

令和 6 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，応用情報技術者試験用）

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	選択処理を示す。 <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。 各<u>処理</u>は，0以上の文の集まりである。 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。 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。
while (<u>条件式</u>) <u>処理</u> endwhile	前判定繰返し処理を示す。 <u>条件式</u>が真の間，<u>処理</u>を繰返し実行する。 <u>処理</u>は，0以上の文の集まりである。
do <u>処理</u> while (<u>条件式</u>)	後判定繰返し処理を示す。 <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。 <u>処理</u>は，0以上の文の集まりである。
for (<u>制御記述</u>) <u>処理</u> endfor	繰返し処理を示す。 <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。 <u>処理</u>は，0以上の文の集まりである。

〔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 に入れる正しい答えを，解答群の中から選べ。

関数 maximum は，異なる三つの整数を引数で受け取り，そのうちの最大値を返す。

〔プログラム〕

○整数型: maximum(整数型: x, 整数型: y, 整数型: z)

```
if (  )
```

```
    return x
```

```
elseif (y > z)
```

```
    return y
```

```
else
```

```
    return z
```

```
endif
```

解答群

ア $x > y$

イ $x > y$ and $x > z$

ウ $x > y$ and $y > z$

エ $x > z$

オ $x > z$ and $z > y$

カ $z > y$

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 `convDecimal` は、引数として与えられた、“0” と “1” だけから成る、1 文字以上の文字列を、符号なしの 2 進数と解釈したときの整数値を返す。例えば、引数として “10010” を与えると 18 が返る。

関数 `convDecimal` が利用する関数 `int` は、引数で与えられた文字が “0” なら整数値 0 を返し、“1” なら整数値 1 を返す。

[プログラム]

```
○整数型: convDecimal(文字列型: binary)
  整数型: i, length, result ← 0
  length ← binaryの文字数
  for (i を 1 から length まで 1 ずつ増やす)
    result ← 
  endfor
  return result
```

解答群

- ア `result + int(binary の (length - i + 1)文字目の文字)`
- イ `result + int(binary の i文字目の文字)`
- ウ `result × 2 + int(binary の (length - i + 1)文字目の文字)`
- エ `result × 2 + int(binary の i文字目の文字)`

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。

図 1 に示すグラフの頂点には，1 から順に整数で番号が付けられている。グラフは無向グラフであり，各頂点間には高々一つの辺がある。一つの辺は両端の頂点の番号を要素にもつ要素数 2 の整数型の配列で表現できる。例えば，{1, 3} は頂点 1 と頂点 3 を端点とする辺を表す。グラフ全体は，グラフに含まれる辺を表す要素数 2 の配列を全て格納した配列（以下，辺の配列という）で表現できる。辺の配列の要素数はグラフの辺の個数と等しい。図 1 のグラフは整数型配列の配列 {{1, 3}, {1, 4}, {3, 4}, {2, 4}, {4, 5}} と表現できる。

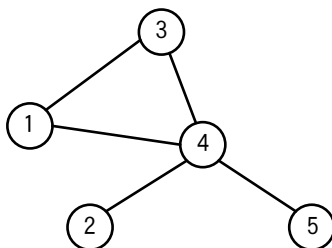


図 1 グラフの例

関数 `edgesToMatrix` は，辺の配列を隣接行列に変換する。隣接行列とは，グラフに含まれる頂点の個数と等しい行数及び列数の正方行列で， i 行 j 列の成分は頂点 i と頂点 j を結ぶ辺があるときに 1 となり，それ以外は 0 となる。行列の対角成分は全て 0 で，無向グラフの場合は対称行列になる。図 1 のグラフを表現する隣接行列を図 2 に示す。

$$\begin{pmatrix} 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

図 2 図 1 のグラフを表現する隣接行列

関数 edgesToMatrix は、引数 edgeList で辺の配列を、引数 nodeNum でグラフの頂点の個数をそれぞれ受け取り、隣接行列を表す整数型の二次元配列を返す。

〔プログラム〕

```
○整数型の二次元配列: edgesToMatrix(整数型配列の配列: edgeList,  
                                     整数型: nodeNum)  
整数型の二次元配列: adjMatrix ← {nodeNum行nodeNum列の 0}  
整数型: i, u, v  
for (i を 1 から edgeListの要素数 まで 1 ずつ増やす)  
  u ← edgeList[i][1]  
  v ← edgeList[i][2]  
    
endfor  
return adjMatrix
```

