[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 ─── ■	前判定繰り返し処理を示す。 条件式が真の間、処理を実行する。

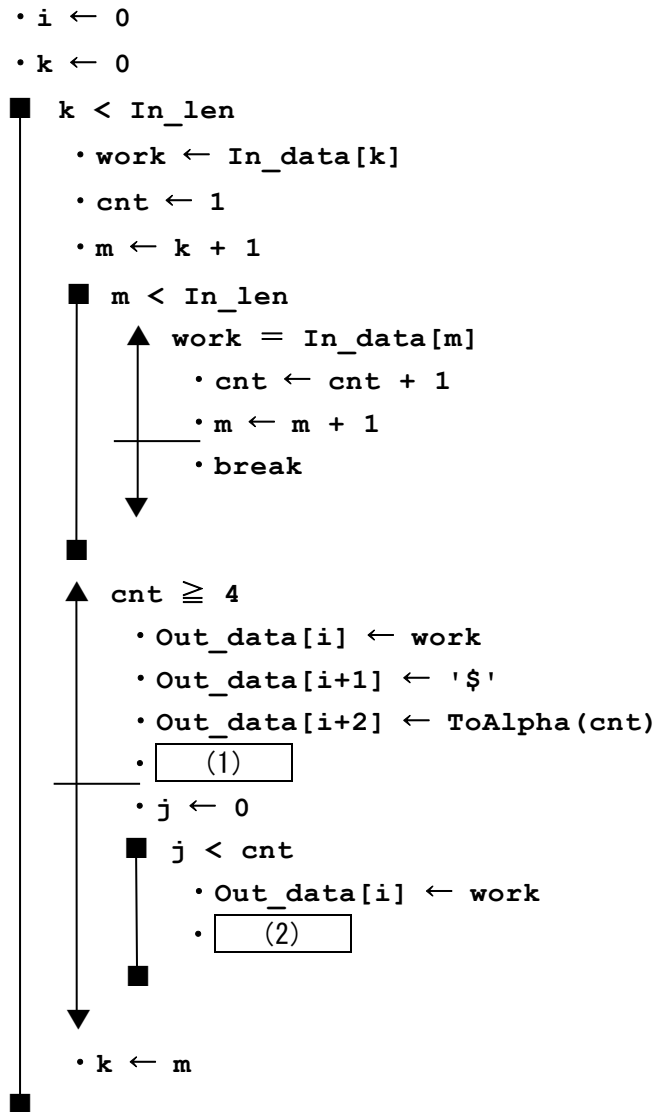
[演算子と優先順位]

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]

○関数 **Comp** (文字型配列 : `In_data[]`, `Out_data[]`, 整数型 : `In_len`)  
 ○整数型 : `cnt`, `i`, `j`, `k`, `m`, `Out_len`;  
 ○文字列型 : `work`;  
 /\* 文字列の圧縮 \*/



```

• Out_len ← i
• return Out_len ← α

```

○関数 DeComp (文字型配列 : In\_data[], Out\_data[], 整数型 : Out\_len)

○整数型 : cnt, i, j, k, In\_len;

○文字列型 : work;

/\* 文字列の展開 \*/

```

• i ← 0

```

```

• k ← 0

```

```

■ k < Out_len - 1

```

```

  ▲ Out_data[k+1] = '$'

```

```

    • cnt ← ToInt(Out_data[k+2])

```

```

    • j ← 0

```

```

    ■ j < cnt

```

```

      • In_data[i] ← Out_data[k]

```

```

      • (2)

```

```

    ■

```

```

    • (3)

```

```

    • In_data[i] ← Out_data[k]

```

```

    • (4)

```

```

■

```

```

▲ k = Out_len - 1

```

```

  • In_data[i] ← Out_data[k]

