

令和 7 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	<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。

関数 function1 が受け取る引数と、関数 function2 が受け取る引数とが同じとき、二つの関数は同じ値を返す。ここで、引数 n と引数 m は正の整数であり、引数 m は引数 n よりも 10 以上大きい数とする。

〔プログラム〕

○整数型: function1(整数型: n, 整数型: m)

整数型: count ← 0

整数型: i

for (iをnからmまで1ずつ増やす)

if ((i mod 4)が0と等しい)

count ← count + 1

endif

endfor

return count

○整数型: function2(整数型: n, 整数型: m)

整数型: count ← 0

整数型: tempN ← n

整数型: i, j

for (  )

if ((tempN mod 4)が0と等しい)

繰返し処理を終了する

endif

tempN ← tempN + 1

endfor

for (  )

count ← count + 1

endfor

return count

解答群

	a	b
ア	iを1から2まで1ずつ増やす	jをnから始めてmを超えない範囲でtempNずつ増やす
イ	iを1から2まで1ずつ増やす	jをtempNからmまで1ずつ増やす
ウ	iを1から2まで1ずつ増やす	jをtempNから始めてmを超えない範囲で4ずつ増やす
エ	iを1から3まで1ずつ増やす	jをnから始めてmを超えない範囲でtempNずつ増やす
オ	iを1から3まで1ずつ増やす	jをtempNからmまで1ずつ増やす
カ	iを1から3まで1ずつ増やす	jをtempNから始めてmを超えない範囲で4ずつ増やす

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 change は，10 より大きい整数を引数  $n$  で受け取り，1 円玉，5 円玉，10 円玉を使ってちょうど  $n$  円にする組合せの総数を返す。

例えば，12 円にする組合せは，次のように数えられる。10 円玉を使わない場合には，1 円玉と 5 円玉だけでちょうど 12 円にすることになる。その組合せは，使える 5 円玉の枚数が 0 以上 ( $12 \div 5$  の商) 以下なので， $(12 \div 5 \text{ の商}) + 1 = 3$  通りある。同様に，10 円玉を 1 枚使う場合には，1 円玉と 5 円玉だけでちょうど 2 円にすることになり，その組合せは  $(2 \div 5 \text{ の商}) + 1 = 1$  通りある。10 円玉を 2 枚以上使う組合せはない。よって，1 円玉，5 円玉，10 円玉を使ってちょうど 12 円にする組合せは， $3 + 1 = 4$  通りである。

[プログラム]

```
○整数型: change(整数型: n)
  整数型: count ← 0
  整数型: rest ← n
  while (  )
    count ← count + (rest ÷ 5 の商) + 1
    rest ← rest - 10
  endwhile
  return count
```

解答群

- ア  $\text{rest} \geq 0$
- イ  $\text{rest} \geq 5$
- ウ  $\text{rest} \geq 10$
- エ  $\text{rest} > 0$
- オ  $\text{rest} > 5$
- カ  $\text{rest} > 10$

〔 メ モ 用 紙 〕

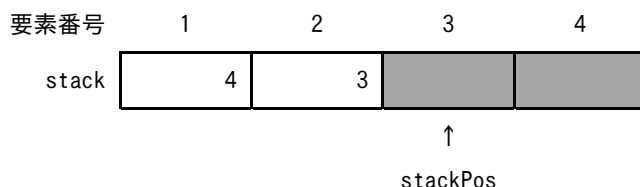
問3 次のプログラム中の a と b に入れる正しい答えの組合せを、解答群の中から選べ。ここで、配列の要素番号は 1 から始まる。

関数 push は、引数で与えられた整数をスタックに格納する。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。

関数 pop は、スタックから値を取り出して返す。スタックが空のときは未定義の値を返す。

スタックを、要素数が 4 である大域の整数型の配列 stack、及び次に値を格納する位置を示す大域の変数 stackPos で表現する。スタックの初期状態を図に示す。

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。



注記 網掛けはその要素が未定義であることを示す。

図 スタックの初期状態



[プログラム]