解答群

- ア adjMatrix[u, u] ← 1
- イ adjMatrix[u, u] ← 1
adjMatrix[v, v] ← 1
- ウ adjMatrix[u, v] ← 1
- エ adjMatrix[u, v] ← 1
adjMatrix[v, u] ← 1
- オ adjMatrix[v, u] ← 1
- カ adjMatrix[v, v] ← 1

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 merge は、昇順に整列された整数型の配列 data1 及び data2 を受け取り、これらを併合してできる昇順に整列された整数型の配列を返す。

関数 merge を merge({2, 3}, {1, 4}) として呼び出すと、`/**  $\alpha$  */` の行は 。

[プログラム]

○整数型の配列: merge(整数型の配列: data1, 整数型の配列: data2)

整数型: $n1 \leftarrow \text{data1の要素数}$

整数型: $n2 \leftarrow \text{data2の要素数}$

整数型の配列: $\text{work} \leftarrow \{(n1 + n2)\text{個の 未定義の値}\}$

整数型: $i \leftarrow 1$

整数型: $j \leftarrow 1$

整数型: $k \leftarrow 1$

while $((i \leq n1) \text{ and } (j \leq n2))$

if $(\text{data1}[i] \leq \text{data2}[j])$

work[k] $\leftarrow \text{data1}[i]$

$i \leftarrow i + 1$

else

work[k] $\leftarrow \text{data2}[j]$

$j \leftarrow j + 1$

endif

$k \leftarrow k + 1$

endwhile

while $(i \leq n1)$

work[k] $\leftarrow \text{data1}[i]$

$i \leftarrow i + 1$

$k \leftarrow k + 1$

endwhile

while $(j \leq n2)$

work[k] $\leftarrow \text{data2}[j]$ /*** α ***/

$j \leftarrow j + 1$

$k \leftarrow k + 1$

endwhile

return work

解答群

ア 実行されない

イ 1回実行される

ウ 2回実行される

エ 3回実行される

問5 次のプログラム中の a ～ c 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。

一度の注文で購入された商品のリストを、注文ごとに記録した注文データがある。表に、注文データの例を示す。

表 注文データの例

注文番号	購入された商品のリスト
1	A, B, D
2	A, D
3	A
4	A, B, E
5	B
6	C, E

注文データから、商品 x と商品 y とが同一の注文で購入されやすい傾向を示す関連度 L_{xy} を、次の式で計算する。

$$L_{xy} = \frac{M_{xy} \times \text{全注文数}}{K_x \times K_y}$$

ここで、 M_{xy} は商品 x と商品 y とが同一の注文で購入された注文数、 K_x は商品 x が購入された注文数、 K_y は商品 y が購入された注文数を表す。表の例では、 M_{AB} が 2、全注文数が 6、 K_A が 4、 K_B が 3 であるので、商品 A と商品 B の関連度 L_{AB} は、 $(2 \times 6) / (4 \times 3) = 1.0$ である。

手続 putRelatedItem は、大域変数 orders に格納された注文データを基に、引数で与えられた商品との関連度が最も大きい商品のうちのひとつと、その関連度を出力する。プログラムでは、商品は文字列で表し、注文は購入された商品の配列、注文データは注文の配列で表している。注文データには 2 種類以上の商品が含まれるものとする。また、注文データにある商品以外の商品が、引数として与えられることはないものとする。

[プログラム]

```
// 注文データ（ここでは表の例を与えている）
大域：文字列型配列の配列：orders ← {{ "A", "B", "D"}, {"A", "D"}, {"A"},
                                         {"A", "B", "E"}, {"B"}, {"C", "E"}}
```