```

```

• In_len ← i

```

```

• return In_len

```

<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.  $i \leftarrow i + 1$

イ.  $i \leftarrow i + 3$

ウ.  $i \leftarrow i + cnt$

(2) , (4) の解答群

ア.  $i \leftarrow i + 1, j \leftarrow j + 1$

イ.  $i \leftarrow i + 1, k \leftarrow k + 1$

ウ.  $j \leftarrow j + 1, k \leftarrow k + 1$

エ.  $j \leftarrow j + 1, k \leftarrow j + 1$

(3) の解答群

ア.  $k \leftarrow k + 1$

イ.  $k \leftarrow k + 2$

ウ.  $k \leftarrow k + 3$

<設問2> 次のプログラムの結果に関する記述中の  に入れるべき適切な字句を解答群から選べ。

関数 Comp において、次の文字列が In\_data に引き渡された場合、プログラム中の  $\alpha$  部分の Out\_len の値は  (5) となる。

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
a	a	a	b	b	b	b	b	c	c	d	d	d	d	d	d

(5) の解答群

ア. 8

イ. 10

ウ. 11

エ. 15

オ. 16

## < 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   | 20 ページ～24 ページ |
| ・ 表計算の問題   | 26 ページ～30 ページ |
| ・ アセンブラの問題 | 31 ページ～34 ページ |

## 選択問題 C言語の問題

次の宅配料金計算に関する記述を読み、各設問に答えよ。

[宅配料金について]

荷物 1 個あたりの宅配料金は、重量と長さ(以下、大きさ)によって 6 つに分けられた区分と発着地域により決まる。なお、ここで扱う荷物の重量は 25000g 以下、縦・横・高さの三辺を足した長さ(以下、長さ)は 1600mm 以下であるものとし、配達地域は日本国内に限定する。

次の表 1 は、重量と長さの基準を区分ごとに分けたものである。

表 1 大きさの区分

区分	最大重量 (g)	最大長 (mm)
1	2000	600
2	5000	800
3	10000	1000
4	15000	1200
5	20000	1400
6	25000	1600

重量と長さのそれぞれの区分で大きい方の区分が適用される。例えば、重量が 3000g で長さが 1100mm の場合は大きさの区分は 4 になり、重量が 23000g で長さが 500mm の場合は大きさの区分は 6 になる。

発着地域は、日本国内を 12 の地域に分けたものとする。宅配料金は大きさの区分と発着地域により決定する。ここでは 10 種類の発着地域区分を扱う。次の表 2 は、大きさの区分ごとに定めた宅配料金である。

表 2 宅配料金

		発着地域区分									
		1	2	3	4	5	6	7	8	9	10
大 き さ の 区 分	1	900	1000	1100	1200	1400	1500	1600	1700	1800	2000
	2	1100	1200	1300	1500	1600	1700	1800	1900	2000	2600
	3	1300	1500	1600	1700	1800	1900	2000	2200	2300	3200
	4	1600	1700	1800	1900	2000	2100	2300	2400	2500	3700
	5	1800	1900	2000	2200	2300	2400	2500	2600	2700	4300
	6	2000	2100	2300	2400	2500	2600	2700	2800	3000	4800

次の表3は、12の発着地域による発着地域区分を表したものである。例えば、“3”の地域から“6”の地域へ送る場合の発着地域区分は“2”になり、表2から大きさの区分と合わせて料金を決定する。

表3 発着地域による発着地域区分

		発着地域											
		1	2	3	4	5	6	7	8	9	10	11	12
着 地 域	1	1	3	4	5	5	6	6	8	9	9	10	10
	2	3	1	1	2	2	3	3	4	5	5	7	8
	3	4	1	1	1	1	2	2	3	5	5	7	7
	4	5	2	1	1	1	1	1	2	3	3	5	5
	5	5	2	1	1	1	1	1	2	3	3	5	6
	6	6	3	2	1	1	1	1	1	2	2	3	6
	7	6	3	2	1	1	1	1	1	2	2	3	5
	8	8	4	3	2	2	1	1	1	1	1	2	5
	9	9	5	5	3	3	2	2	1	1	1	1	5
	10	9	5	5	3	3	2	2	1	1	1	2	5
	11	10	7	7	5	5	3	3	2	1	2	1	4
	12	10	8	7	5	6	6	5	5	5	5	4	1