大域: 整数型:  $\text{stackPos} \leftarrow 3$

大域: 整数型の配列:  $\text{stack} \leftarrow \{4, 3, \text{未定義の値}, \text{未定義の値}\}$

```
○論理型: push(整数型: inputData)
  if ( $\text{stackPos} \leq \text{stackの要素数}$ )
     $\text{stack}[\text{a}] \leftarrow \text{inputData}$ 
     $\text{stackPos} \leftarrow \text{stackPos} + 1$ 
    return true
  else
    return false
  endif
```

```
○整数型: pop()
  整数型: popData  $\leftarrow$  未定義の値
  if ( $\text{stackPos} > 1$ )
     $\text{stackPos} \leftarrow \text{b}$ 
    popData  $\leftarrow \text{stack}[\text{stackPos}]$ 
     $\text{stack}[\text{stackPos}] \leftarrow$  未定義の値
  endif
  return popData
```

解答群

	a	b
ア	$\text{stackPos}$	$\text{stackPos} + 1$
イ	$\text{stackPos}$	$\text{stackPos} - 1$
ウ	$\text{stackPos} - 1$	$\text{stackPos} + 1$
エ	$\text{stackPos} - 1$	$\text{stackPos} - 1$

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 search は、二つの文字型の配列を、それぞれ引数 data 及び key で受け取り、data から、key の要素の並びと同じ並びを全て探し、その先頭の要素番号を全て格納した配列を返す。見つからなければ、要素数 0 の配列を返す。

関数 search を search({"a", "b", "a", "b", "c", "a", "b", "c"}, {"a", "b", "c"}) として呼び出すと、`/** β /**`の行の条件式が真となる回数は  回である。

[プログラム]

```
/* keyは、要素数1以上の配列である */
○整数型の配列: search(文字型の配列: data, 文字型の配列: key)
  整数型: i, j, lenData, lenKey
  整数型の配列: result ← {} // 要素数0の配列

lenData ← dataの要素数
lenKey ← keyの要素数

/* (lenData - lenKey + 1) が0以下のときは繰返し処理を実行しない */
for (i を 1 から (lenData - lenKey + 1) まで 1 ずつ増やす)
  for (j を 1 から lenKey まで 1 ずつ増やす) // α
    if (data[i + j - 1] が key[j] と等しい) /** β /**
      if (j が lenKey と等しい)
        resultの末尾 に iの値 を追加する
      endif
    else
      αの行から始まる繰返し処理を終了する
    endif
  endfor
endfor
return result
```

解答群

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

〔 メ モ 用 紙 〕

問5 次の記述中の a と b 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。

予防接種の病気 X に対する予防効果を調査するために集めたデータの集計結果を基に，病気 X にかかるかどうか，予防接種の有無に影響されないと仮定した場合の人数を計算する。この人数を理論度数という。表 1 に集計結果の例を示し，表 2 に表 1 を基に計算した理論度数を示す。

関数 f は，引数 data で受け取った集計結果を基に計算した理論度数を返す。引数と戻り値は二次元配列で，その行が表の行，その列が表の列に対応する。

表 1 集計結果の例

	単位 人	
	病気 X にかからなかった	病気 X にかかった
予防接種を受けた	82	6
予防接種を受けていない	58	8

表 2 表 1 を基に計算した理論度数

	単位 人	
	病気 X にかからなかった	病気 X にかかった
予防接種を受けた	<span style="border: 1px solid black; padding: 2px 10px;">a</span>	
予防接種を受けていない		<span style="border: 1px solid black; padding: 2px 10px;">b</span>

注記 網掛けの部分は表示していない。

[プログラム]

```
○実数型の二次元配列: f(実数型の二次元配列: data)
  実数型: t ← dataの要素の和
  整数型: row ← dataの行数
  整数型: col ← dataの列数
  実数型の二次元配列: result ← {row行col列の 未定義の値}
  整数型: r, c
  for (r を 1 から row まで 1 ずつ増やす)
    for (c を 1 から col まで 1 ずつ増やす)
      result[r, c] ← (dataの行番号rの要素の和) ×
                      (dataの列番号cの要素の和) ÷ t
    endfor
  endfor
  return result
```

