

# 平成 26 年度 春期

## 基本情報技術者

- 午前 問題 ..... 2  
(全80問 試験時間:2時間30分)
- 午後 問題 ..... 48  
(全 7 問 試験時間:2時間30分)
  - 問1 : 必須解答
  - 問2～問7 : 6 問中 4 問選択
  - 問8 : 必須解答
  - 問9～問13 : 5 問中 1 問選択 ※



※「表計算ソフトの機能・用語」の仕様改訂に伴い、問 13 の表記を一部改変しています。

平成 27 年度 秋

平成 27 年度 春

平成 26 年度 秋

平成 26 年度 春

# 平成 26 年度 春 午前 問題

問 1 から問 50 までは、テクノロジ系の問題です。

☐ **問 1** 次の 10 進小数のうち、2 進数で表すと無限小数になるものはどれか。

☐ **ア** 0.05

☐ **イ** 0.125

☐ **ウ** 0.375

☐ **エ** 0.5

☐ **問 2** 最上位をパリティビットとする 8 ビット符号において、パリティビット以外の下位 7 ビットを得るためのビット演算はどれか。

☐ **ア** 16 進数 0F との AND をとる。

☐ **イ** 16 進数 0F との OR をとる。

☐ **ウ** 16 進数 7F との AND をとる。

☐ **エ** 16 進数 FF との XOR（排他的論理和）をとる。

☐ **問 3** 論理式  $\bar{A} \cdot \bar{B} \cdot C + A \cdot \bar{B} \cdot C + \bar{A} \cdot B \cdot C + A \cdot B \cdot C$  と恒等的に等しいものはどれか。ここで、 $\cdot$  は論理積、 $+$  は論理和、 $\bar{A}$  は  $A$  の否定を表す。

☐ **ア**  $A \cdot B \cdot C$

☐ **イ**  $A \cdot B \cdot C + \bar{A} \cdot \bar{B} \cdot C$

☐ **ウ**  $A \cdot B + B \cdot C$

☐ **エ**  $C$

解説

## 問 1 2 進小数

2 進小数  $0.X_1X_2X_3X_4\cdots$  は、10 進数では  $X_1 \times 2^{-1} + X_2 \times 2^{-2} + X_3 \times 2^{-3} + X_4 \times 2^{-4} \cdots$  のように表すことができます。これを分数で表すと、次のようになります。

$$0.X_1X_2X_3X_4\cdots = \frac{X_1}{2} + \frac{X_2}{4} + \frac{X_3}{8} + \frac{X_4}{16} + \cdots$$

右辺が有限個の分数の足し算であれば、通分すると必ず分母が 2 の  $n$  乗の分数になります。したがって、10 進小数を分数に直して、分母が 2 の  $n$  乗にならないものは、2 進小数では無限小数になります。

○ **ア**  $0.05 = \frac{5}{100} = \frac{1}{20} \rightarrow$  無限小数

× **イ**  $0.125 = \frac{125}{1000} = \frac{1}{8} = \frac{1}{2^3} \rightarrow$  有限小数

## 午前のカギ

### 覚えよう！

問 1

#### 10 進小数 → 2 進小数への変換

小数部に 2 を掛け、結果の整数部を取り出す。これを、小数点以下が 0 になるまで繰り返し、整数部を上から並べる

例：10 進小数 0.625 を 2 進小数に変換

$$\begin{array}{rcl} 0.625 \times 2 & = & 1.25 \\ 0.25 \times 2 & = & 0.5 \\ 0.5 \times 2 & = & 1.0 \\ & \downarrow & \\ & 101 & \end{array}$$

$$\times \text{ウ} \quad 0.375 = \frac{375}{1000} = \frac{3}{8} = \frac{3}{2^3} \rightarrow \text{有限小数}$$

$$\times \text{エ} \quad 0.5 = \frac{5}{10} = \frac{1}{2} \rightarrow \text{有限小数}$$

## 問 2 論理演算 キホン!

選択肢から、最上位ビットが 0、それ以外は元のビットのままになるような論理演算を選びます（例として、元のビットを 10101010 とします）。

$$\begin{array}{rcl} \times \text{ア} & 10101010 & \leftarrow \text{元のビット} \\ & \text{AND } 00001111 & \leftarrow 0F \text{ との AND をとる} \\ \hline & 00001010 & \leftarrow \text{上位 4 ビットが 0 になる} \end{array}$$

$$\begin{array}{rcl} \times \text{イ} & 10101010 & \leftarrow \text{元のビット} \\ & \text{OR } 00001111 & \leftarrow 0F \text{ との OR をとる} \\ \hline & 10101111 & \leftarrow \text{下位 4 ビットが 1 になる} \end{array}$$

$$\begin{array}{rcl} \bigcirc \text{ウ} & 10101010 & \leftarrow \text{元のビット} \\ & \text{AND } 01111111 & \leftarrow 7F \text{ との AND をとる} \\ \hline & 00101010 & \leftarrow \text{最上位ビットが 0 になる} \end{array}$$

$$\begin{array}{rcl} \times \text{エ} & 10101010 & \leftarrow \text{元のビット} \\ & \text{XOR } 11111111 & \leftarrow FF \text{ との XOR をとる} \\ \hline & 01010101 & \leftarrow \text{全ビットが反転する} \end{array}$$

## 問 3 論理式

論理式は、①  $\bar{A} \cdot \bar{B} \cdot C$ 、②  $A \cdot \bar{B} \cdot C$ 、③  $\bar{A} \cdot B \cdot C$ 、④  $A \cdot B \cdot C$  の 4 つの論理積からなります。①～④の各項が真になるのは、 $A$ 、 $B$ 、 $C$  の値が以下の組合せの場合だけです（真 = 1、偽 = 0 とする）。

- ①  $\bar{A} \cdot \bar{B} \cdot C \rightarrow A = 0, B = 0, C = 1$  のとき 1
- ②  $A \cdot \bar{B} \cdot C \rightarrow A = 1, B = 0, C = 1$  のとき 1
- ③  $\bar{A} \cdot B \cdot C \rightarrow A = 0, B = 1, C = 1$  のとき 1
- ④  $A \cdot B \cdot C \rightarrow A = 1, B = 1, C = 1$  のとき 1

各項は論理和でつながっているので、①～④のどれか 1 つが真なら、論理式全体も真になります。以上を真理値表にまとめると、結果は下表のように、 $C$  の真偽値と同じになります。正解は **エ** です。

A	B	C	①	②	③	④	①+②+③+④
0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	1
0	1	0	0	0	0	0	0
0	1	1	0	0	1	0	1
1	0	0	0	0	0	0	0
1	0	1	0	1	0	0	1
1	1	0	0	0	0	0	0
1	1	1	0	0	0	1	1

以下のように、論理式を変形しても解けます。

$$\begin{aligned} & \bar{A} \cdot \bar{B} \cdot C + A \cdot \bar{B} \cdot C + \bar{A} \cdot B \cdot C + A \cdot B \cdot C \\ &= (\bar{A} \cdot \bar{B} + A \cdot \bar{B}) \cdot C + (\bar{A} \cdot B + A \cdot B) \cdot C \\ &= ((\bar{A} + A) \cdot \bar{B}) \cdot C + ((\bar{A} + A) \cdot B) \cdot C \\ &= \bar{B} \cdot C + B \cdot C \\ &= (\bar{B} + B) \cdot C \\ &= C \end{aligned}$$

$$\ast \bar{P} + P = 1, 1 \cdot P = P$$

## パリティビット 問 2

パリティチェック（誤り検出の方式）において、元のビット列に付加するビット。偶数パリティと奇数パリティがあり、元のビット列とパリティビットとを合わせて、1 のビットの個数が偶数または奇数になるように、パリティビットの値が決まる。

## AND（論理積） 問 2

2 つのビットが両方とも 1 のとき 1、それ以外は 0 になる論理演算。

$$\begin{aligned} 0 \text{ AND } 0 &\rightarrow 0 \\ 0 \text{ AND } 1 &\rightarrow 0 \\ 1 \text{ AND } 0 &\rightarrow 0 \\ 1 \text{ AND } 1 &\rightarrow 1 \end{aligned}$$

## OR（論理和） 問 2

2 つのビットが両方とも 0 のとき 0、それ以外は 1 になる論理演算。

$$\begin{aligned} 0 \text{ OR } 0 &\rightarrow 0 \\ 0 \text{ OR } 1 &\rightarrow 1 \\ 1 \text{ OR } 0 &\rightarrow 1 \\ 1 \text{ OR } 1 &\rightarrow 1 \end{aligned}$$

## XOR（排他的論理和） 問 2

入力が 2 つとも同じ値のとき 0、異なる値のとき 1 になる論理演算。

$$\begin{aligned} 0 \text{ XOR } 0 &\rightarrow 0 \\ 0 \text{ XOR } 1 &\rightarrow 1 \\ 1 \text{ XOR } 0 &\rightarrow 1 \\ 1 \text{ XOR } 1 &\rightarrow 0 \end{aligned}$$

## 解答

- 問 1 **ア** 問 2 **ウ**  
問 3 **エ**



**問 4** 正規分布の説明として、適切なものはどれか。

- ア** 故障確率に用いられ、バスタブのような形状をした連続確率分布のこと
- イ** 全ての事象の起こる確率が等しい現象を表す確率分布のこと
- ウ** 平均値を中心とする左右対称で釣鐘状の連続確率分布のこと
- エ** 離散的に発生し、発生確率は一定である離散確率分布のこと



**問 5** 表は、文字列を検査するための状態遷移表である。検査では、初期状態を a とし、文字列の検査中に状態が e になれば不合格とする。

解答群で示される文字列のうち、不合格となるものはどれか。ここで、文字列は左端から検査し、解答群中の△は空白を表す。

		文字				
		空白	数字	符号	小数点	その他
現在の状態	a	a	b	c	d	e
	b	a	b	e	d	e
	c	e	b	e	d	e
	d	a	e	e	e	e

- ア** + 0010
- イ** - 1
- ウ** 12.2
- エ** 9.△

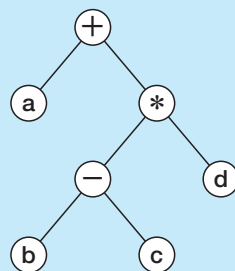


**問 6** 2 分木の各ノードがもつ記号を出力する再帰的なプログラム Proc(*n*) の定義は、次のとおりである。このプログラムを、図の 2 分木の根（最上位のノード）に適用したときの出力はどれか。

```

Proc(n) {
  n に左の子 l があれば Proc(l) を呼び出す。
  n に右の子 r があれば Proc(r) を呼び出す。
  n の記号を出力して終了する。
}

```



- ア** + a \* - b c d
- イ** a + b - c \* d
- ウ** a b c - d \* +
- エ** b - c \* d + a

解説

**問 4** 正規分布

たとえば、ある年齢の日本人の身長を調べると、平均値の人数が最も多く、左右対称の釣鐘型の分布になります。このような分布を**正規分布**といいます。身長や学年のテスト成績、製品の工作誤差など、社会現象や自然

 **午前のカギ**

現象の中には、正規分布にしたがうものが数多く存在します。

- × **ア** バスタブ曲線の説明です。
- × **イ** 一様分布の説明です。
- **ウ** 正解です。
- × **エ** 正規分布は、離散的に発生する事象ではなく、連続的な値に関する確率分布です。離散確率分布には、二項分布やポアソン分布などの種類があります。