<設問1> 次の宅配料金の算出に関する記述中の  に入れるべき適切な字句を解答群から選べ。

次の場合の宅配料金を計算する。

- ・重量 4300g
- ・長さ 1155mm
- ・発着地域 3
- ・着地域 9

表1より、重量での区分は  (1) ，長さでの区分は  (2) になるので、大きさの区分は  (2) である。

表3より、発着地域における発着地域区分は  (3) である。

よって、宅配料金は表2より  (4) 円となる。

(1) ~ (3) の解答群

- |      |      |      |      |       |
|------|------|------|------|-------|
| ア. 1 | イ. 2 | ウ. 3 | エ. 4 | オ. 5  |
| カ. 6 | キ. 7 | ク. 8 | ケ. 9 | コ. 10 |

(4) の解答群

ア. 2000                      イ. 2100                      ウ. 2300                      エ. 2600

<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]

重量と長さを受け取り、大きさの区分を返却する関数 `getKubun` である。

重量と長さの基準は、それぞれ `weight.csv`, `length.csv` という CSV 形式のファイルに図 1 のように記録されている。それぞれ基準値と区分の値を 1 行に記録している。

weight.csv	length.csv
2000, 1	600, 1
5000, 2	800, 2
10000, 3	1000, 3
15000, 4	1200, 4
20000, 5	1400, 5
25000, 6	1600, 6

図 1 `weight.csv` と `length.csv` の内容

[関数の説明]

`getKubun` 関数

引 数: `weight`(重量: 整数値), `length`(長さ: 整数値)

機 能: 重量の区分と長さの区分を求め、大きい方を大きさの区分として返却する。  
同値の場合はどちらを選択しても良い。

戻り値: 大きさの区分の値。ただし、上限値を超えた場合またはファイル入力時に障害があった場合は-1。

`srcKubun` 関数

引 数: `filename`(ファイル名: 文字列), `size`(重量または長さ: 整数値)

機 能: `filename` で与えられた CSV 形式のファイルから基準値と区分の値を読み、  
`size` で与えられた重量または長さに合致する区分を求める。

戻り値: 区分の値。ただし、上限値を超えた場合またはファイル入力時に障害があった場合は-1。



[プログラム]

```
int getKubun(int weight, int length) {
    int kubunW, kubunL, result;
    kubunW = srcKubun("weight.csv", weight); /* 重量の検索 */
    kubunL = srcKubun("length.csv", length); /* 大きさの検索 */
    result = -1;
    if ( (5) ) {
        if ( (6) ) {
            result = kubunL;
        } else {
            result = kubunW;
        }
    }
    return result;
}

int srcKubun(char *filename, int size) {
    FILE *fp;
    int limit, kubun, result;
    result = -1;
    fp = fopen(filename, "rt");
    if ( (7) ) {
        printf("open error: %s\n", filename);
    } else {
        while( fscanf(fp, "%d %d", &limit, &kubun) != EOF) {
            if ( (8) ) {
                result = kubun;
                break;
            }
        }
    }
    fclose(fp);
    return result;
}
```

(5) , (6) の解答群

- |  |  |
|--|--|
| ア. <code>kubunL &gt; kubunW</code>                     | イ. <code>kubunL &gt; 0</code>                  |
| ウ. <code>kubunW &gt; kubunL</code>                     | エ. <code>kubunW &gt; 0</code>                  |
| オ. <code>kubunL &gt; 0 &amp;&amp; kubunW &gt; 0</code> | カ. <code>kubunL &gt; 0    kubunW &gt; 0</code> |
| キ. <code>kubunL == -1 &amp;&amp; kubunW == -1</code>   | ク. <code>kubunL == -1    kubunW == -1</code>   |

(7) の解答群

ア. `fp == -1`

ウ. `fp == EOF`

イ. `fp == '\n'`

エ. `fp == NULL`

(8) の解答群

ア. `limit < size`

ウ. `limit > size`

イ. `limit <= size`

エ. `limit >= size`

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。

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。

AND 関数

論理式のすべての評価が真であれば真、1つでも偽であれば偽を返す。

書式：AND(論理式 1, 論理式 2, …)

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)

IFERROR 関数

値にエラーが無ければ値を返し、そうでなければエラーの場合の値を返す

書式：IFERROR(値, エラーの場合の値)

