

令和8年度  
基本情報技術者試験 科目 B  
公開問題

|      |       |
|------|-------|
| 問題番号 | 問1～問6 |
| 選択方法 | 全問必須  |

注意事項

1. 実際の試験は20問で構成されますが、そのうちの6問を公開しています。
2. 問題に関する質問にはお答えできません。文意どおり解釈してください。

## 擬似言語の記述形式（基本情報技術者試験用）

擬似言語を使用した問題では、各問題文中に注記がない限り、次の記述形式が適用されているものとする。

〔擬似言語の記述形式〕

| 記述形式   | 説明   |
|--|--|
| ○ <u>手続名又は関数名</u>  | 手続又は関数を宣言する。   |
| <u>型名</u> : <u>変数名</u>   | 変数を宣言する。   |
| /* <u>注釈</u> */  | 注釈を記述する。   |
| // <u>注釈</u>   |  |
| <u>変数名</u> ← <u>式</u>  | 変数に <u>式</u> の値を代入する。  |
| <u>手続名又は関数名</u> ( <u>引数</u> , …)   | 手続又は関数を呼び出し、 <u>引数</u> を受け渡す。  |
| if ( <u>条件式 1</u> )<br><u>処理 1</u><br>elseif ( <u>条件式 2</u> )<br><u>処理 2</u><br>elseif ( <u>条件式 n</u> )<br><u>処理 n</u><br>else<br><u>処理 n + 1</u><br>endif | <p>選択処理を示す。</p> <p><u>条件式</u>を上から評価し、最初に真になった<u>条件式</u>に対応する<u>処理</u>を実行する。以降の<u>条件式</u>は評価せず、対応する<u>処理</u>も実行しない。どの<u>条件式</u>も真にならないときは、<u>処理 n + 1</u>を実行する。</p> <p>各<u>処理</u>は、0以上の文の集まりである。</p> <p>elseif と<u>処理</u>の組みは、複数記述することがあり、省略することもある。</p> <p>else と<u>処理 n + 1</u>の組みは一つだけ記述し、省略することもある。</p> |
| while ( <u>条件式</u> )<br><u>処理</u><br>endwhile  | <p>前判定繰返し処理を示す。</p> <p><u>条件式</u>が真の間、<u>処理</u>を繰返し実行する。</p> <p><u>処理</u>は、0以上の文の集まりである。</p>   |
| do<br><u>処理</u><br>while ( <u>条件式</u> )  | <p>後判定繰返し処理を示す。</p> <p><u>処理</u>を実行し、<u>条件式</u>が真の間、<u>処理</u>を繰返し実行する。</p> <p><u>処理</u>は、0以上の文の集まりである。</p>   |
| for ( <u>制御記述</u> )<br><u>処理</u><br>endfor   | <p>繰返し処理を示す。</p> <p><u>制御記述</u>の内容に基づいて、<u>処理</u>を繰返し実行する。</p> <p><u>処理</u>は、0以上の文の集まりである。</p>   |

[演算子と優先順位]

| 演算子の種類 |     | 演算子         | 優先度    |
|--------|-----|-------------|--------|
| 式      |     | () .        | 高      |
| 単項演算子  |     | not + -     | ↑<br>↓ |
| 二項演算子  | 乗除  | mod × ÷     |        |
|        | 加減  | + -         |        |
|        | 関係  | ≠ ≤ ≥ < = > |        |
|        | 論理積 | and         |        |
|        | 論理和 | or          | 低      |

注記 演算子 . は、メンバ変数又はメソッドのアクセスを表す。

演算子 mod は、剰余算を表す。

[論理型の定数]

true, false

[配列]

配列の要素は、“[”と“]”の間にアクセス対象要素の要素番号を指定することでアクセスする。なお、二次元配列の要素番号は、行番号、列番号の順に“,”で区切って指定する。

“{”は配列の内容の始まりを、“}”は配列の内容の終わりを表す。ただし、二次元配列において、内側の“{”と“}”に囲まれた部分は、1行分の内容を表す。

[未定義、未定義の値]