## 問 5 状態遷移表 キホン!

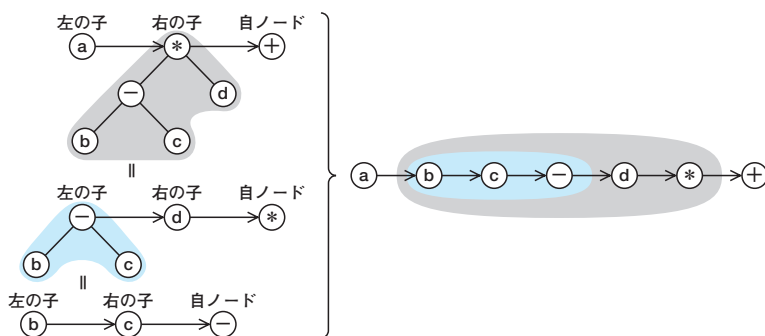
現在の状態 a から開始します。たとえば「+ 0010」の場合、1文字目は符号なので、現在の状態 a の“符号”の列をみます。すると c とあるので、状態 c に遷移します。2文字目は数字なので、現在の状態 c の“数字”の列をみます。すると b とあるので、状態 b に遷移します。以下同様にして、状態 e に遷移するものを探します。

- × **ア**  $(a) \xrightarrow{+ \text{符号}} (c) \xrightarrow{0 \text{数字}} (b) \xrightarrow{0 \text{数字}} (b) \xrightarrow{1 \text{数字}} (b) \xrightarrow{0 \text{数字}} (b)$
- × **イ**  $(a) \xrightarrow{- \text{符号}} (c) \xrightarrow{1 \text{数字}} (b)$
- **ウ**  $(a) \xrightarrow{1 \text{数字}} (b) \xrightarrow{2 \text{数字}} (b) \xrightarrow{\cdot \text{小数点}} (d) \xrightarrow{2 \text{数字}} (e)$
- × **エ**  $(a) \xrightarrow{9 \text{数字}} (b) \xrightarrow{\cdot \text{小数点}} (d) \xrightarrow{\Delta \text{空白}} (a)$

## 問 6 再帰呼出し (2 分木の走査)

プログラムは、各ノードについて、①左の子、②右の子、③自ノードの順に処理します。処理するノードに子がなければ記号を出力して終了しますが、子がある場合はさらに①左の子、②右の子、③自ノードの順に実行します。

これを問題文の 2 分木に当てはめると、次のようになります。

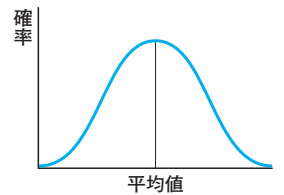


以上から、正解は **ウ** です。このような 2 分木の走査方法を、後順走査といいます。

## 午前のカギ

### 正規分布の例

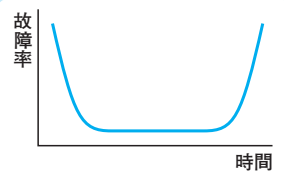
問 4



### バスタブ曲線

問 4

経過時間ともなうシステムの故障率の変化をグラフ化したもの。故障率は一般に導入当初と運用末期に高く、その中間は低いため、バスタブ型の曲線になる。



### 一様分布

問 4

すべての事象の起こる確率が一定である場合の確率分布。たとえばサイコロの目が出る確率は、すべての目で等しく 1/6 になる。

### 2 分木

問 6

各節(ノード)の子が 2 つ以下である木構造のこと。

問 6

**参考** 数式  $a + (b - c) * d$  を、後置記法(逆ポーランド記法)に変換する作業だよ。



### 解答

- 問 4 **ウ**
- 問 5 **ウ**
- 問 6 **ウ**



## 問 7

空の状態のキューとスタックの二つのデータ構造がある。次の手続を順に実行した場合、変数  $x$  に代入されるデータはどれか。ここで、手続で引用している関数は、次のとおりとする。

〔関数の定義〕

push( $y$ ) : データ  $y$  をスタックに積む。

pop() : データをスタックから取り出して、その値を返す。

enq( $y$ ) : データ  $y$  をキューに挿入する。

deq() : データをキューから取り出して、その値を返す。

〔手続〕

push(a)

push(b)

enq(pop())

enq(c)

push(d)

push(deq())

$x \leftarrow \text{pop}()$

**ア** a

**イ** b

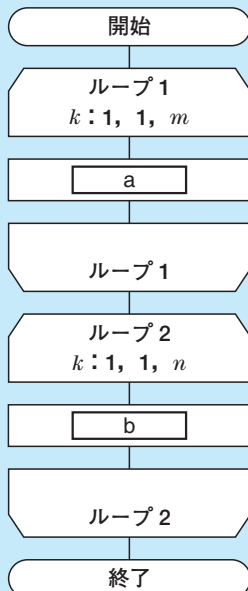
**ウ** c

**エ** d



## 問 8

長さ  $m$ ,  $n$  の文字列をそれぞれ格納した配列  $X$ ,  $Y$  がある。図は、配列  $X$  に格納した文字列の後ろに、配列  $Y$  に格納した文字列を連結したものを、配列  $Z$  に格納するアルゴリズムを表す流れ図である。図中の a, b に入れる処理として、適切なものはどれか。ここで、1 文字が一つの配列要素に格納されるものとする。

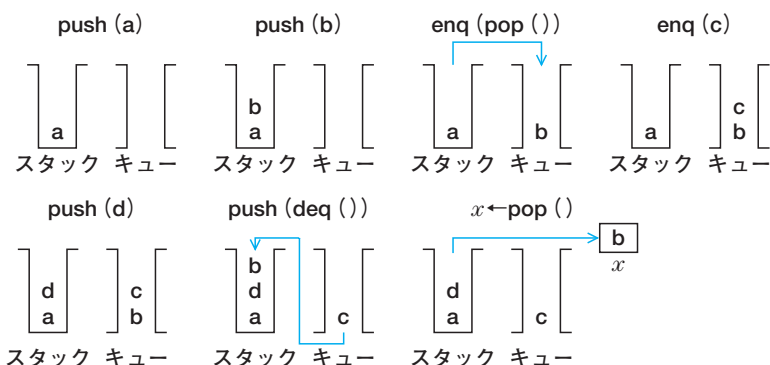


(注) ループ端の繰返し指定は、  
変数名：初期値，増分，終値  
を示す。

	a	b
<b>ア</b>	$X(k) \rightarrow Z(k)$	$Y(k) \rightarrow Z(m+k)$
<b>イ</b>	$X(k) \rightarrow Z(k)$	$Y(k) \rightarrow Z(n+k)$
<b>ウ</b>	$Y(k) \rightarrow Z(k)$	$X(k) \rightarrow Z(m+k)$
<b>エ</b>	$Y(k) \rightarrow Z(k)$	$X(k) \rightarrow Z(n+k)$

## 問 7 スタックとキュー キホン!

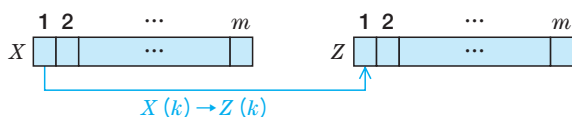
**スタック**は後入れ先出し（後に格納したものを先に取り出す）、**キュー**は先入れ先出し（先に格納したものを先に取り出す）のデータ構造です。問題文の手続にしたがって、スタックとキューの変化を調べてみましょう。なお、 $\text{enq}(\text{pop}())$ は、スタックから取り出した値をキューに挿入する操作です。



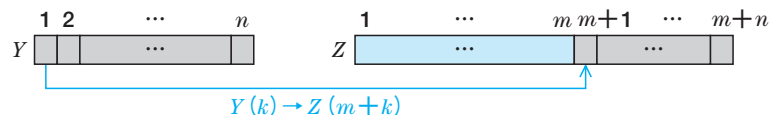
以上のように、変数  $x$  には  $b$  が代入されます。正解は **イ** です。

## 問 8 流れ図 キホン!

このアルゴリズムでは、はじめに配列  $X$  の内容を、 $X(1)$  から  $X(2)$ ,  $X(3)$  ...  $X(m)$  まで、1 つずつ配列  $Z$  に代入していきます。この作業は、変数  $k$  を 1 から  $m$  まで 1 ずつ増やしながら、 $X(k) \rightarrow Z(k)$  を繰り返し実行することで行えます。



次に、配列  $Y$  の内容を、配列  $Z$  に代入します。配列  $Z$  の  $Z(1) \sim Z(m)$  には、配列  $X$  の内容が代入されているので、代入先はその直後からになります。すなわち、 $Y(1)$  の内容を  $Z(m+1)$  に、 $Y(2)$  の内容を  $Z(m+2)$  に、...  $Y(n)$  の内容を  $Z(m+n)$  に代入します。この作業は、変数  $k$  を 1 から  $n$  まで 1 ずつ増やしながら、 $Y(k) \rightarrow Z(m+k)$  を繰り返し実行することで行えます。



以上から、流れ図の空欄 **a** には「 $X(k) \rightarrow Z(k)$ 」、空欄 **b** には「 $Y(k) \rightarrow Z(m+k)$ 」が入ります。正解は **ア** です。

## 覚えよう! 問 7

## キューといえば

- 格納した順番にデータを取り出すデータ構造 (FIFO: First-In First-Out)
- キューにデータを格納することを**エンキュー**、キューからデータを取り出すことを**デキュー**という

## スタックといえば

- 格納したデータを積み上げ、上から順に取り出すデータ構造 (LIFO: Last-In First-Out)
- スタックにデータを格納することを**プッシュ**、スタックからデータを取り出すことを**ポップ**という

## 解答

問 7 **イ** 問 8 **ア**



## 問 9

主記憶に記憶されたプログラムを，CPU が順に読み出しながら実行する方式はどれか。

- ア DMA 制御方式
- ウ 仮想記憶方式

- イ アドレス指定方式
- エ プログラム格納方式



## 問 10

主記憶のアクセス時間が 60 ナノ秒，キャッシュメモリのアクセス時間が 10 ナノ秒であるシステムがある。キャッシュメモリを介して主記憶にアクセスする場合の実効アクセス時間が 15 ナノ秒であるとき，キャッシュメモリのヒット率は幾らか。

ア 0.1

イ 0.17

ウ 0.83

エ 0.9



## 問 11

メモリモジュールのパリティチェックの目的として，適切なものはどれか。

- ア メモリモジュールに電源が供給されているかどうかを判定する。
- イ 読出し時に，エラーが発生したかどうかを検出する。
- ウ 読出し時に，エラーを検出して自動的に訂正する。
- エ 読み出したデータを暗号化する。



## 問 12

磁気ディスク装置の性能に関する記述のうち，適切なものはどれか。

- ア アクセス時間は，回転速度を上げるか位置決め時間を短縮すると短くなる。
- イ アクセス時間は，処理装置の前処理時間，データ転送後の後処理時間も含む。
- ウ 記憶容量は，トラック当たりの記憶容量と 1 シリンダ当たりのトラック数だけで決まる。
- エ データ転送速度は，回転速度と回転待ち時間で決まる。

### 解説

#### 問 9 プログラム格納方式

プログラムをあらかじめ主記憶装置内に記憶しておき，CPU が読み出して実行する方式を**プログラム格納方式**（プログラム記憶方式）といいます。現在のコンピュータの基本となっている方式です。

- × **ア DMA 制御方式**は，入出力装置と主記憶装置との間のデータ転送を，CPU を介せずに直接行う方式です。
- × **イ アドレス指定方式**は，CPU の命令で，主記憶装置上の位置を指定する方式のことです。
- × **ウ 仮想記憶方式**は，OS による記憶管理方式のひとつで，主記憶を仮想的なアドレスによって管理し，物理的な記憶容量より大きな主記憶空間を利用できるようにする方式です。