ISERROR 関数

値がエラーのときに TRUE を返し、そうでなければ FALSE を返す。

書式：ISERROR(値)

MONTH 関数

日付を表すシリアル値から月の値(1~12)を返す。

書式：MONTH(シリアル値)

OR 関数

論理式の評価が1つでも真であれば真、すべて偽であれば偽を返す。

書式：OR(論理式 1, 論理式 2, …)

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、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)

YEAR 関数

日付を表すシリアル値から西暦年を表す 4 桁の整数(1900～9999)を返す。

書式：YEAR(シリアル値)

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 社では、表計算ソフトを使用して社員の健康診断情報を管理している。社員は 30 名おり、「社員表」ワークシートの 2 行目から 31 行目にまとめられている。本年度健康診断の結果が診療機関から CSV 形式で送られてきたため、ダウンロードし「本年度健診結果」ワークシートの 2 行目から 26 行目に入力した。なお、身長単位は cm、体重単位は kg である。また、各検査結果の判定の見方は「判定表」ワークシートに入力した。

	A	B	C	D	E	F
1	社員番号	氏名	性別	生年月日	入社年月日	所属ID
2	12096	門田正二	男	1969/06/03	1991/04/01	E01
3	12111	塩見清佳	女	1971/01/28	1993/10/01	E01
4	12886	佐伯三平	男	1973/07/17	1995/07/01	E01
5	13128	矢吹隆明	男	1975/09/24	1997/10/01	E01
6	13438	高岡千咲	女	1985/01/24	2007/04/01	S01
:	:	:	:	:	:	:
31	22382	毛利文隆	男	1968/08/31	1990/09/01	E01

図 1 「社員表」ワークシート

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	受診年月日	社員番号	性別	身長	体重	血圧	血中脂質	肝機能	血糖	尿	貧血	心電図検査	眼底検査
2	2020/06/01	15084	女	159.9	49.9	A	A	A	A	A	F	A	A
3	2020/06/01	19057	男	173.0	58.1	A	A	A	A	A	A	A	A
4	2020/06/02	15651	女	155.8	55.9	A	A	A	A	A	A	A	A
5	2020/06/02	16231	女	155.8	48.7	A	A	A	A	A	A	A	A
6	2020/06/02	18434	女	156.5	50.6	A	A	A	A	C	A	A	A
:	:	:	:	:	:	:	:	:	:	:	:	:	:
26	2020/11/13	21398	男	173.1	96.5	A	E	A	E	A	A	A	A

図 2 「本年度健診結果」ワークシート

	A	B
1	判定	所見
2	A	なし
3	B	わずかに異常を認めるが、日常生活に差し支えなし
4	F	継続して治療を受ける必要あり
5	C	日常生活に注意し、経過観察が必要あり
6	E	再検査が必要あり

図3 「判定表」ワークシート

<設問1> 次の「本年度健診結果」ワークシートの拡張に関する記述中の  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	H	I	J	K	L	M	N	O	P	Q
1	社員番号	性別	生年月日	年齢	受診年月日	身長	体重	BMI	血圧	血中脂質	肝機能	血糖	尿	貧血	心電図検査	眼底検査	再検査数
2	12096	男	1969/06/03	51	2020/11/11	172.1	61.5	20.8	A	A	A	A	A	A	A	A	0
3	12111	女	1971/01/28	50	2020/06/03	153.9	75.3	31.8	A	A	A	A	A	A	F	A	0
4	13128	男	1975/09/24	45	2020/06/05	174.2	60.8	20.0	A	A	A	A	E	A	A	A	1
5	13438	女	1985/01/24	36	2020/06/03	156.8	48.4	19.7	A	A	A	A	A	A	A	A	0
6	14023	女	1980/05/05	40	2020/06/03	161.7	57.0	21.8	A	B	A	A	A	A	A	A	0
:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
26	22382	男	1968/08/31	52	2020/11/12	176.2	74.8	24.1	A	E	A	A	A	A	A	A	1