変数に値が格納されていない状態を、“未定義”という。変数に“未定義の値”を代入すると、その変数は未定義になる。

問1 次のプログラム中の  に入れる正しい答えを、解答群の中から選べ。ここで、配列の要素番号は1から始まる。

次のプログラムは、整数型の配列 data の末尾の要素の値を先頭の要素に移動する。この際、末尾以外の各要素の値は、一つずつ後ろの要素に移動する。

[プログラム]

```
整数型の配列: data ← {1, 2, 3, 4, 5, 6, 7, 8, 9}
整数型: top, i
整数型: len ← dataの要素数
top ← data[len]
for (i を )
  data[i] ← data[i - 1]
endfor
data[1] ← top
```

解答群

- ア 2 から len - 1 まで 1 ずつ増やす
- イ 2 から len まで 1 ずつ増やす
- ウ len - 1 から 2 まで 1 ずつ減らす
- エ len から 2 まで 1 ずつ減らす

問2 次のプログラム中の  に入れる正しい答えを，解答群の中から選べ。

関数 complement は，引数として渡された 8 ビット型の値  $x$  について， $x$  に加算すると 00000000 になる値を返す。8 ビット型の加算は，値を符号なし 2 進数とみなしたときの加算とし，桁あふれが発生したときのあふれた桁は無視する。演算子  $\wedge$ ， $\vee$ ， $\nabla$  は，それぞれビット単位の論理積，論理和，排他的論理和を表す。

[プログラム]

```
○8ビット型: complement(8ビット型: x)
  8ビット型: y
  y ← 
  y ← y + 00000001
  return y
```

解答群

ア  $x \wedge 01111111$

イ  $x \wedge 11111111$

ウ  $x \vee 01111111$

エ  $x \vee 11111111$

オ  $x \nabla 01111111$

カ  $x \nabla 11111111$

問3 次のプログラム中の  に入れる正しい答えを、解答群の中から選べ。ここで、配列の要素番号は1から始まる。

関数 func1 に与える引数と、関数 func2 に与える引数とが同じとき、二つの関数は同じ値を返す。プログラムでは、配列の領域外を参照してはならないものとする。

[プログラム]

```
○整数型: func1(整数型: n)
  if (nが2以下)
    return 1
  endif
  return 2 × func1(n - 2) + func1(n - 1)

○整数型: func2(整数型: n)
  整数型の配列: data ← {1, 1, 1}
  整数型: i

  /* nが3より小さいときは繰返し処理を実行しない */
  for (iを3からnまで1ずつ増やす)
    data[1] ← data[2]
    data[2] ← data[3]
    data[3] ← 
  endfor
  return data[3]
```

解答群

- ア  $2 \times \text{data}[1] + \text{data}[2]$
- イ  $2 \times \text{data}[2] + \text{data}[1]$
- ウ  $2 \times \text{data}[i - 1] + \text{data}[i - 2]$
- エ  $2 \times \text{data}[i - 2] + \text{data}[i - 1]$
- オ  $\text{data}[3] + 2 \times \text{data}[1] + \text{data}[2]$
- カ  $\text{data}[3] + 2 \times \text{data}[2] + \text{data}[1]$
- キ  $\text{data}[3] + 2 \times \text{data}[i - 1] + \text{data}[i - 2]$
- ク  $\text{data}[3] + 2 \times \text{data}[i - 2] + \text{data}[i - 1]$

[ メモ用紙 ]

問4 次のプログラム中の  と  に入れる正しい答えの組合せを、解答群の中から選べ。ここで、配列の要素番号は1から始まる。

単方向リストを、配列 `dataList` と配列 `pointerList` の二つの配列で表現する。`dataList` にリストの要素の値を格納し、`pointerList` にリストの次の要素に対応する `dataList` の要素番号を格納する。単方向リストの先頭は、`dataList[1]` 及び `pointerList[1]` の組みである。単方向リストの末尾に対応する `pointerList` の要素は未定義である。`dataList` のうち単方向リストの要素の値を格納していない要素と、対応する `pointerList` の要素は未定義である。