### 午前のカギ



○ **エ** 正解です。

## 問10 キャッシュメモリのヒット率 **よく出る!**

**キャッシュメモリのヒット率**とは、CPU が読み込もうとするデータが、キャッシュメモリに存在する確率です。実効アクセス時間とヒット率との関係は、次の式で表せます。

$$\text{実効アクセス時間} = c \times \text{ヒット率} + m \times (1 - \text{ヒット率})$$

$c$  : キャッシュメモリのアクセス時間  
 $m$  : 主記憶のアクセス時間

ヒット率を  $x$  とすれば、問題文より次の式が成り立ちます。

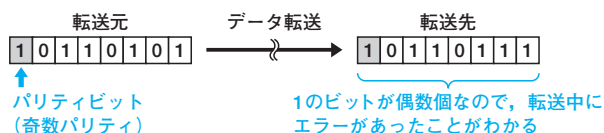
$$15 = 10x + 60 \times (1 - x) \quad \rightarrow \quad 50x = 45$$
$$\therefore x = 0.9$$

以上から、ヒット率は0.9になります。正解は **エ** です。

## 問11 パリティチェック **キホン!**

**パリティチェック**とは、データにパリティビットと呼ばれるビットを付加し、転送エラーを検出する方式です。

元データと合わせて、1のビットが奇数個になるようにパリティビットを付加する方式を奇数パリティ、偶数個になるように付加する方式を偶数パリティといいます。奇数パリティの場合、転送先で1のビットが偶数個になっていれば、転送エラーが発生していることがわかります。ただし、どのビットが誤りかまでは検出できないので、誤りの訂正はできません。以上から、正解は **イ** です。



## 問12 磁気ディスク装置の性能

磁気ディスク装置の**アクセス時間**は、①**位置決め時間**、②**回転待ち時間**、③**データ転送時間**の合計です。このうち、回転待ち時間はディスクが回転して目的のデータが磁気ヘッドの位置にくるまでの時間なので、回転速度が速いほど短くなります。

○ **ア** 正解です。

× **イ** 処理装置の前処理時間や後処理時間は、アクセス時間には含まれません。

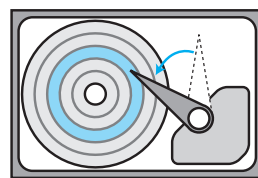
× **ウ** 磁気ディスク装置の記憶容量は、トラック当たりの記憶容量 × 1 シリンダ当たりのトラック数 × シリンダ数で決まります。

× **エ** データ転送速度は、ディスクからデータを読み取る時間なので、回転速度とトラック当たりの記憶容量で決まります。

## **午前のカギ**

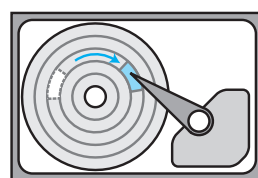
### **位置決め時間 (シーク時間)** 問12

磁気ヘッドが、目的のデータが記録されているトラックに移動する時間。



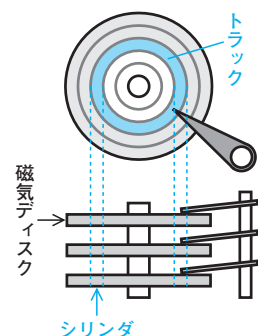
### **回転待ち時間 (サーチ時間)** 問12

ディスクが回転して目的のデータが磁気ヘッドの位置にくるまでの時間。最大で1回転、平均では1/2回転するのにかかる時間になる。



### **トラックとシリンダ** 問12

トラックは、磁気ディスクの同心円状の記録領域。また、複数の磁気ディスクの同一トラックをまとめてシリンダという。



### ° 解答 °

問9 **エ** 問10 **エ**  
問11 **イ** 問12 **ア**



**問 13** データベース (DB) へのアクセスを行うブラウザからのリクエスト処理を、Web サーバが受信し解釈した後に行う一連の実行処理の順序はどれか。ここで、Web サーバはリクエスト処理ごとに DB への接続と切断を行うものとする。

〔実行処理〕

- ① DB の切断
- ② DB へのアクセス
- ③ DB への接続
- ④ HTML 文書の組立て
- ⑤ SQL 文の組立て
- ⑥ ブラウザへの送信

**ア** ③, ②, ⑤, ④, ⑥, ①

**イ** ③, ④, ②, ⑤, ⑥, ①

**ウ** ③, ④, ⑤, ②, ⑥, ①

**エ** ③, ⑤, ②, ④, ⑥, ①



**問 14** スループットに関する記述のうち、適切なものはどれか。

- ア** ジョブの終了と次のジョブの開始との間にオペレータが介入することによってシステムに遊休時間が生じてても、スループットには影響を及ぼさない。
- イ** スループットは CPU 性能の指標であり、入出力の速度、オーバヘッド時間などによって影響を受けない。
- ウ** 多重プログラミングはターンアラウンドタイムの短縮に貢献するが、スループットの向上には役立たない。
- エ** プリンタへの出力を一時的に磁気ディスク装置に保存するスプーリングは、スループットの向上に役立つ。



**問 15** フェールセーフ設計の考え方に該当するものはどれか。

- ア** 作業範囲に人間が入ったことを検知するセンサが故障したとシステムが判断した場合、ロボットアームを強制的に停止させる。
- イ** 数字入力フィールドに数字以外のものが入力された場合、システムから警告メッセージを出力して正しい入力を要求する。
- ウ** 専用回線に障害が発生した場合、すぐに公衆回線に切り替え、システムの処理能力が低下しても処理を続行する。
- エ** データ収集システムでデータ転送処理に障害が発生した場合、データ入力処理だけを行い、障害復旧時にまとめて転送する。

### 問13 Web サーバのリクエスト処理

Web サーバは、ブラウザからのリクエストに応じて DB に問合せを行います。DB への問合せは、SQL と呼ばれる言語で行うので、ブラウザからのリクエストに合わせて、SQL 文を組み立てる必要があります。この SQL 文を使って DB にアクセスします。

DB は問合せ結果を Web サーバに戻します。Web サーバは、その結果を HTML 文書にして、ブラウザに送信します。ここまでの手順をまとめると、次のようになります。

⑤ SQL 文の組立て→② DB へのアクセス→④ HTML 文の組立て→⑥ブラウザへの送信

本問ではリクエスト処理ごとに DB への接続、切断を行うので、最初と最後に③ DB への接続と① DB の切断を行います。以上から、実行処理の順序は、③、⑤、②、④、⑥、①となります。正解は **エ** です。

### 問14 スループット よく出る!

コンピュータシステムの単位時間当たりの仕事量を **スループット** といいます。

- × **ア** 遊休時間が増えると単位時間当たりの仕事量は減るので、スループットは低下します。
- × **イ** 入出力の速度やオーバーヘッド時間もスループットに影響します。
- × **ウ** 多重プログラミングによって、複数のジョブを並行して処理できるようになれば、スループットの向上に役立ちます。
- **エ** 正解です。印刷待ちの間ジョブが中断してしまうと、スループットが低下します。したがって、出力データをいったん磁気ディスク装置に保存して、印刷が終わるのを待たずにジョブを再開できる **スプーリング** は、スループットの向上に役立ちます。

### 問15 フェールセーフ設計 よく出る!

**フェールセーフ** とは、障害が発生した場合には安全性を優先して、被害を最小限に抑える設計のことです。選択肢の中では、人間が立ち入るとロボットアームを強制停止させ、人間に危険が及ばないようにするのがフェールセーフ設計に当たります。

- **ア** 正解です。
- × **イ** 利用者の誤操作を防止したり、誤操作による異常が発生しないようにした設計を **フールプルーフ** といいます。
- × **ウ** 障害が発生したとき、処理能力が低下してもシステムが稼働し続けるようにした設計を **フェールソフト** といいます。
- × **エ** 障害が発生した部分を切り離して、残った部分だけで処理を続行するようにした設計を **フォールバック** といいます。

## 午前のカギ

問 13

**参考** 本問の選択肢にはないけど、①は、④や⑥の前に実行してもいいね。



問 14

#### 🔑 ターンアラウンドタイム

ジョブを投入してから結果の出力が終了するまでの時間。

問 14

#### 🔑 スプーリング

プリンタなど低速な装置への出力データを、いったん高速な記憶装置に格納しておくこと。

問 15

#### 🔑 覚えよう!

##### フェールセーフといえば

- システムが故障したときに安全性を優先

##### フールプルーフといえば

- 利用者が誤操作しても異常が生じないようにする

##### フェールソフトといえば

- システムが故障したときに運転の継続を優先

#### ◦ 解答 ◦

問 13 **エ** 問 14 **エ**  
問 15 **ア**



**問 16** ページング方式の仮想記憶を用いることによる効果はどれか。

- ア** システムダウンから復旧するときに、補助記憶のページを用いることによって、主記憶の内容が再現できる。
- イ** 処理に必要なページを動的に主記憶に割り当てることによって、主記憶を効率的に使用できる。
- ウ** 頻繁に使用されるページを仮想記憶に置くことによって、アクセス速度を主記憶へのアクセスよりも速めることができる。
- エ** プログラムの大きさに応じて大小のページを使い分けることによって、主記憶を無駄なく使用できる。



**問 17** データ管理ユーティリティの一つである、アーカイバの機能を説明したものはどれか。

- ア** 磁気ディスクに、データを記録するための領域と、それを管理するための領域を作成する。
- イ** データのバックアップや配布のために、複数のファイルを一つにまとめたり、元に戻したりする。
- ウ** 不正使用や破壊からデータを守るファイルプロテクトや、不正コピー防止のためのコピープロテクトなどによって、データを保護する。
- エ** フラグメンテーションが発生した磁気ディスクで、ファイルを可能な限り連続した領域に再配置する。



**問 18** コンパイラにおける最適化の説明として、適切なものはどれか。

- ア** オブジェクトコードを生成する代わりに、インタプリタ用の中間コードを生成する。
- イ** コンパイルを実施するコンピュータとは異なるアーキテクチャをもったコンピュータで動作するオブジェクトコードを生成する。
- ウ** ソースコードを解析して、実行時の処理効率を高めたオブジェクトコードを生成する。
- エ** プログラムの実行時に、呼び出されたサブプログラム名やある時点での変数の内容を表示するようなオブジェクトコードを生成する。



**問 19** 静的テストツールの機能に分類されるものはどれか。

- ア** ソースコードを解析して、プログラムの誤りを検出する。
- イ** テスト対象モジュールに必要なドライバ又はスタブを生成する。
- ウ** テストによって実行した経路から網羅度を算出する。
- エ** プログラムの特定の経路をテストするためのデータを生成する。

## 問16 ページング方式 キホン!

ページング方式の仮想記憶では、記憶領域を一定サイズのページに分割し、必要なページを主記憶に読み込んだり、ハードディスクに退避させたりします。これにより、物理的な容量より大きな主記憶容量を効率的に利用できるようになります。

- × ア システムダウンによって主記憶の内容は消えてしまいます。退避されているページが補助記憶に残っていても、システムの復旧には使用できません。
- イ 正解です。
- × ウ キャッシュメモリの説明です。
- × エ セグメント方式の仮想記憶の説明です。ページング方式では、一般にページは固定長になります。

## 問17 アーカイバ

複数のファイルを1つにまとめたファイルを「アーカイブ（書庫）」といいます。アーカイバは、アーカイブを作成したり、アーカイバに格納されているファイルを元に戻したりするユーティリティソフトです。

- × ア フォーマッタの説明です。
- イ 正解です。
- × ウ コピープロテクタの説明です。
- × エ デフラグツールの説明です。

## 問18 コンパイラの最適化 キホン!

コンパイラは、プログラム言語で記述されたソースコードを、CPUが解釈できる機械語のオブジェクトコードに変換するソフトです。コンパイラにおける最適化とは、ソースコードを解析して冗長な部分を取り除き、より処理効率の高いオブジェクトコードに変換する処理をいいます。