図4 拡張した「本年度健診結果」ワークシート

- ・A列に社員番号を、E列に受診年月日を移動し、社員番号の昇順に並べ替えた。
- ・C列の生年月日は、「社員表」ワークシートから社員番号で検索をして表示する。セルC2に次の式を入力し、セルC3～C26まで複写した。

=  (1)

- ・D列は年齢を表示する。なお、年齢は当該年度(4月1日～翌年3月31日)で求めるため、受診年月日が誕生日前だとしても誕生日を迎えたものとする。セルD2に次の式を入力し、セルD3～D26まで複写した。

= IF( (2), YEAR(E2) - YEAR(C2) + 1,  
IF( (3), YEAR(E2) - YEAR(C2) - 1, YEAR(E2) - YEAR(C2)))

- ・H列はBMIを表示する。BMIは肥満度を表す数値であり、体重(kg)÷身長(m)÷身長(m)で計算をする。セルH2に次の式を入力し、セルH3～H26まで複写した。

=  (4)

- ・Q列は再検査数を表示する。セルQ2に次の式を入力し、セルQ3～Q26まで複写した。

=  (5)

(1) の解答群

- ア. VLOOKUP(A\$2, 社員表!\$A2:\$D31, 4, 0)
- イ. VLOOKUP(A\$2, 社員表!A2:D31, 4, 0)
- ウ. VLOOKUP(A2, 社員表!A2:D31, 4, 0)
- エ. VLOOKUP(A2, 社員表!A\$2:D\$31, 4, 0)

(2) , (3) の解答群

- ア. AND(MONTH(C2) > 3, MONTH(E2) > 3)
- イ. AND(MONTH(C2) > 3, MONTH(E2) < 4)
- ウ. AND(MONTH(C2) < 4, MONTH(E2) > 3)
- エ. OR(MONTH(C2) > 3, MONTH(E2) > 3)
- オ. OR(MONTH(C2) > 3, MONTH(E2) < 4)
- カ. OR(MONTH(C2) < 4, MONTH(E2) > 3)

(4) の解答群

- ア. G2 / F2 / F2
- イ. G2 / F2 / F2 \* 100
- ウ. G2 / F2 \* 100 / F2 \* 100
- エ. G2 / F2 \* 100 / F2

(5) の解答群

- ア. COUNTIF(\$I2:P\$2, "E")
- イ. COUNTIF(I\$2:\$P2, "E")
- ウ. COUNTIF(I\$2:P\$2, "E")
- エ. COUNTIF(I2:P2, "E")

<設問 2 > 次の「健康診断確認表」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
1	社員番号	氏名	所属ID	受診	再検査数	BMI (25以上)
2	12096	門田正二	E01		0	
3	12111	塩見清佳	E01		0	注意
4	12886	佐伯三平	E01	未受診		
5	13128	矢吹隆明	E01		1	
6	13438	高岡千咲	S01		0	
:	:	:	:	:	:	:
31	22382	毛利文隆	E01		1	

図 5 「健康診断確認表」ワークシート

- ・セル A1~F1 に項目名を入力し、「社員表」ワークシートから A 列に社員番号, B 列に氏名, C 列に所属 ID を複写した。

- ・ D 列は今年度の健康診断を受けていない社員の場合，“未受診”と表示する。セル D2 に次の式を入力し，セル D3～D31 まで複写した。

= IF (  , "未受診", "" )

- ・ E 列は再検査数を表示する。セル E2 に次の式を入力し，セル E3～E31 まで複写した。

=

- ・ F 列は再検査の必要はないが，BMI が 25 以上の社員の場合，“注意”と表示する。セル F2 に次の式を入力し，セル F3～F31 まで複写した。

= IF (  , "" ,  , "注意", "" )

#### (6) の解答群

- ア． ISERROR (VLOOKUP (A\$2, 本年度検診結果!A2:A26, 1, 0))
- イ． ISERROR (VLOOKUP (A2, 本年度検診結果!A\$2:A\$26, 1, 0))
- ウ． VLOOKUP (ISERROR (A\$2, 本年度検診結果!A2:A26, 1, 0))
- エ． VLOOKUP (ISERROR (A2, 本年度検診結果!A\$2:A\$26, 1, 0))