プログラムが扱う `dataList` 及び `pointerList` の内容を図1に示す。先頭の次の要素の要素番号は、`pointerList[1]` に格納された3であり、値は `dataList[3]` に格納された20である。その次の要素の要素番号は `pointerList[3]` に格納された2であり、値は `dataList[2]` に格納された30である。

|                       |    |    |    |    |   |
|-----------------------|----|----|----|----|---|
| 要素番号                  | 1  | 2  | 3  | 4  | 5 |
| <code>dataList</code> | 10 | 30 | 20 | 40 |   |

|                          |   |   |   |   |   |
|--------------------------|---|---|---|---|---|
| 要素番号                     | 1 | 2 | 3 | 4 | 5 |
| <code>pointerList</code> | 3 | 4 | 2 |   |   |

注記 網掛けはその要素が未定義であることを示す。

図1 `dataList` 及び `pointerList` の内容

関数 `orderList` は、図1の `dataList` 及び `pointerList` で表現した単方向リストの値を、単方向リストの先頭からたどって順番に格納した配列を返す。関数 `orderList` が返す配列を図2に示す。

|      |    |    |    |    |
|------|----|----|----|----|
| 要素番号 | 1  | 2  | 3  | 4  |
|      | 10 | 20 | 30 | 40 |

図2 関数 `orderList` が返す配列

[プログラム]

大域: 整数型の配列: dataList ← {10, 30, 20, 40, 未定義の値}

大域: 整数型の配列: pointerList ← {3, 4, 2, 未定義の値, 未定義の値}

○整数型の配列: orderList()

整数型: i, p ← 1

整数型の配列: linearList ← {} // 要素数0の配列

for (i を 1 から dataListの要素数 まで 1 ずつ増やす)

linearListの末尾 に dataList[p]の値 を追加する

if (  が 未定義)

繰返し処理を終了する

endif

p ←

endfor

return linearList

解答群

|   | a              | b              |
|---|----------------|----------------|
| ア | dataList[p]    | i              |
| イ | dataList[p]    | pointerList[p] |
| ウ | pointerList[p] | i              |
| エ | pointerList[p] | pointerList[p] |

問5 次のプログラム中の  と  に入れる正しい答えの組合せを、解答群の中から選べ。ここで、配列の要素番号は1から始まる。

一つの要素だけが1で他の要素が0であるような整数型の配列による表現を、本問では One-Hot 表現という。関数 `oneHotEncoding` は、図のように、色の名前が格納されている要素数1以上の文字列型の配列を引数として受け取り、配列に含まれる色の名前（図の例では3種類）に基づいて、各要素を One-Hot 表現に変換し、整数型配列の配列に格納して返す。関数 `oneHotEncoding` における変換の例を図に示す。

引数の例： {"Red", "Green", "Blue", "Red"}

↓

"Red"の One-Hot 表現 : {1, 0, 0}

"Green"の One-Hot 表現 : {0, 1, 0}

"Blue"の One-Hot 表現 : {0, 0, 1}

↓

引数の例に対応する戻り値 : {{1, 0, 0}, {0, 1, 0}, {0, 0, 1}, {1, 0, 0}}

図 関数 `oneHotEncoding` における変換の例

[プログラム]

```

○整数型配列の配列: oneHotEncoding(文字列型の配列: colors)
  整数型: i, j, k
  文字列型の配列: colorVector ← {} // 要素数0の配列
  整数型の配列: tempVector
  整数型配列の配列: oneHotVector ← {} // 要素数0の配列

/* 名前一覧の作成 */
for (i を 1 から colorsの要素数 まで 1 ずつ増やす)
  if (colorVector の要素のいずれにも colors[i]の値 が格納されていない)
    colorVectorの末尾 に a を追加する
  endif
endfor

/* One-Hot表現への変換 */
for (j を 1 から colorsの要素数 まで 1 ずつ増やす)
  tempVector ← {} // 要素数0の配列
  for (k を 1 から colorVectorの要素数 まで 1 ずつ増やす)
    if ( b )
      tempVectorの末尾 に 1 を追加する
    else
      tempVectorの末尾 に 0 を追加する
    endif
  endfor
  oneHotVectorの末尾 に tempVector を追加する
endfor
return oneHotVector