- × ア バイトコンパイルの説明です。
- × イ クロスコンパイルの説明です。
- ウ 正解です。
- × エ トレーサの説明です。

## 問19 静的テストツール

テスト対象となるプログラムを実行せずに行うテストを「静的テスト」といいます。静的テストツールは、主にソースコードの解析からプログラムの構造や問題点を分析します。正解は「ア」です。

静的テストに対して、実際にプログラムを実行して行うテストを「動的テスト」といいます。ドライバやスタブを生成したり、網羅度を算出したり、テストデータを生成するツールは、いずれも動的テストを支援するツールです。

## 仮想記憶

問16

主記憶上にある実記憶領域の一部を必要に応じてハードディスクなどに退避させることで、物理的な容量より大きな主記憶領域を利用できるようにする仕組み。

参考 PCではファイル圧縮機能のあるアーカイバが一般的だけど、ファイル圧縮機能とアーカイブ機能は本来別のものなんだ。



## 覚えよう!

問19

静的テストといえば

- プログラムを実行せずに行うテスト
- ソースコードを解析してプログラムの誤りを検出

## 解答

- |     |   |     |   |
|-----|---|-----|---|
| 問16 | イ | 問17 | イ |
| 問18 | ウ | 問19 | ア |



**問 20** メモリセルにフリップフロップ回路を利用したものはどれか。

- ア DRAM      イ EEPROM      ウ SDRAM      エ SRAM

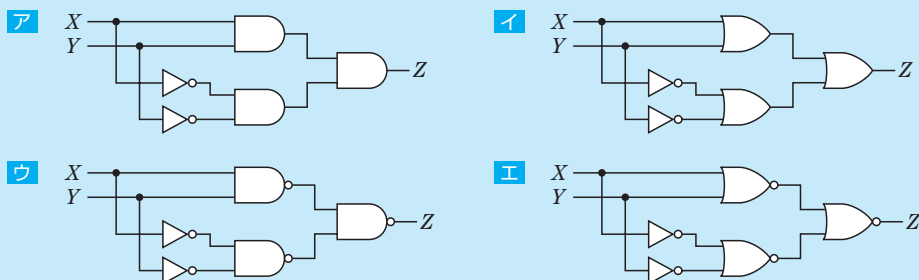


**問 21** フラッシュメモリに関する記述として、適切なものはどれか。

- ア 高速に書換えができ、CPU のキャッシュメモリなどに用いられる。  
 イ 紫外線で全内容の消去ができる。  
 ウ 周期的にデータの再書き込みが必要である。  
 エ ブロック単位で電氣的に内容の消去ができる。



**問 22** 入力  $X$  と  $Y$  の値が同じときにだけ、出力  $Z$  に 1 を出力する回路はどれか。



**問 23** ヒューマンインタフェースの機能のうち、定型的な作業を頻繁に行う利用者の操作数を少なくする目的で用いるものはどれか。

- ア 一連のコマンドをひとまとめにしたマクロ機能  
 イ 最後の画面でまとめて入力エラーを表示する機能  
 ウ 全工程のうち、現在どこまで進んでいるかを表示する機能  
 エ 操作を誤ったときに前の状態に戻ることができる機能

### 解説

#### 問20 メモリの種類 よく出る!

**SRAM** (Static RAM) は電源を切ると内容が消えてしまう揮発性メモリの一種で、リフレッシュ動作が不要で高速なのが特徴です。ただし、DRAMと比較して高価なので、主に容量の少ないキャッシュメモリに用いられています。

× **ア DRAM** (Dynamic RAM) は、SRAMと同じく揮発性のメモリの一種で、内容を保持するために定期的なリフレッシュ動作が必要です。構造が単純なので大容量化・高集積化が簡単にでき、主に主記憶装置に用いられています。

### 午前のカギ

#### フリップフロップ回路

問 20

異なる信号を入力するまで 0 または 1 の状態 (2 つの安定状態) を保持することができる回路。SRAM のメモリセルに利用されている。

- × **イ** EEPROM は、データの消去や書き込みが可能な不揮発性メモリの一種で、主に電源を切った後もデータを保持しておく設定情報などの記録に用いられます。
- × **ウ** SDRAM (Synchronous DRAM) は、クロック信号に同期して動作する DRAM で、非同期の DRAM より高速に動作します。
- **エ** 正解です。

## 問21 フラッシュメモリ よく出る!

**フラッシュメモリ**は、電源を切っても内容が消えない不揮発性のメモリの一種で、内容を電氣的に消去・書き込みできるのが特徴です。USB メモリや SSD、SD カードなど、記憶媒体として幅広く利用されています。

- × **ア** SRAM の説明です。
- × **イ** EPROM の説明です。
- × **ウ** DRAM の説明です。
- **エ** 正解です。

## 問22 論理回路 キホン!

$X$  と  $Y$  が両方とも 1 か、 $X$  と  $Y$  が両方とも 0 のときだけ 1 になる論理回路を論理式で表すと、次のようになります。

$$\frac{(X \cdot Y) + (\bar{X} \cdot \bar{Y})}{\text{両方とも1のとき1 両方とも0のとき1}}$$

選択肢の論理回路を論理式で表し、上の論理式と同じになるものを選びます。

$\square$  は論理積 ( $\cdot$ )、 $\square$  は論理和 ( $+$ )、 $\square$  は否定論理積 (NAND)、 $\square$  は否定論理和 (NOR)、 $\neg$  は論理否定 (NOT) を表します。

- × **ア**  $(X \cdot Y) \cdot (\bar{X} \cdot \bar{Y})$
- × **イ**  $(X + Y) + (\bar{X} + \bar{Y})$

$$\begin{aligned} \text{○ **ウ** } & \overline{(X \cdot Y) \cdot (\bar{X} \cdot \bar{Y})} = \overline{(X \cdot Y)} + \overline{(\bar{X} \cdot \bar{Y})} = (X + Y) + (\bar{X} + \bar{Y}) \\ & \quad \text{ド・モルガンの法則} \end{aligned}$$

$$\begin{aligned} \times \text{ **エ** } & \overline{(X + Y) + (\bar{X} + \bar{Y})} = \overline{(X + Y)} \cdot \overline{(\bar{X} + \bar{Y})} = (X \cdot Y) \cdot (\bar{X} \cdot \bar{Y}) \\ & \quad \text{ド・モルガンの法則} \end{aligned}$$

## 問23 ヒューマンインタフェース

頻繁に行われる定型的な作業を簡略化するには、アプリケーションソフトが備える**マクロ機能**を利用します。マクロとは、複数のコマンドをひとまとめにしたり、簡単なマクロプログラムを記述して一連の操作を 1 つの操作にまとめたものです。マクロにすることで、操作数を大幅に少なくできます。

- **ア** 正解です。
- × **イ** 入力エラーが最後までわからず、かえって操作しにくくなります。
- × **ウ** 作業の進捗状況は把握しやすくなりますが、操作数を少なくすることにはなりません。
- × **エ** 誤操作からの復旧はしやすくなりますが、操作数は少なくなります。

## 午前のカギ

### 覚えよう! 問 20

#### SRAM といえば

- フリップフロップ回路を利用
- リフレッシュ動作が不要で高速
- キャッシュメモリに用いる

#### DRAM といえば

- 構造が単純で安価
- リフレッシュ動作が必要
- 主記憶装置に用いる

### リフレッシュ 問 20

DRAM が内容を保持するために、記憶セルの内容を定期的に再書き込みすること。

### EPROM 問 21

不揮発性メモリの一種で、紫外線を照射して内容を消去できる。

### 否定論理積 (NAND) 問 22

論理積 (AND) の否定。X NAND Y =  $\overline{X \text{ AND } Y}$  となる。

### 否定論理和 (NOR) 問 22

論理和 (OR) の否定。X NOR Y =  $\overline{X \text{ OR } Y}$  となる。

### 覚えよう! 問 22

#### ド・モルガンの法則

といえば

$$\begin{aligned} \overline{X \cdot Y} &= \bar{X} + \bar{Y} \\ \overline{X + Y} &= \bar{X} \cdot \bar{Y} \end{aligned}$$

#### 解答

- 問20 **エ** 問21 **エ**
- 問22 **ウ** 問23 **ア**





**問 24** アニメーションの作成過程で、センサやビデオカメラなどを用いて人間や動物の自然な動きを取り込む技法はどれか。

- ア キーフレーム法
- ウ モーションキャプチャ

- イ ピクセルシェーダ
- エ モーフィング



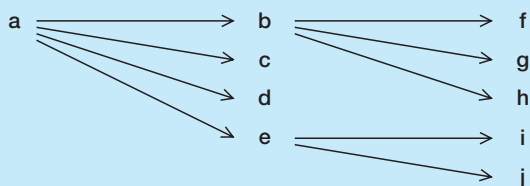
**問 25** SQL 文を実行する際に、効率が良いと考えられるアクセス経路を選択する関係データベース管理システム (RDBMS) の機能はどれか。

- ア オプティマイザ
- ウ クラスタリング

- イ ガーベジコレクション
- エ マージソート



**問 26** 項目 a の値が決まれば項目 b の値が一意に定まることを、 $a \rightarrow b$  で表す。例えば、社員番号が決まれば社員名が一意に定まるという表現は、社員番号  $\rightarrow$  社員名である。この表記法に基づいて、図の関係が成立している項目  $a \sim j$  を、関係データベース上の三つのテーブルで定義する組合せとして、適切なものはどれか。



- ア テーブル 1 (a)  
テーブル 2 (b, c, d, e)  
テーブル 3 (f, g, h, i, j)

- イ テーブル 1 (a, b, c, d, e)  
テーブル 2 (b, f, g, h)  
テーブル 3 (e, i, j)

- ウ テーブル 1 (a, b, f, g, h)  
テーブル 2 (c, d)  
テーブル 3 (e, i, j)

- エ テーブル 1 (a, c, d)  
テーブル 2 (b, f, g, h)  
テーブル 3 (e, i, j)



## 問24 アニメーション技法

身体に取り付けたセンサやビデオカメラによって、身体の動きを記録し、これをもとにリアルな動きの動画を作成する技法を、**モーションキャプチャ**といいます。

- × **ア キーフレーム法**とは、動画を構成する複数の画像（フレーム）のうち、いくつかをキーフレームに設定し、その中間の画像を補間計算によって生成する手法です。
- × **イ ピクセルシェーダ**とは、三次元 CG で陰影の処理（シェーディング）を行う機能の一種で、画像をピクセル単位で処理する方式です。
- **ウ** 正解です。
- × **エ モーフィング**とは、ある画像から別の画像へと変化する過程の画像を複数作成し、画像が徐々に変化していく様子を表す技術です。

## 問25 RDBMS の機能 初モノ!

SQL 文を高速に実行するために、最も効率のよいアクセスパス（表から目的のデータを取り出す経路）を選択する **RDBMS** の機能を**オプティマイザ**といいます。

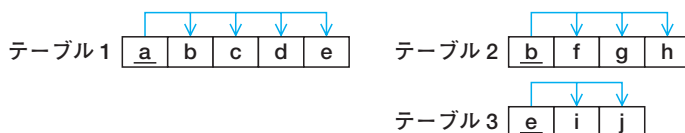
- **ア** 正解です。
- × **イ ガーベジコレクション**とは、プログラムが確保したメモリのうち、未使用になった領域を自動的に回収して、再び利用可能にする機能です。
- × **ウ クラスタリング**とは、多数の PC を接続して 1 つのコンピュータのように動作させ、高い処理能力を得る技術です。
- × **エ マージソート**は、整列アルゴリズムの一種です。