#### (7) の解答群

- ア． IFERROR (VLOOKUP (A2, 本年度検診結果!A\$2:Q\$26, 17, 0), "")
- イ． IFERROR (VLOOKUP (A2, 本年度検診結果!A2:Q26, 17, 0), "")
- ウ． VLOOKUP (IFERROR (A2, 本年度検診結果!A\$2:Q\$26, 17, 0), "")
- エ． VLOOKUP (IFERROR (A2, 本年度検診結果!A2:Q26, 17, 0), "")

#### (8) の解答群

- ア． E\$2 = ""
- イ． E2 = ""
- ウ． E\$2 <> ""
- エ． E2 <> ""

#### (9) の解答群

- ア． IF (AND (E2 = 0, VLOOKUP (A2, 本年度検診結果!A\$2:H\$26, 8, 0) >= 25)
- イ． IF (AND (E2 <> 0, VLOOKUP (A2, 本年度検診結果!A\$2:H\$26, 8, 0) >= 25)
- ウ． IF (OR (E2 = 0, VLOOKUP (A2, 本年度検診結果!A\$2:H\$26, 8, 0) >= 25)
- エ． IF (OR (E2 <> 0, VLOOKUP (A2, 本年度検診結果!A\$2:H\$26, 8, 0) >= 25)



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]

TBL+0 番地～TBL+9 番地に格納されているデータを、ルールに従って HOME+0 番地～HOME+4 番地または SYNO+1 番地～SYNO+9 番地の対応する番地に格納するプログラム MOD である。TBL 番地以降のデータは 1～99 の値であり、HOME 番地以降および SYNO+1 番地以降は図 1 の形式で、データを下位 8 ビットに格納する。上位 8 ビットはポインタで SYNO 番地以降の相対番地を示している。SYNO 番地には SYNO+1 番地以降に格納されたデータの個数を示す。なお、SYNO 番地の初期値は 0 で、それ以外の領域 (HOME 番地～HOME+4 番地と SYNO+1 番地～SYNO+9 番地) の初期値は -1 とする。

ポインタ (8 ビット)	データ (8 ビット)
-----------------	----------------

図 1 1 語の形式

[格納のルール]

- ① データを 5 で割った余りを  $m$  とする。
  - ・ HOME+ $m$  番地 = -1 のとき、HOME+ $m$  番地にデータを格納する。
  - ・ HOME+ $m$  番地  $\neq$  -1 のとき、②、③を実行する。
- ② SYNO 番地の値に 1 を加えた値を  $n$  とし、SYNO+ $n$  番地にデータを、SYNO 番地に  $n$  を格納する。
- ③ HOME+ $m$  番地のポインタを  $p$  に代入する。
  - ・  $p=0$  のとき、HOME+ $m$  番地のポインタに  $n$  を格納する。
  - ・  $p \neq 0$  のとき、SYNO+ $p$  番地のポインタを新たな  $p$  として SYNO+ $p$  番地のポインタが 0 になるまでたどって行き、ポインタが 0 になった番地のポインタに  $n$  を格納する。

<設問 1 > 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。

格納のルールに従い、1次元配列 TBL の先頭から TBL+4 番地の 31 までを格納した時点について図 2 に示す。なお、TBL 番地以降のデータと、ポインタとデータに分割された HOME 番地以降 SYN0 番地以降の数値は 10 進数で表示している。

TBL+0 番地	13	HOME+0 番地	- 1		SYNO+0 番地	3	
+1 番地	26	+1 番地	1	26	+1 番地	3	11
+2 番地	11	+2 番地	- 1		+2 番地	0	8
+3 番地	8	+3 番地	2	13	+3 番地	0	31
+4 番地	31	+4 番地	- 1		+4 番地	- 1	
+5 番地	18				+5 番地	- 1	
+6 番地	7				+6 番地	- 1	
+7 番地	16				+7 番地	- 1	
+8 番地	12				+8 番地	- 1	
+9 番地	19				+9 番地	- 1	

図 2 1次元配列 TBL の先頭から TBL+4 番地までを処理したときの各配列の内容

図 2 の状態から TBL+5 番地以降も処理すると、TBL+5 番地のデータ 18 は  (1) 番地に、TBL+6 番地のデータ 7 は  (2) 番地に、TBL+7 番地のデータ 16 は  (3) 番地に格納される。

