

令和5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</u> 1) <u>処理</u> 1 elseif (<u>条件式</u> 2) <u>処理</u> 2 elseif (<u>条件式</u> n) <u>処理</u> n else <u>処理</u> n + 1 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</u> n + 1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</u> n + 1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>) <u>処理</u> 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 <u>処理</u> 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>) <u>処理</u> 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 + -	↑ ↓
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 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。

関数 findPrimeNumbers は、引数で与えられた整数以下の、全ての素数だけを格納した配列を返す関数である。ここで、引数に与える整数は2以上である。

[プログラム]

```

○整数型の配列: findPrimeNumbers(整数型: maxNum)
  整数型の配列: pnList ← {} // 要素数0の配列
  整数型: i, j
  論理型: divideFlag
  for (i を 2 から  まで 1 ずつ増やす)
    divideFlag ← true

    /* iの正の平方根の整数部分が2未満のときは、繰返し処理を実行しない */
    for (j を 2 から iの正の平方根の整数部分 まで 1 ずつ増やす) // α
      if (  )
        divideFlag ← false
        αの行から始まる繰返し処理を終了する
      endif
    endfor
    if (divideFlag が true と等しい)
      pnListの末尾に iの値 を追加する
    endif
  endfor
  return pnList

```

解答群

	a	b
ア	maxNum	$i \div j$ の余りが 0 と等しい
イ	maxNum	$i \div j$ の商が 1 と等しくない
ウ	maxNum + 1	$i \div j$ の余りが 0 と等しい
エ	maxNum + 1	$i \div j$ の商が 1 と等しくない

問2 次の記述中の に入れる正しい答えを、解答群の中から選べ。

次のプログラムにおいて、手続 proc2 を呼び出すと、 の順に出力される。

[プログラム]

○proc1()
 "A" を出力する
 proc3()

○proc2()
 proc3()
 "B" を出力する
 proc1()

○proc3()
 "C" を出力する

解答群

ア "A", "B", "B", "C"

ウ "A", "C", "B", "C"

オ "B", "C", "B", "A"

キ "C", "B", "A"

イ "A", "C"

エ "B", "A", "B", "C"

カ "C", "B"

ク "C", "B", "A", "C"

問3 次の記述中の 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。

次の手続 sort は、大域の整数型の配列 data の、引数 first で与えられた要素番号から引数 last で与えられた要素番号までの要素を昇順に整列する。ここで、 $first < last$ とする。手続 sort を $sort(1, 5)$ として呼び出すと、`/** α **/` の行を最初に実行したときの出力は “ ” となる。

[プログラム]

大域: 整数型の配列: $data \leftarrow \{2, 1, 3, 5, 4\}$

```
○sort(整数型: first, 整数型: last)
  整数型: pivot, i, j
  pivot  $\leftarrow$  data[(first + last)  $\div$  2 の商]
  i  $\leftarrow$  first
  j  $\leftarrow$  last

  while (true)
    while (data[i] < pivot)
      i  $\leftarrow$  i + 1
    endwhile
    while (pivot < data[j])
      j  $\leftarrow$  j - 1
    endwhile
    if (i  $\geq$  j)
      繰返し処理を終了する
    endif
    data[i]とdata[j]の値を入れ替える
    i  $\leftarrow$  i + 1
    j  $\leftarrow$  j - 1
  endwhile
  dataの全要素の値を要素番号の順に空白区切りで出力する  /**  $\alpha$  **/
  if (first < i - 1)
    sort(first, i - 1)
  endif
  if (j + 1 < last)
    sort(j + 1, last)
  endif
```

解答群

ア 1 2 3 4 5

ウ 2 1 3 4 5

イ 1 2 3 5 4

エ 2 1 3 5 4

問4 次の記述中の 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 add は、引数で指定された正の整数 value を大域の整数型の配列 hashArray に格納する。格納できた場合は true を返し、格納できなかった場合は false を返す。ここで、整数 value を hashArray のどの要素に格納すべきかを、関数 calcHash1 及び calcHash2 を利用して決める。

手続 test は、関数 add を呼び出して、hashArray に正の整数を格納する。手続 test の処理が終了した直後の hashArray の内容は、 である。

[プログラム]

大域：整数型の配列：hashArray

- 論理型: add(整数型: value)
整数型: $i \leftarrow \text{calcHash1}(\text{value})$
if (hashArray[i] = -1)
hashArray[i] \leftarrow value
return true
else
i \leftarrow calcHash2(value)
if (hashArray[i] = -1)
hashArray[i] \leftarrow value
return true
endif
endif
return false
- 整数型: calcHash1(整数型: value)
return (value mod hashArrayの要素数) + 1
- 整数型: calcHash2(整数型: value)
return ((value + 3) mod hashArrayの要素数) + 1
- test()
hashArray \leftarrow {5個の -1}
add(3)
add(18)
add(11)

解答群

ア $\{-1, 3, -1, 18, 11\}$

イ $\{-1, 11, -1, 3, -1\}$

ウ $\{-1, 11, -1, 18, -1\}$

エ $\{-1, 18, -1, 3, 11\}$

オ $\{-1, 18, 11, 3, -1\}$

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。

コサイン類似度は、二つのベクトルの向きの類似性を測る尺度である。関数 calcCosineSimilarity は、いずれも要素数が $n(n \geq 1)$ である実数型の配列 vector1 と vector2 を受け取り、二つの配列のコサイン類似度を返す。配列 vector1 が $\{a_1, a_2, \dots, a_n\}$ 、配列 vector2 が $\{b_1, b_2, \dots, b_n\}$ のとき、コサイン類似度は次の数式で計算される。ここで、配列 vector1 と配列 vector2 のいずれも、全ての要素に 0 が格納されていることはないものとする。

$$\frac{a_1b_1 + a_2b_2 + \dots + a_nb_n}{\sqrt{a_1^2 + a_2^2 + \dots + a_n^2} \sqrt{b_1^2 + b_2^2 + \dots + b_n^2}}$$