## 問26 データベースの設計

**関係データベース**のテーブルを構成する各項目のうち、行を特定できる値をもつ項目を**主キー**といいます。関係データベースのテーブルには、必ず主キーが含まれます。

問題文の図によれば、項目 a の値が決まれば項目 b, c, d, e の値が一意に決まります。ここから、a を主キーとし、b, c, d, e を他の項目とするテーブルを作成できます。

同様に、項目 b の値が決まれば項目 f, g, h の値が一意に決まり、項目 e の値が決まれば項目 i, j の値が一意に決まります。ここから、主キーが b で f, g, h を他の項目とするテーブルと、主キーが e で i, j を他の項目とするテーブルを作成できます。



以上から、正解は**イ**です。

**RDBMS** 問 25  
 Relational Database Management System : 関係データベース管理システムの略。データベースを表形式で管理し、利用者は SQL 文によってデータベースにアクセスする。現在普及している DBMS（データベース管理システム）の多くは RDBMS である。

**マージソート** 問 25  
 データを細かい単位に分割して並べ替え、それらを併合して最終的に全体を整列させる整列アルゴリズム。分割・整列・併合を再帰的に実行する。

**覚えよう!** 問 26  
**関係データベース** といえば  
 ●データの集まりを 2 次元の表で表す

° 解答 °

問24 **ウ** 問25 **ア**  
 問26 **イ**

問 27 関係データベースの操作のうち、射影（projection）の説明として、適切なものはどれか。

- ア ある表の照会結果と、別の表の照会結果を合わせて一つの表にする。
- イ 表の中から特定の条件に合致した行を取り出す。
- ウ 表の中から特定の列だけを取り出す。
- エ 二つ以上の表の組から条件に合致した組同士を合わせて新しい表を作り出す。

問 28 “商品” 表，“在庫” 表に対する次の SQL 文の結果と同じ結果が得られる SQL 文はどれか。ここで、下線部は主キーを表す。

```
SELECT 商品番号 FROM 商品
WHERE 商品番号 NOT IN (SELECT 商品番号 FROM 在庫)
```

商品			在庫		
<u>商品番号</u>	商品名	単価	<u>倉庫番号</u>	<u>商品番号</u>	在庫数

- ア SELECT 商品番号 FROM 在庫  
WHERE EXISTS (SELECT 商品番号 FROM 商品)
- イ SELECT 商品番号 FROM 在庫  
WHERE NOT EXISTS (SELECT 商品番号 FROM 商品)
- ウ SELECT 商品番号 FROM 商品  
WHERE EXISTS (SELECT 商品番号 FROM 在庫  
WHERE 商品.商品番号 = 在庫.商品番号)
- エ SELECT 商品番号 FROM 商品  
WHERE NOT EXISTS (SELECT 商品番号 FROM 在庫  
WHERE 商品.商品番号 = 在庫.商品番号)

問 29 トランザクションが、データベースに対する更新処理を完全に行うか、全く処理しなかったかのように取り消すか、のどちらかの結果になることを保証する特性はどれか。

- ア 一貫性（consistency）
- イ 原子性（atomicity）
- ウ 耐久性（durability）
- エ 独立性（isolation）

解説

## 問27 関係演算 よく出る！

関係データベースに対する操作には、選択、射影、結合などの操作があります。このうち射影（projection）は、表から特定の列を得る操作です。

🔑 午前のカギ

- × **ア** 併合の説明です。
- × **イ** 選択の説明です。
- **ウ** 正解です。
- × **エ** 結合の説明です。

## 問28 SQL (相関副問合せ)

問題のSQL文では、まず副問合せ「SELECT 商品番号 FROM 在庫」で、“在庫”表に登録されている商品番号がリストアップされます。次に主問合せのNOT IN 述語によって、その中に含まれていない商品番号が“商品”表から選択されます。要するに、問題のSQL文は「在庫がない商品の商品番号」を表示するものです。

選択枝のSQL文には、いずれもEXISTS 述語が使われています。また、

- ウ**と**エ**は相関副問合せを使ったSQL文です。
- × **ア** “商品”表に1つ以上の商品が登録されていればEXISTS が真になり、“在庫”表の商品番号一覧を表示します。
- × **イ** “商品”表に商品が1つも登録されてないときNOT EXISTS が真になり、“在庫”表の商品番号一覧を表示します。
- × **ウ** 副問合せでは、まず“商品”表に登録されている商品番号が“在庫”表にあるかどうかを1つずつ調べ、見つかった場合はその商品番号を返します。するとEXISTS が真になるので、主問合せは該当する商品番号を表示します。SQL文は、結果として「在庫のある商品の商品番号」を表示します。
- **エ** 副問合せでは、まず“商品”表に登録されている商品番号が“在庫”表にあるかどうかを1つずつ調べます。見つからなかった場合はNOT EXISTS 述語が真になるので、主問合せは該当する商品番号を表示します。SQL文は、結果として「在庫がない商品の商品番号」を表示します。

## 問29 ACID 特性 よく出る!

DBMS がデータベースの更新処理について保証すべき4つの特性(原子性、一貫性、独立性、耐久性)を、まとめて**ACID 特性**といいます。トランザクションの結果が、データベースに対する更新処理を完全に行うか、まったく処理しないかのどちらかになることを保証する特性は、**原子性**(atomicity)です。

- × **ア** **一貫性**(consistency)は、トランザクションの実行前後で、データベースの整合性を維持することを保証します。
- **イ** 正解です。
- × **ウ** **耐久性**(durability)は、いったんトランザクションが完了した後は、その結果が失われないことを保証します。
- × **エ** **独立性**(isolation)は、トランザクション実行中の操作の過程が外部から隠ぺいされ、他の処理の干渉を受けないことを保証します。

## 午前のカギ

### 覚えよう! 問27

#### 関係演算といえば

- 射影：特定の列を取り出す
- 選択：特定の行を取り出す
- 結合：2つの表を共通の項目をキーにして結合する

### EXISTS 述語 問28

副問合せの結果が1件でも存在すれば真、1件もなければ偽を返す。NOT EXISTS はその逆になる。

### 相関副問合せ 問28

主問合せに指定されている表の各行について、1件ずつ副問合せを実行し、その結果によってその行を選択するかどうかを決定するSQLの問合せ方式。

### 覚えよう! 問29

#### ACID 特性といえば

- 原子性 (atomicity)
- 一貫性 (consistency)
- 独立性 (isolation)
- 耐久性 (durability)

### トランザクション 問29

データベースの更新処理の単位。1つのトランザクション中には、複数の表の更新が含まれる場合もある。

#### 解答

- 問27 **ウ** 問28 **エ**  
問29 **イ**

問30 OSI 基本参照モデルの各層で中継する装置を、物理層で中継する装置、データリンク層で中継する装置、ネットワーク層で中継する装置の順に並べたものはどれか。

ア ブリッジ、リピータ、ルータ  
ウ リピータ、ブリッジ、ルータ

イ ブリッジ、ルータ、リピータ  
エ リピータ、ルータ、ブリッジ

問31 TCP/IP ネットワークで DNS が果たす役割はどれか。

- ア PC やプリンタなどからの IP アドレス付与の要求に対し、サーバに登録してある IP アドレスの中から使用されていない IP アドレスを割り当てる。  
イ サーバにあるプログラムを、サーバの IP アドレスを意識することなく、プログラム名の指定だけで呼び出すようにする。  
ウ 社内のプライベート IP アドレスをグローバル IP アドレスに変換し、インターネットへのアクセスを可能にする。  
エ ドメイン名やホスト名などと IP アドレスとを対応付ける。

問32 IPv6 アドレスの特徴として、適切なものはどれか。

- ア アドレス長は 96 ビットである。  
イ 全てグローバルアドレスである。  
ウ 全ての IPv6 アドレスと IPv4 アドレスを、1 対 1 に対応付けることができる。  
エ 複数のアドレス表記法があり、その一つは、アドレスの 16 進数表記を 4 文字（16 ビット）ずつコロン “:” で区切る方法である。

問33 インターネットにおける電子メールの規約で、ヘッダフィールドの拡張を行い、テキストだけでなく、音声、画像なども扱えるようにしたものはどれか。

ア HTML

イ MHS

ウ MIME

エ SMTP

### 解説

#### 問30 LAN 間接接続装置 よく出る!

OSI 基本参照モデルは、ネットワークプロトコルを 7 つの階層に分類し、それぞれの役割を規定したものです。ネットワークを中継する装置は、OSI 基本参照モデルのどの層のデータを中継するかによって分類できます。

- ① **リピータ**はケーブルの伝送距離を延長するために、信号を**物理層**で中継します。  
② **ブリッジ**は MAC アドレスを基に、**データリンク層**でフレームを中継します。

### 🔑 午前のカギ

#### 🔑 物理層

問30

OSI 基本参照モデルの第 1 層。信号を物理的に伝送する。

- ③ **ルータ**はIPアドレスを基に、**ネットワーク層**でパケットを中継します。  
以上から、**ウ**の「リピータ、ブリッジ、ルータ」の順が正解です。

### 問31 DNSの役割 **キホン!**

**DNS** (Domain Name System) は、TCP/IP ネットワークの通信相手を、IP アドレスの代わりにドメイン名と呼ばれる名前指定できるようにする仕組みです。DNS サーバは、IP アドレスとドメイン名との対応付けを行うデータベースをもち、ブラウザなどからの問合せに対して、対応する IP アドレスやドメイン名を返す役割を果たします。

- × **ア** **DHCP** (Dynamic Host Configuration Protocol) の説明です。
- × **イ** **RPC** (Remote Procedure Call) で用いられるネームサービスの説明です。
- × **ウ** **NAT** (Network Address Translation) の説明です。
- **エ** 正解です。

### 問32 IPv6アドレスの特徴

**IPv6** (IP バージョン 6) は、IP の次世代バージョンです。現在広く利用されている **IPv4** (IP バージョン 4) と比較して、アドレスの長さやセキュリティ機能などが改良されています。

**IPv6** アドレスの長さは 128 ビットあります。表記する場合は 16 ビットごとにコロン (:) で区切り、「2001:0db8:0000:0002:1000:0100:0020:0002」のような 16 進数の並びで表記するのが一般的です。

- × **ア** **IPv6** アドレスのアドレス長は 128 ビットです。
- × **イ** インターネット上で一意に割り当てられるグローバルアドレス以外に、**IPv4** のプライベートアドレスに相当するアドレス (ユニークローカルユニキャストアドレス) が用意されています。
- × **ウ** **IPv6** アドレスの数は **IPv4** アドレスよりはるかに多いので、両者を 1 対 1 に対応付けることはできません。
- **エ** 正解です。

### 問33 電子メールの規約 **よく出る!**

インターネットの電子メールで、音声や画像ファイルなどの様々な形式のデータをやり取りするための規約を **MIME** (Multipurpose Internet Mail Extension) といいます。

- × **ア** **HTML** (HyperText Markup Language) は、インターネットの Web ページを記述するためのマークアップ言語です。
- × **イ** **MHS** (Message Handling System) は、OSI 基本参照モデルに基づく電子メールサービスの国際規格です。
- **ウ** 正解です。
- × **エ** **SMTP** (Simple Mail Transfer Protocol) は、インターネットの電子メールの送信プロトコルです。

## 午前のカギ

### データリンク層 **問30**

OSI 基本参照モデルの第 2 層。隣り合うノード間でデータ (フレーム) を伝送する。

### ネットワーク層 **問30**

OSI 基本参照モデルの第 3 層。ネットワークを相互に中継し、端末間でデータ (パケット) を伝送する。

### 覚えよう! **問30**

#### OSI基本参照モデルとLAN間接続装置

OSI基本参照モデル	LAN間接続装置
第4～7層 その他	ゲートウェイ
第3層 ネットワーク層	ルータ
第2層 データリンク層	ブリッジ、 レイヤ2スイッチ
第1層 物理層	リピータハブ、 リピータ