解答群

	a	b
ア	44	33
イ	58	8
ウ	70	7
エ	75	2
オ	80	6
カ	80	8
キ	82	6

問6 A社は従業員200名の電子機器メーカーである。東京に本社があり、新潟に工場がある。

A社では、ファイルサーバを図1のように運用している。

- ・ファイルサーバは本社と工場のサーバールームに設置している。
- ・ファイルサーバは、磁気テープを使用してファイルのバックアップを取得している。
- ・土曜日の午前2時からフルバックアップを取得し、翌週の火曜日と木曜日の午前2時から増分バックアップを取得している。
- ・情報システム部の担当者は毎週月曜日、火曜日、木曜日の朝10時頃、バックアップジョブの実行結果の確認と磁気テープの交換を行い、磁気テープは耐火金庫に保管している。
- ・磁気テープには上書きモードでバックアップを取得し、1か月分の磁気テープを世代管理している。
- ・これまで、フルバックアップからのリストアには平均して4時間、1回の増分バックアップからは平均して0.25時間掛かっている。
- ・A社はICT継続のための試験を実施しており、ファイルサーバも試験対象である。

注記 事業影響度分析の結果に基づき、ファイルサーバは、72時間の目標復旧時点(RP0)と120時間の目標復旧時間(RT0)が要求事項として定められている。

図1 A社のファイルサーバの運用

A社はISMS認証を取得しており、最高情報セキュリティ責任者(CISO)を中心に情報セキュリティに取り組み、定期的に、情報セキュリティリスクアセスメントを行っている。今般、ISMS認証基準がJIS Q 27001:2023に改正されたことを受け、情報セキュリティリーダーのBさんは、新基準への移行審査に向けて準備することになった。改正によって新たに追加された情報セキュリティ管理策について、Bさんは情報システム部の担当者を取組状況を確認し、その評価結果を表1にまとめた。

表1 Bさんの評価結果(抜粋)

項番	情報セキュリティ管理策	評価結果
1	(事業継続のためのICTの備え)事業継続の目的及びICT継続の要求事項に基づいて、ICTの備えを計画し、実施し、維持し、試験しなければならない。	実施している

Bさんは、移行審査前の内部監査で、内部監査室から表1の項番1に関するファイルサーバの運用について何点が質問を受け、表2のとおり回答した。

表 2 内部監査室への B さんの回答（抜粋）

項番	B さんの回答
1	例えば、金曜日の正午に障害が発生した場合、少なくとも <span style="border: 1px solid black; padding: 0 10px;">a1</span> の時点のデータは復元しなければならない。
2	例えば、木曜日の正午に障害が発生し、ファイルサーバの全データが消失したとすると、バックアップからのリストアには <span style="border: 1px solid black; padding: 0 10px;">a2</span> 時間掛かると予想される。
3	ICT 継続の計画書は、 <span style="border: 1px solid black; padding: 0 10px;">a3</span> が承認している。

設問 表 2 中の a1 ～ a3 に入れる字句の適切な組合せを、a に関する解答群の中から選べ。

a に関する解答群

	a1	a2	a3
ア	月曜日の正午	4.25	CIS0
イ	月曜日の正午	4.25	情報システム部の担当者
ウ	月曜日の正午	4.25	内部監査室長
エ	月曜日の正午	4.50	CIS0
オ	月曜日の正午	4.50	情報システム部の担当者
カ	火曜日の正午	4.25	情報システム部の担当者
キ	火曜日の正午	4.25	内部監査室長
ク	火曜日の正午	4.50	CIS0
ケ	火曜日の正午	4.50	情報システム部の担当者
コ	火曜日の正午	4.50	内部監査室長

〔 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、<sup>TM</sup> 及び <sup>®</sup> を明記していません。

©2025 独立行政法人情報処理推進機構