(1) ~ (3) の解答群

- |           |           |           |
|-----------|-----------|-----------|
| ア. HOME+0 | イ. HOME+2 | ウ. HOME+4 |
| エ. SYN0+4 | オ. SYN0+5 | カ. SYN0+6 |

<設問 2 > 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	MOD	START		
110		RPUSH		
120		LAD	GR1, 0	
130	LOOP1	LD	GR2, TBL, GR1	
140	LOOP2	SUBA	GR2, =5	; 繰り返しによる除算
150		JPL	LOOP2	
160		JZE	NEXT1	
170			<input type="text"/> (4)	; m を求める

180	<b>NEXT1</b>	<b>LD</b>	<b>GR3 , TBL , GR1</b>	;	データを GR3 に設定
190		<b>LD</b>	<b>GR0 , HOME , GR2</b>		
200		<b>CPA</b>	<b>GR0 , =-1</b>		
210		<b>JNZ</b>	<b>NEXT2</b>		
220			(5)	;	HOME 番地以降に格納
230		<b>JUMP</b>	<b>NEXT5</b>		
240	<b>NEXT2</b>	<b>LD</b>	<b>GR4 , SYNO</b>		
250		<b>LAD</b>	<b>GR4 , 1 , GR4</b>	;	n を GR4 に設定
260		<b>ST</b>	<b>GR4 , SYNO</b>		
270		<b>ST</b>	<b>GR3 , SYNO , GR4</b>		
280		<b>LD</b>	<b>GR5 , HOME , GR2</b>		
290		<b>SRL</b>	<b>GR5 , 8</b>	;	ポインタの取り出し
300		<b>JNZ</b>	<b>NEXT3</b>		
310			(6)		
320		<b>OR</b>	<b>GR4 , HOME , GR2</b>	;	ポインタとデータを合成
330		<b>ST</b>	<b>GR4 , HOME , GR2</b>		
340		<b>JUMP</b>	<b>NEXT5</b>		
350	<b>NEXT3</b>	<b>LD</b>	<b>GR6 , SYNO , GR5</b>		
360		<b>SRL</b>	<b>GR6 , 8</b>		
370		<b>JZE</b>	<b>NEXT4</b>		
380		<b>LD</b>	<b>GR5 , GR6</b>	;	新ポインタの設定
390		<b>JUMP</b>	<b>NEXT3</b>		
400	<b>NEXT4</b>	<b>SLL</b>	<b>GR4 , 8</b>		
410			(7)		
420		<b>ST</b>	<b>GR4 , SYNO , GR5</b>		
430	<b>NEXT5</b>	<b>LAD</b>	<b>GR1 , 1 , GR1</b>	;	次のデータの準備
440		<b>CPA</b>	<b>GR1 , =10</b>		
450		<b>JMI</b>	<b>LOOP1</b>		
460		<b>RPOP</b>			
470		<b>RET</b>			
480	<b>TBL</b>	<b>DS</b>	<b>10</b>		
490	<b>HOME</b>	<b>DS</b>	<b>5</b>		
500	<b>SYNO</b>	<b>DS</b>	<b>10</b>		
510		<b>END</b>			

(4) の解答群

ア. `ADDA GR2,=1`

ウ. `SLL GR2,1`

イ. `ADDA GR2,=5`

エ. `SLL GR2,2`

(5) の解答群

ア. `ST GR0,HOME,GR1`

ウ. `ST GR3,HOME,GR1`

イ. `ST GR0,HOME,GR2`

エ. `ST GR3,HOME,GR2`

(6) の解答群

ア. `SLL GR2,8`

ウ. `SLL GR4,8`

イ. `SLL GR3,8`

エ. `SLL GR5,8`

(7) の解答群

ア. `OR GR3,SYNO,GR2`

ウ. `OR GR4,SYNO,GR2`

イ. `OR GR3,SYNO,GR5`

エ. `OR GR4,SYNO,GR5`

<メモ欄>

<メモ欄>

<メモ欄>