### DHCP **問31**

ネットワークに接続した端末に IP アドレスを自動的に割り当てるプロトコル。

### IPv6 **問32**

ネットワークでは、ネットワークに接続する端末ごとに固有のアドレスを割り当てる。**IPv6** では IP アドレスの長さを 128 ビットに拡張し、割当て可能なアドレスを一気に  $2^{128}$  個に増やした。

### 覚えよう! **問33**

#### MIME といえば

- 電子メールで音声、動画などをやり取りするための規格
- データを文字データに変換して送信。このときの符号化方式を Base64 という

#### 解答

- 問30 **ウ** 問31 **エ**  
問32 **エ** 問33 **ウ**



**問 34** IP ネットワークにおいて、ICMP のエコー要求、エコー応答、到達不能メッセージなどによって、通信相手との接続性を確認するコマンドはどれか。

**ア** arp

**イ** echo

**ウ** ipconfig

**エ** ping



**問 35** IPv4 で 192.168.30.32/28 のネットワークに接続可能なホストの最大数はどれか。

**ア** 14

**イ** 16

**ウ** 28

**エ** 30



**問 36** メッセージ認証符号におけるメッセージダイジェストの利用目的はどれか。

**ア** メッセージが改ざんされていないことを確認する。

**イ** メッセージの暗号化方式を確認する。

**ウ** メッセージの概要を確認する。

**エ** メッセージの秘匿性を確保する。



**問 37** PKI（公開鍵基盤）の認証局が果たす役割はどれか。

**ア** 共通鍵を生成する。

**イ** 公開鍵を利用しデータの暗号化を行う。

**ウ** 失効したデジタル証明書の一覧を発行する。

**エ** データが改ざんされていないことを検証する。





## 問34 ネットワークコマンド

ICMP (Internet Control Message Protocol) というプロトコルの機能を利用して、IP ネットワークを介して通信相手との接続性を確認するのは、ping コマンドです。

- × ア arp は、IP アドレスと MAC アドレスとの対応付けを管理するコマンドです。
- × イ echo は、指定した文字列を標準出力に出力するコマンドです。
- × ウ ipconfig は、IP アドレスの割当てや DNS サーバのアドレスなど、現在の IP の設定情報を調べるコマンドです。
- エ 正解です。

## 問35 IPv4 アドレス キホン!

ネットワークアドレス「192.168.30.32/28」の「/28」は、このアドレスの上位 28 ビットがネットワーク部であることを示します。したがって、32 ビットのうち、ホストに割り当てられるのは下位 4 ビットだけです。



4 ビットのホスト部には、0000 ~ 1111 まで 16 個のアドレスが割当て可能です。ただし、ホスト部の全ビットが 0 のアドレスと、全ビットが 1 のアドレスは、ホストに割り当てることができません。したがって、接続可能なホストは最大 14 個になります。以上から、正解はアです。

## 問36 メッセージダイジェストの利用目的 キホン!

メッセージダイジェストは、元のメッセージから生成される短い符号です。送信者がメッセージにメッセージダイジェストを添付して送信すると、受信者は受け取ったメッセージから同じようにメッセージダイジェストを生成し、それを添付されたものと照合します。両者が一致すれば、そのメッセージは途中で改ざんされていないことがわかります。以上から正解はアです。

## 問37 認証局の役割 キホン!

認証局 (CA) は、公開鍵が本人のものであることを証明するデジタル証明書を発行しています。この証明書はデジタル署名に付属して、署名が偽造でないことの保証になります。

ところで、いったん発行されたデジタル証明書が、何らかの理由で失効した場合、そのデジタル証明書で署名の正当性を保証することはできません。そのために認証局は、失効したデジタル証明書の一覧を発行して、失効したデジタル証明書を利用できないようにしています。この一覧を CRL (証明書失効リスト) といいます。以上から、正解はウです。

## ICMP

問 34

IP の送信エラーの報告やネットワークの接続テストに使われるプロトコル。

## メッセージ認証符号 (MAC)

問 36

メッセージが改ざんされていないかどうかを確認するための短いデータ。メッセージダイジェストを共通鍵暗号方式で暗号化したものである。

## PKI (公開鍵基盤)

問 37

公開鍵の正当性を認証局が保証する仕組み。この仕組みにより、公開鍵暗号方式による暗号化やデジタル署名などを安心して利用できる。

## CRL (証明書失効リスト)

問 37

失効したデジタル証明書のリスト。デジタル証明書が失効する理由としては、秘密鍵が漏えいした場合 (対になる公開鍵が使用できなくなる) や、証明書を不正に取得したことが判明した場合などがある。

## 解答

- 問34 エ 問35 ア  
問36 ア 問37 ウ



**問 38** IC カードと PIN を用いた利用者認証における適切な運用はどれか。

- ア** IC カードによって個々の利用者を識別できるので、管理負荷を軽減するために全利用者に共通の PIN を設定する。
- イ** IC カードの表面に刻印してある数字情報を組み合わせて、PIN を設定する。
- ウ** IC カード紛失時には、新たな IC カードを発行し、PIN を再設定した後で、紛失した IC カードの失効処理を行う。
- エ** IC カードを配送する場合には、PIN を同封せず、別経路で利用者に知らせる。



**問 39** 情報セキュリティにおける“完全性”を脅かす攻撃はどれか。

- ア** Web ページの改ざん
- イ** システム内に保管されているデータの不正コピー
- ウ** システムを過負荷状態にする DoS 攻撃
- エ** 通信内容の盗聴



**問 40** 会社や団体が、自組織の従業員に貸与するスマートフォンに対して、セキュリティポリシーに従った一元的な設定をしたり、業務アプリケーションを配信したりして、スマートフォンの利用状況などを一元管理する仕組みはどれか。

- ア** BYOD (Bring Your Own Device)
- イ** ECM (Enterprise Contents Management)
- ウ** LTE (Long Term Evolution)
- エ** MDM (Mobile Device Management)



**問 41** 緊急事態を装って組織内部の人間からパスワードや機密情報を入手する不正な行為は、どれに分類されるか。

- |                        |                 |
|------------------------|-----------------|
| <b>ア</b> ソーシャルエンジニアリング | <b>イ</b> トロイの木馬 |
| <b>ウ</b> パスワードクラック     | <b>エ</b> 踏み台攻撃  |

解説

**問 38** IC カードと PIN

キャッシュカードなどの **IC カード**は、利用者本人以外使用できないようにするため、利用時に暗証番号を入力します。この暗証番号を **PIN** (Personal Identification Number) といいます。

IC カードが不正に利用されるのを防ぐため、暗証番号は第三者に知られないようにします。



午前のカギ



- × **ア** PIN は本人だけが知っている秘密の暗証番号にするべきです。
- × **イ** PIN は第三者が簡単に推測できないようにします。
- × **ウ** 紛失した IC カードの失効処理を最優先で行うべきです。
- **エ** 正解です。IC カードと PIN が同時に盗まれると簡単に不正利用されてしまうので、別経路で配送するのは適切な運用です。

### 問39 完全性 よく出る！

機密性、完全性、可用性は、情報セキュリティの三大要素です。このうち**完全性**とは、情報が正確で、改ざんや破壊がないことをいいます。

- **ア** 正解です。Web ページの改ざんは、完全性を脅かす攻撃です。
- × **イ** 不正コピーは、機密性を脅かす攻撃です。
- × **ウ** DoS 攻撃は、可用性を脅かす攻撃です。
- × **エ** 盗聴は、機密性を脅かす攻撃です。

### 問40 携帯端末の管理 初モノ！

スマートフォンやタブレット端末といった高性能な携帯端末が普及するにつれ、それらを業務に利用する機会も増えてきました。それにともない、これらの携帯端末の利用状況を管理したり、情報セキュリティ対策なども必要になっています。これらを実現する仕組みを **MDM** (Mobile Device Management) といいます。

- × **ア** **BYOD** (Bring Your Own Device) は、従業員が、自分の私物である情報端末を職場に持ち込み、業務に使用することです。
- × **イ** **ECM** (Enterprise Contents Management) は、組織内のあらゆる文書やコンテンツをまとめて一元管理することです。
- × **ウ** **LTE** (Long Term Evolution) は、3G (第3世代携帯電話) に代わって現在普及がすすんでいる携帯電話の高速通信規格です。
- **エ** 正解です。

### 問41 攻撃の手口 よく出る！

緊急事態を装ったり、会話を立ち聞きするといった、人間の心理や油断をついて機密情報を入手する手口を**ソーシャルエンジニアリング**といいます。

- **ア** 正解です。
- × **イ** **トロイの木馬**とは、無害なファイルを装ってシステム内に保存され、利用者が実行すると様々な不正を行うプログラムです。
- × **ウ** **パスワードクラック**とは、パスワードを不正に入手する攻撃です。可能な組合せをすべて試す総当たり攻撃や、予想される字句を優先して試す辞書攻撃などがあります。
- × **エ** **踏み台攻撃**は、不正侵入したサーバを中継して、攻撃対象を攻撃する手口です。

### 🔑 覚えよう！ 問39

#### 情報セキュリティのCIA

- **機密性 (Confidentiality)** : 機密性が保たれること
- **完全性 (Integrity)** : 情報が正確なこと
- **可用性 (Availability)** : 必要なときに利用できること

### 🔑 覚えよう！ 問41

#### ソーシャルエンジニアリングといえば

- 人間の心理の隙を突いて機密情報を入手する手口
- 緊急事態を装ってパスワードを聞き出す

#### ○ 解答 ○

- |     |          |     |          |
|-----|----------|-----|----------|
| 問38 | <b>エ</b> | 問39 | <b>ア</b> |
| 問40 | <b>エ</b> | 問41 | <b>ア</b> |



**問 42** パスワードを用いて利用者を認証する方法のうち、適切なものはどれか。

- ア パスワードに対応する利用者 ID のハッシュ値を登録しておき、認証時に入力されたパスワードをハッシュ関数で変換して比較する。
- イ パスワードに対応する利用者 ID のハッシュ値を登録しておき、認証時に入力された利用者 ID をハッシュ関数で変換して比較する。
- ウ パスワードをハッシュ値に変換して登録しておき、認証時に入力されたパスワードをハッシュ関数で変換して比較する。
- エ パスワードをハッシュ値に変換して登録しておき、認証時に入力された利用者 ID をハッシュ関数で変換して比較する。



**問 43** 企業内ネットワークやサーバに侵入するために攻撃者が組み込むものはどれか。

- ア シンククライアントエージェント
- イ ストリクトルーティング
- ウ デジタルフォレンジックス
- エ バックドア



**問 44** PC への侵入に成功したマルウェアがインターネット上の指令サーバと通信を行う場合に、宛先ポートとして TCP ポート番号 80 が多く使用される理由はどれか。

- ア DNS のゾーン転送に使用されるので、通信がファイアウォールで許可されている可能性が高い。
- イ Web サイトの HTTPS 通信での閲覧に使用されることから、侵入検知システムで検知される可能性が低い。
- ウ Web サイトの閲覧に使用されることから、通信がファイアウォールで許可されている可能性が高い。
- エ ドメイン名の名前解決に使用されるので、侵入検知システムで検知される可能性が低い。



**問 45** 生体認証システムを導入するときに考慮すべき点として、最も適切なものはどれか。

- ア システムを誤作動させるデータを無害化する機能をもつライブラリを使用する。
- イ パターンファイルの頻繁な更新だけでなく、ヒューリスティックスなど別の手段と組み合わせる。
- ウ 本人のデジタル証明書を信頼できる第三者機関に発行してもらう。
- エ 本人を誤って拒否する確率と他人を誤って許可する確率の双方を勘案して装置を調整する。