[プログラム]

```
○実数型: calcCosineSimilarity(実数型の配列: vector1,  
                             実数型の配列: vector2)  
実数型: similarity, numerator, denominator, temp ← 0  
整数型: i  
numerator ← 0  
  
for (i を 1 から vector1の要素数 まで 1 ずつ増やす)  
    numerator ← numerator +   
endfor  
  
for (i を 1 から vector1の要素数 まで 1 ずつ増やす)  
    temp ← temp + vector1[i]の2乗  
endfor  
denominator ← tempの正の平方根  
  
temp ← 0  
for (i を 1 から vector2の要素数 まで 1 ずつ増やす)  
    temp ← temp + vector2[i]の2乗  
endfor  
denominator ←   
  
similarity ← numerator ÷ denominator  
return similarity
```

解答群

	a	b
ア	$(\text{vector1}[i] \times \text{vector2}[i])$ の正の平方根	denominator \times (tempの正の平方根)
イ	$(\text{vector1}[i] \times \text{vector2}[i])$ の正の平方根	denominator + (tempの正の平方根)
ウ	$(\text{vector1}[i] \times \text{vector2}[i])$ の正の平方根	tempの正の平方根
エ	$\text{vector1}[i] \times \text{vector2}[i]$	denominator \times (tempの正の平方根)
オ	$\text{vector1}[i] \times \text{vector2}[i]$	denominator + (tempの正の平方根)
カ	$\text{vector1}[i] \times \text{vector2}[i]$	tempの正の平方根
キ	$\text{vector1}[i]$ の2乗	denominator \times (tempの正の平方根)
ク	$\text{vector1}[i]$ の2乗	denominator + (tempの正の平方根)
ケ	$\text{vector1}[i]$ の2乗	tempの正の平方根

問6 A社は、放送会社や運輸会社向けに広告制作ビジネスを展開している。A社は、人事業務の効率化を図るべく、人事業務の委託を検討することにした。A社が委託する業務（以下、B業務という）を図1に示す。

- ・採用予定者から郵送されてくる入社時の誓約書、前職の源泉徴収票などの書類をPDFファイルに変換し、ファイルサーバに格納する。
(省略)

図1 B業務

委託先候補のC社は、B業務について、次のようにA社に提案した。

- ・B業務だけに従事する専任の従業員を割り当てる。
- ・B業務では、図2の複合機のスキャン機能を使用する。

- ・スキャン機能を使用する際は、従業員ごとに付与した利用者IDとパスワードをパネルに入力する。
- ・スキャンしたデータをPDFファイルに変換する。
- ・PDFファイルを従業員ごとに異なる鍵で暗号化して、電子メールに添付する。
- ・スキャンを実行した本人宛てに電子メールを送信する。
- ・PDFファイルが大きい場合は、PDFファイルを添付する代わりに、自社の社内ネットワーク上に設置したサーバ（以下、Bサーバという）¹⁾に自動的に保存し、保存先のURLを電子メールの本文に記載して送信する。

注¹⁾ Bサーバにアクセスする際は、従業員ごとの利用者IDとパスワードが必要になる。

図2 複合機のスキャン機能（抜粋）

A社は、C社と業務委託契約を締結する前に、秘密保持契約を締結した。その後、C社に質問表を送付し、回答を受けて、業務委託での情報セキュリティリスクの評価を実施した。その結果、図3の発見があった。

- ・複合機のスキャン機能では、電子メールの差出人アドレス、件名、本文及び添付ファイル名を初期設定¹⁾の状態で使用しており、誰がスキャンを実行しても同じである。
- ・複合機のスキャン機能の初期設定情報はベンダーのWebサイトで公開されており、誰でも閲覧できる。

注¹⁾ 複合機の初期設定はC社の情報システム部だけが変更可能である。

図3 発見事項

そこで、A 社では、初期設定の状態のままでは A 社にとって情報セキュリティリスクがあり、初期設定から変更するという対策が必要であると評価した。

設問 対策が必要であると A 社が評価した情報セキュリティリスクはどれか。解答群のうち、最も適切なものを選べ。

解答群

ア B 業務に従事する従業員が、攻撃者からの電子メールを複合機からのものと信じて本文中にある URL をクリックし、フィッシングサイトに誘導される。その結果、A 社の採用予定者の個人情報が漏えいする。

イ B 業務に従事する従業員が、複合機から送信される電子メールをスパムメールと誤認し、電子メールを削除する。その結果、再スキャンが必要となり、B 業務が遅延する。

ウ 攻撃者が、複合機から送信される電子メールを盗聴し、添付ファイルを暗号化して身代金を要求する。その結果、A 社が復号鍵を受け取るために多額の身代金を支払うことになる。

エ 攻撃者が、複合機から送信される電子メールを盗聴し、本文に記載されている URL を使って B サーバにアクセスする。その結果、A 社の採用予定者の個人情報が漏えいする。

[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、TM 及び [®] を明記していません。

©2023 独立行政法人情報処理推進機構