OutputRelatedItem(文字列型：item)

```
文字列型の配列：allItems ← ordersに含まれる文字列を
                           重複なく辞書順に格納した配列
                           // 表の例では {"A", "B", "C", "D", "E"}
```

```
文字列型の配列：otherItems ← allItemsの複製から値がitemである
                           要素を除いた配列
```

```
整数型：i, itemCount ← 0
整数型の配列：arrayK ← {otherItemsの要素数個の0}
整数型の配列：arrayM ← {otherItemsの要素数個の0}
実数型：valueL, maxL ←  $-\infty$ 
文字列型の配列：order
文字列型：relatedItem
```

```
for (orderにordersの要素を順に代入する)
  if (orderのいずれかの要素の値がitemの値と等しい)
    itemCountの値を1増やす
  endif
  for (iを1からotherItemsの要素数まで1ずつ増やす)
    if (orderのいずれかの要素の値がotherItems[i]の値と等しい)
      if (orderのいずれかの要素の値がitemの値と等しい)
        a の値を1増やす
      endif
      b の値を1増やす
    endif
  endfor
endfor
for (iを1からotherItemsの要素数まで1ずつ増やす)
  valueL ← (arrayM[i] × c) ÷ (itemCount × arrayK[i])
  /* 実数として計算する */

  if (valueLがmaxLより大きい)
    maxL ← valueL
    relatedItem ← otherItems[i]
  endif
endfor
relatedItemの値とmaxLの値をこの順にコンマ区切りで出力する
```

解答群

	a	b	c
ア	arrayK[i]	arrayM[i]	allItemsの要素数
イ	arrayK[i]	arrayM[i]	ordersの要素数
ウ	arrayK[i]	arrayM[i]	otherItemsの要素数
エ	arrayM[i]	arrayK[i]	allItemsの要素数
オ	arrayM[i]	arrayK[i]	ordersの要素数
カ	arrayM[i]	arrayK[i]	otherItemsの要素数

〔 メ モ 用 紙 〕

問6 A社は従業員450名の商社であり、昨年から働き方改革の一環として、在宅でのテレワークを推進している。A社のシステム環境を図1に示す。

- ・従業員には、一人に1台デスクトップPC（以下、社内PCという）を貸与している。
- ・従業員が利用するシステムには、自社で開発しA社に設置している業務システムのほかに、次の二つのSaaS（以下、二つのSaaSをA社利用クラウドサービスという）がある。
 1. メール機能、チャット機能及びクラウドストレージ機能をもつグループウェア（以下、A社利用グループウェアという）
 2. オンライン会議サービス
- ・テレワークでは、従業員の個人所有PC（以下、私有PCという）の業務利用（BYOD）を許可している。
- ・テレワークでは、社内PC及び私有PCのそれぞれに専用のアプリケーションソフトウェア（以下、専用アプリという）を導入し、社内PCのデスクトップから私有PCに画面転送を行うリモートデスクトップ方式を採用している。
- ・専用アプリには、リモートデスクトップからPCへのファイルのダウンロード及びファイル、文字列、画像などのコピー＆ペーストを禁止する機能（以下、保存禁止機能という）があり、A社では私有PCに対して当該機能を有効にしている。
- ・業務システムには、社内PCのデスクトップから利用者ID及びパスワードを入力してログインしている。
- ・A社利用クラウドサービスへのログインは、A社利用クラウドサービス側の設定によってA社の社内ネットワークからだけ可能になるように制限している。ログインには利用者ID及びパスワードを用いている。

図1 A社のシステム環境（抜粋）

テレワークの定着が進むにつれて、社内PCからインターネットへの接続が極端に遅くなり、業務に支障をきたしているのを改善できないかと、従業員から問合せがあった。A社の社内ネットワークとインターネットとの間の通信量を調査したところ、テレワーク導入前に比べ、業務時間帯で顕著に増加していることが判明した。そのため、情報システム部では、テレワークでA社利用クラウドサービスに接続する場合には、A社の社内ネットワークも社内PCも介さずに直接接続することを可能にするネットワークの設定変更を実施することにした。

設定変更に当たり、情報セキュリティ上の問題がないかをA社の情報セキュリティリーダーであるBさんが検討したところ、幾つか問題があることが分かった。その一つは、A社利用クラウドサービスへの不正アクセスのリスクが増加することである。そこでBさんは、リスクを低減するために、情報システム部に対策を依頼することにした。

設問 次の対策のうち、情報システム部に依頼することにしたものはどれか。解答群のうち、最も適切なものを選び。

解答群

- ア A社の社内ネットワークからA社利用クラウドサービスへの通信を監視する。
- イ A社の社内ネットワークとA社利用クラウドサービスとの間の通信速度を制限する。
- ウ A社利用クラウドサービスにA社外から接続する際の認証に2要素認証を導入する。
- エ A社利用クラウドサービスのうち、A社利用グループウェアだけを直接接続の対象とする。
- オ 専用アプリの保存禁止機能を無効にする。

〔 メ モ 用 紙 〕

〔 メ モ 用 紙 〕

〔 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。

©2024 独立行政法人情報処理推進機構