## 問42 パスワード認証 キホン!

利用者のパスワードをそのままサーバに記録していると、サーバが不正アクセスされたときにパスワードが漏えいする危険があります。そこで、サーバにはパスワードそのものではなく、パスワードから生成した**ハッシュ値**を記録しておきます。ハッシュ値から元のパスワードは復元できないので、万一サーバが不正侵入されても、パスワードが漏えいするおそれがありません。

パスワードを照合するときは、利用者が入力したパスワードをハッシュ値に変換し、記録されているハッシュ値と照合します。正解は **ウ** です。

## 問43 バックドア

攻撃者が、企業内のネットワークやサーバに侵入するために組み込む、通常のアクセス経路以外の経路を**バックドア**（裏口）といいます。

- × **ア シンククライアントエージェント**とは、サーバ上で実行されるサービスを利用するために、シンククライアント側で動作するソフトウェアです。
- × **イ ストリクトルーティング**とは、送信元がパケットの転送経路を厳密に指定する経路制御方式です。
- × **ウ デジタルフォレンジックス**とは、コンピュータ犯罪を明らかにするために、証拠となるデータやログを収集・解析する調査技術です。
- **エ** 正解です。

## 問44 TCP ポート番号 80

内部に侵入した**マルウェア**が外部と密かに通信するためには、ファイアウォールで通信が遮断されない経路を利用する必要があります。**TCPポート番号 80**は、**ウェルノウンポート番号**と呼ばれる既定のポート番号の1つで、Webサーバとの通信に普段から利用されています。したがって、Webサイトの閲覧に使用しているPCなら、通信が許可されている可能性が高いと考えられます。

- × **ア** DNS のゾーン転送には、TCP ポート 53 番が使われます。
- × **イ** HTTPS 通信には TCP ポート 443 番が使われます。
- **ウ** 正解です。
- × **エ** DNS の名前解決には、UDP ポートの 53 番が使われます。

## 問45 生体認証システム

**生体認証**（バイオメトリクス認証）とは、指紋や虹彩といった、個人個人で異なる人間の身体的・行動的特徴を利用して個人認証を行う技術です。

生体認証の認証エラーには、①間違って本人を拒否してしまう場合（FRR）と、②間違って他人を許可してしまう場合（FAR）があります。本人判定の基準を厳しくすると、②のエラーが少なくなる代わりに①のエラーが増え、基準をゆるくすると①のエラーが少なくなる代わりに②のエラーが増えてしまいます。そのため、双方を勘案して検出装置を調整しなければなりません。正解は **エ** です。

## ハッシュ値

問 42

任意の長さのデータから生成された、固定長の比較的小さなデータ。ハッシュ関数によって生成する。

## シンククライアント

問 43

利用者が操作するクライアント端末に、必要最低限の機能だけを搭載し、処理の大部分をサーバ上で行うシステム形態。

## マルウェア

問 44

コンピュータウイルスやスパイウェア、ワームといった悪質なプログラムの総称。

## ポート番号

問 44

インターネットの TCP や UDP プロトコルで、宛先のアプリケーションを識別するために使用される番号。代表的なアプリケーションについては、標準のポート番号があらかじめ決められており、ウェルノウンポート番号と呼ばれる。ファイアウォールのパケットフィルタリングでは、パケットの宛先ポート番号や送信元ポート番号を調べて、通信を許可／遮断するかどうかを決定する。

## FRR

問 45

False Rejection Rate：本人拒否率。生体認証で本人を誤って拒否する確率。

## FAR

問 45

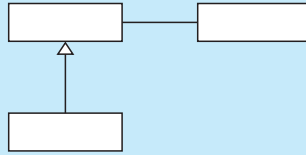
False Acceptance Rate：他人受入率。生体認証で他人を誤って許可する確率。

## 解答

- 問42 **ウ** 問43 **エ**  
問44 **ウ** 問45 **エ**



**問 46** UML における図の  の中に記述するものはどれか。



- ア** 関連名      **イ** クラス名      **ウ** 集約名      **エ** ユースケース名



**問 47** オブジェクト指向に基づく開発では、オブジェクトの内部構造が変更されても利用者がその影響を受けないようにすることができ、それによってオブジェクトの利用者がオブジェクトの内部構造を知らなくてもよいようにすることができる。これを実現するための概念を表す用語はどれか。

- ア** カプセル化      **イ** クラス化      **ウ** 構造化      **エ** モジュール化



**問 48** 要求の分析・設計時に使用する状態遷移図の説明として、適切なものはどれか。


- ア** 階層構造の形でプログラムの全体構造を記述する。  
**イ** 時間の経過や制御信号の変化などの、状態を変化させるきっかけと、変化に伴って実行する動作を記述する。  
**ウ** システムの機能を概要から詳細へと段階的に記述する。  
**エ** 処理間のデータの流れをデータフロー、処理、データストア及び外部の四つの記号で記述する。



**問 49** ブラックボックステストにおけるテストケースの設計方法として、適切なものはどれか。

- ア** プログラム仕様書の作成又はコーディングが終了した段階で、仕様書やソースリストを参照して、テストケースを設計する。  
**イ** プログラムの機能仕様やインタフェースの仕様に基づいて、テストケースを設計する。  
**ウ** プログラムの処理手順や内部構造に基づいて、テストケースを設計する。  
**エ** プログラムの全ての条件判定で、真と偽をそれぞれ1回以上実行させることを基準に、テストケースを設計する。

## 問46 UMLのクラス図 キホン!

図は、UMLのクラス図の例です。クラス図は、クラスとその属性、クラス間の関係を表した図で、の中にクラス名を書き、関連のあるクラス同士を線でつなぎます。正解はイです。

なお、白い三角のついた線は汎化関係を表します。

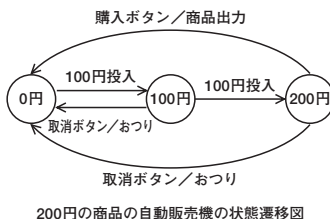
## 問47 オブジェクト指向開発 キホン!

オブジェクト指向の開発では、データとそれを操作する手続きをまとめてオブジェクトといいます。オブジェクトの内部構造を外部から隠ぺいし、利用者がオブジェクト内部に直接アクセスできないようにすることをカプセル化といいます。カプセル化によって、オブジェクトの内部構造を変更しても、利用者がその影響を受けないようにできます。正解はアです。

## 問48 状態遷移図

状態遷移図は、システムが取り得る状態と、各状態に移り変わるための条件(きっかけ)や動作を図で示したものです。

- × ア 構造化チャートなどの説明です。
- イ 正解です。
- × ウ HIPOチャートなどの説明です。
- × エ DFDの説明です。



## 問49 ブラックボックステスト よく出る!

ブラックボックステストは、プログラムが仕様どおりに動作するかどうかを、内部のロジックは考慮せず、入力に対する出力に着目してテストします。したがって、テストケースはプログラムの機能仕様やインタフェース仕様に基づいて設計します。

ブラックボックステストに対して、プログラム内部のロジックが正しいかどうかを検証するテストをホワイトボックステストといいます。

- × ア プログラム仕様書やソースリストは、ホワイトボックステストで参照します。
- イ 正解です。
- × ウ プログラムの内部構造に基づいてテストケースを設計するのは、ホワイトボックステストです。
- × エ すべての条件判定が真になる場合と偽になる場合をテストすることを条件網羅といいます。条件網羅はホワイトボックステストのテストケースを設計する手法です。

## 午前のカギ

### UML

問 46

Unified Modeling Language : オブジェクト指向開発で利用する様々な種類の図式を規格化したもの。

### 覚えよう!

問 46

#### 汎化 (is-a) 関係といえば

- 下位クラスと、下位クラスから共通の特性を取り出した上位クラスとの関係

汎化  自動車 ← 乗用車

#### 集約 (has-a) 関係

といえば

- 下位クラスが、上位クラスの一部であるような関係

集約  自動車 ◇ タイヤ

### DFD

問 48

データフローダイアグラム。業務過程をデータの流れ(データフロー)に着目して図式化したもの。

### 覚えよう!

問 49

#### ブラックボックステスト

といえば

- プログラムの外部仕様に沿ってテストケースを設定
- 同値分割法、限界値分析などがある

#### ホワイトボックステスト

といえば

- プログラムの内部構造に沿ってテストケースを設定
- 条件網羅、判定条件網羅、分岐網羅などがある

### 解答

問46 イ 問47 ア  
問48 イ 問49 イ

問 50 ソフトウェア開発の活動のうち、リファクタリングはどれか。

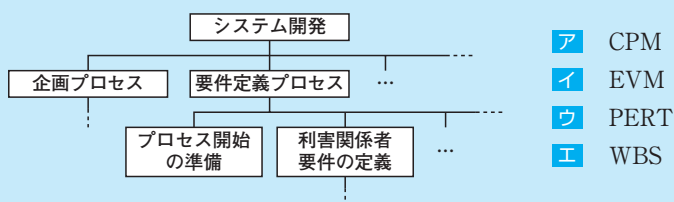
- ア ソフトウェアの品質を高めるために、2人のプログラマが協力して、一つのプログラムをコーディングする。
- イ ソフトウェアの保守性を高めるために、外部仕様を変更することなく、プログラムの内部構造を変更する。
- ウ 動作するソフトウェアを迅速に開発するために、テストケースを先に設定してから、プログラムをコーディングする。
- エ 利用者からのフィードバックを得るために、提供予定のソフトウェアの試作品を早期に作成する。

問 51 から問 60 までは、マネジメント系の問題です。

問 51 システム開発の進捗管理やソフトウェアの品質管理などで用いられる PDCA サイクルの“P”、“D”、“C”、“A”は、それぞれ英単語の頭文字をとったものである。3番目の文字“C”が表す単語はどれか。

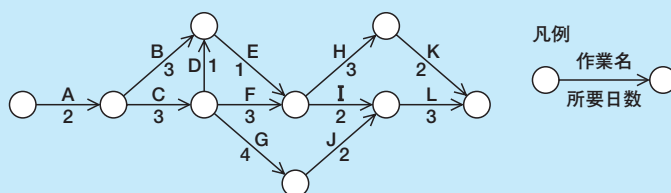
- ア Challenge
- イ Change
- ウ Check
- エ Control

問 52 図のように、プロジェクトチームが実行すべき作業を上位の階層から下位の階層へ段階的に分解したものを何と呼ぶか。



- ア CPM
- イ EVM
- ウ PERT
- エ WBS

問 53 図に示すプロジェクト活動のクリティカルパスはどれか。



- ア A → B → E → I → L
- イ A → C → D → E → H → K
- ウ A → C → F → I → L
- エ A → C → G → J → L



## 問50 リファクタリング

**リファクタリング**とは、外側から見える仕様は変更せずに、プログラムの内部構造を変更するソフトウェア開発です。主に、処理速度を高速化したり、ソフトウェアの保守性を高めるために行います。

- × **ア** ベアプログラミングの説明です。
- **イ** 正解です。
- × **ウ** テスト駆動開発の説明です。
- × **エ** プロトタイピングの説明です。

## 問51 PDCAサイクル

**PDCAサイクル**とは、Plan (計画) → Do (実行) → Check (評価) → Action (改善) の4段階の活動を繰り返し行い、業務プロセスを継続的に管理していく手法です。PDCAの“C”はCheck (評価)を表します。正解は**ウ**です。

## 問52 WBS よく出る!

プロジェクト全体をいくつかの工程に分割し、各工程をさらに細かい作業に分割し…のように、階層的に分割して管理していく手法を**WBS** (Work Breakdown Structure) といいます。

- × **ア** CPM (Critical Path Method) は、プロジェクトの日程を管理する手法です。
- × **イ** EVM (Earned Value Management) は、プロジェクトの進捗状況を、金額をベースに定量的に管理する手法です。
- × **ウ** PERT (Program Evaluation and Review Technique) は、アローダイアグラムなどでプロジェクトの進捗を管理する手法です。
- **エ** 正解です。