```

解答群

|   | a           | b  |
|---|-------------|--|
| ア | colors[i]の値 | colors の要素のいずれかに colorVector[k]の値 が格納されている |
| イ | colors[i]の値 | colors[j] が colorVector[k] と等しい            |
| ウ | 未定義の値       | colors の要素のいずれかに colorVector[k]の値 が格納されている |
| エ | 未定義の値       | colors[j] が colorVector[k] と等しい            |

問6 A社は、従業員1,000名の広告代理店である。A社では、各部署がA社の社内ネットワークに設置したファイルサーバを、設置した部署の運用担当者が管理している。各部署で新たなクラウドサービスを利用する場合、各部署の責任者の承認のもと契約し、各部署の運用担当者が管理している。

営業部では、営業部がA社の社内ネットワークに設置したファイルサーバ（以下、Yサーバという）及び顧客管理のためのクラウドサービス（以下、Zサービスという）を利用している。営業部の各従業員にはYサーバの一般利用者アカウントが割り当てられ、Yサーバ上の全てのファイルが編集可能である。また、営業部の各従業員には、Zサービスの一般利用者アカウントが割り当てられ、営業部の運用担当者には、YサーバとZサービスの管理者アカウントが割り当てられている。

情報セキュリティ部門では、クラウドサービスに関連した情報セキュリティインシデントの発生に備えて、各部署で利用しているクラウドサービスのログ管理に関するルールを図1のとおり整備した。

|  |
|--|
| <p>1. ログの取得<br/>次のログを取得すること</p> <ul style="list-style-type: none"><li>・ログイン及びログアウトのログ（以下、ログインログという）</li><li>・クラウドサービスでの重要な操作及びその成否に関するログ</li></ul> <p>2. ログの項目<br/>少なくとも次が記録されること</p> <ul style="list-style-type: none"><li>・対象のアカウント名、日本標準時での日時、操作内容</li></ul> <p>3. ログのアクセス管理</p> <ul style="list-style-type: none"><li>・運用担当者だけがログにアクセスできるようにすること</li><li>・クラウドサービスからログをエクスポートして保管する場合は、社内ネットワークに設置した自部署のファイルサーバにログを保管し、運用担当者だけがアクセスできるようにすること</li><li>・運用担当者は複数名にすること</li><li>・クラウドサービスへの運用担当者のログインには、2要素認証を必要とすること</li></ul> <p>4. ログの保存期間</p> <ul style="list-style-type: none"><li>・少なくとも過去1年間のログが参照できるようにすること</li></ul> <p>5. 改ざんへの対策</p> <ul style="list-style-type: none"><li>・ログは運用担当者でも書込み及び消去ができないようにすること</li></ul> |
|--|

図1 各部署で利用しているクラウドサービスのログ管理に関するルール

営業部の情報セキュリティリーダーであるB課長は、運用担当者のC主任及び情報セキュリティ部門の協力を得て、図1のルールが順守されているかどうかを調査した。

B課長は、Zサービスでの現在の運用の中で、図1のルールに違反しているものを表1のとおりまとめた。

表1 図1のルールに違反している現在の運用

| 項番  | 現在の運用  |
|-----|--|
| (一) | Zサービスのログは、空き容量が一定値以下になった場合、古いログから上書きされる。                   |
| (二) | Zサービスの運用は、営業部のC主任とD主任の2名で担当していたが、D主任が先月退職したので、現在はC主任だけである。 |
| (三) | Zサービスのログの日時は、UTCで記録されている。                                  |
| (四) | Zサービスのログは、毎月末に全てエクスポートし、Yサーバに保管している。                       |

設問 表1中の運用のうち、図1のルール3に違反しているものの項番だけを全て挙げた組合せを、解答群の中から選べ。

解答群

- |                 |                 |
|-----------------|-----------------|
| ア (一), (二)      | イ (一), (二), (三) |
| ウ (一), (二), (四) | エ (一), (三)      |
| オ (一), (三), (四) | カ (一), (四)      |
| キ (二), (三)      | ク (二), (三), (四) |
| ケ (二), (四)      | コ (三), (四)      |

[ メモ用紙 ]

[ メモ用紙 ]

試験問題に記載されている会社名又は製品名は、それぞれ各社の商標又は登録商標です。

なお、試験問題では、<sup>TM</sup> 及び <sup>®</sup> を明記していません。

©2026 独立行政法人情報処理推進機構