## 問53 クリティカルパス キホン!

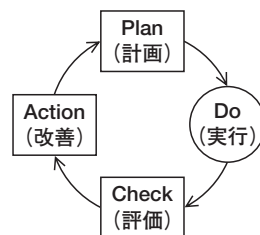
開始から終了までの各作業のうち、時間的余裕のない一連の作業経路を、**クリティカルパス**といいます。クリティカルパス上にある各作業が1日でも遅れると、プロジェクト全体の進捗の遅れにつながります。

**アローダイアグラム**では、開始から終了に至る複数の作業経路のうち、合計の所要日数の最も多くなる経路がクリティカルパスになります。したがって、選択肢の中で、所要日数の合計が最も多い経路が正解です。

- × **ア**  $2日 + 3日 + 1日 + 2日 + 3日 = 11日$
- × **イ**  $2日 + 3日 + 1日 + 1日 + 3日 + 2日 = 12日$
- × **ウ**  $2日 + 3日 + 3日 + 2日 + 3日 = 13日$
- **エ**  $2日 + 3日 + 4日 + 2日 + 3日 = 14日$

## 覚えよう! 問51

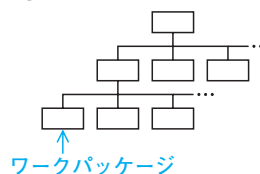
## PDCAサイクルといえば



## 覚えよう! 問52

## WBS といえば

- 作業を階層的に分割して管理
- 細分化された最小単位の作業をワークパッケージという



## アローダイアグラム 問53

プロジェクトを達成するのに必要な作業を矢線で、作業の結合点を○印で表し、所要日数を示して日程計画を立てるのに効果的な図。

## 解答

- |     |          |     |          |
|-----|----------|-----|----------|
| 問50 | <b>イ</b> | 問51 | <b>ウ</b> |
| 問52 | <b>エ</b> | 問53 | <b>エ</b> |

☐ **問 54** ある新規システムの開発規模を見積もったところ、500FP（ファンクションポイント）であった。このシステムを構築するプロジェクトには、開発工数の他にシステムの導入や開発者教育の工数が 10 人月必要である。また、プロジェクト管理に、開発と導入・教育を合わせた工数の 10%を要する。このプロジェクトに要する全工数は何人月か。ここで、開発の生産性は 1 人月当たり 10FP とする。

☐ **ア** 51

☐ **イ** 60

☐ **ウ** 65

☐ **エ** 66

☐ **問 55** システムの移行方式の一つである一斉移行方式の特徴として、最も適切なものはどれか。

☐ **ア** 新旧システム間を接続するアプリケーションが必要となる。

☐ **イ** 新旧システムを並行させて運用し、ある時点で新システムに移行する。

☐ **ウ** 新システムへの移行時のトラブルの影響が大きい。

☐ **エ** 並行して稼働させるための運用コストが発生する。

☐ **問 56** IT サービスマネジメントにおける“既知の誤り（既知のエラー）”の説明はどれか。

☐ **ア** 根本原因が特定されている又は回避策が存在している問題

☐ **イ** サービスデスクに問合せがあった新たなインシデント

☐ **ウ** サービスマネジメント計画での矛盾や漏れ

☐ **エ** 静的検査で検出したプログラムの誤り

☐ **問 57** 次の条件で IT サービスを提供している。SLA を満たすための、1 か月のサービス時間帯中の停止時間は最大何時間か。ここで、1 か月の営業日は 30 日とし、サービス時間帯中は保守などのサービス計画停止は行わないものとする。

〔SLA の条件〕

・ サービス時間帯は、営業日の午前 8 時から午後 10 時までとする。

・ 可用性を 99.5%以上とする。

☐ **ア** 0.3

☐ **イ** 2.1

☐ **ウ** 3.0

☐ **エ** 3.6



## 問54 工数計算

1人月当たり10FP開発がすすむので、システム全体の開発工数は  $500 \div 10 = 50$  人月です。このほかに、システム導入と開発者教育に10人月、プロジェクト管理に  $(50 + 10) \times 10\% = 6$  人月が必要となります。以上から、プロジェクト全体の工数は、 $50 + 10 + 6 = 66$  人月になります。正解は **エ** です。

## 問55 一斉移行方式 よく出る！

新規に開発したシステムを、旧システムと置き換える作業を**移行**といいます。**一斉移行**は、ある日を境に、旧システムをまるごと新システムに置き換えてしまう方式です。旧システムを並行して運用する必要がないので移行コストは少なく済みませんが、移行時にトラブルがあると、業務が停止するなど影響が大きいのがデメリットです。

- × **ア** 一斉移行では新旧システムを同時に稼働しないので、接続アプリケーションは必要ありません。
- × **イ** 一斉移行では、旧システムは並行運用しません。
- **ウ** 正解です。
- × **エ** 並行運用にともなうコストが少ないのが、一斉移行のメリットです。

## 問56 既知の誤り(既知のエラー) 初モノ！

インシデントが発生したとき、そのインシデントの原因を**問題**といいます。ITサービスマネジメントでは、問題管理プロセスでインシデントの根本的な原因を調査しますが、調査の結果その原因や回避策が判明すれば、その問題は「**既知の誤り**」と呼ばれるようになります。以上から、正解は **ア** です。既知の誤りは **KEDB** (既知のエラーデータベース) に登録され、インシデント管理や問題管理によって参照されます。

## 問57 SLA キホン！

1か月のサービス時間の合計は、「営業日の午前8時から午後10時までの14時間」×1か月、すなわち  $14 \text{ 時間} \times 30 \text{ 日} = 420 \text{ 時間}$  です。**可用性**が99.5%以上なので、サービス停止時間は0.5% (= 0.005) 以下でなければいけません。したがって、最大の停止時間は  $420 \times 0.005 = 2.1 \text{ 時間}$  となります。正解は **イ** です。

## 午前のカギ

## 覚えよう！ 問54

ファンクションポイント法  
といえば

- 入出力、画面、ファイルなどの機能の個数によって開発規模を見積もる手法
- 個々の機能を複雑さに応じて点数化する

## 人月 問54

1人が1か月に行う工数の単位。

## 覚えよう！ 問55

## 一斉移行方式といえば

- 移行コストが少ない
- 移行トラブルの影響が大きい

## インシデント 問56

ITサービスの予定外の停止や、サービス品質の低下、障害などのこと。インシデントが発生したときは、インシデント管理プロセスで可能な限り速やかにサービスを復旧させる。

## SLA 問57

Service Level Agreement: ITサービスの利用者と提供者が、提供するサービス品質について取り交わす合意。

## 可用性 問57

利用者が必要とするときに、サービスが利用可能なこと。通常、サービス時間中にサービスを利用できる確率として計算される。

## ○ 解答 ○

- 問54 **エ** 問55 **ウ**  
問56 **ア** 問57 **イ**



**問 58** システム監査の実施体制に関する記述のうち、適切なものはどれか。

- ア 監査依頼者が監査報告に基づく改善指示を行えるように、システム監査人は監査結果を監査依頼者に報告する。
- イ 業務監査の一部として情報システムの監査を行う場合には、利用部門のメンバによる監査チームを編成して行う。
- ウ システム監査人が他の専門家の支援を受ける場合には、支援の範囲、方法、監査結果の判断などは、他の専門家の責任において行う。
- エ 情報システム部門における開発の状況の監査を行う場合は、開発内容を熟知した情報システム部門員による監査チームを編成して行う。



**問 59** システム設計の段階において、利用者要件が充足されないリスクを低減するコントロールを監査するときのチェックポイントはどれか。

- ア システム設計書に基づき、プログラム仕様書を作成していること
- イ システムテスト要件に基づいてテスト計画を作成し、システム運用部門の責任者の承認を得ていること
- ウ プログラミングは定められた標準に従っていること
- エ 利用部門が参画して、システム設計書のレビューを行っていること



**問 60** “システム管理基準”に基づいて、システムの信頼性、安全性、効率性を監査する際に、システムが不正な使用から保護されているかどうかという安全性の検証項目として、最も適切なものはどれか。

- ア アクセス管理機能の検証
- イ フェールソフト機能の検証
- ウ フォールトトレラント機能の検証
- エ リカバリ機能の検証

問 61 から問 80 までは、ストラテジ系の問題です。



**問 61** 共通フレームによれば、システム化構想の立案で作成されるものはどれか。

- ア 企業で将来的に必要な最上位の業務機能と業務組織を表した業務の全体像
- イ 業務手順やコンピュータ入出力情報など実現すべき要件
- ウ 日次や月次で行う利用者業務やコンピュータ入出力作業の業務手順
- エ 必要なハードウェアやソフトウェアを記述した最上位レベルのシステム方式

問58 システム監査の実施体制 **キホン!**

改善の必要性を監査を受けた部署に直接報告しても、改善が実施されるとは限りません。**システム監査**では、監査結果は監査対象ではなく、経営者などの監査依頼者に提出します。

- **ア** 正解です。
- × **イ** 業務監査では、利用部門のメンバは監査される側に属するので、監査人の独立性の観点から、監査チームに加えるべきではありません。
- × **ウ** システム監査人は必要に応じて専門家の支援を受けるべきですが、支援の範囲や方法、監査結果の判断などは、あくまでもシステム監査人の責任において行います。
- × **エ** 情報部門の監査では、情報システム部門員も監査の対象になるので、監査人の独立性の観点から、監査チームに加えるべきではありません。

## 問59 システム設計の監査

**システム設計**の段階で、利用者のニーズが反映されているかどうかチェックしているものを選択肢から選びます。

- × **ア** システム設計書の内容をチェックする必要があるので、プログラム仕様書は関係ありません。
- × **イ** システムテストの段階では、システム設計は終わっています。
- × **ウ** プログラムの段階では、利用者のニーズは考慮されません。
- **エ** 正解です。システム設計書の内容を利用部門がチェックすることで、利用者のニーズがきちんと反映されているかどうか確認できます。

## 問60 安全性の検証項目

「システムが不正な使用から保護されているかどうか」を検証するには、アクセス管理機能を検証します。**アクセス管理**とは、適切な権限のある者だけが、必要な情報にアクセスできるようにするプロセスのことです。以上から、**ア**が正解です。なお、**イ**～**エ**は、いずれもシステムの信頼性の検証項目です。

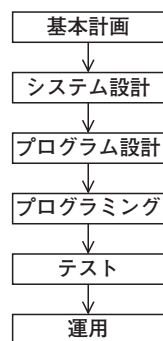
問61 共通フレーム **よく出る!**

システム化構想の立案とは、**共通フレーム**の**企画プロセス**において、経営上のニーズや課題に沿ったシステムを構想することです。構想の立案に当たっては、対象となる業務を明確化するとともに、将来的に必要となる業務機能や業務組織を取り込んだ、業務の新全体像を描く必要があります。

- **ア** 正解です。
- × **イ** **企画プロセス**の利害関係者要件の定義で作成します。
- × **ウ** **運用プロセス**の準備段階で作成します。
- × **エ** **開発プロセス**のシステム方式設計で作成します。

**監査人の独立性** **問58**  
監査人が、被監査主体と身分上密接な利害関係をもたないようにすること。

**ソフトウェア開発の工程モデルの例** **問59**



**レビュー** **問59**  
システム開発の各工程でできた成果物を、次の工程に進む前に評価・検証すること。

**共通フレーム** **問61**  
ソフトウェアの企画から開発、保守に至るライフサイクルの各段階で、利用者（システムの発注者）やベンダ（受注者）が果たす役割や業務の範囲などを規定したもの。

° 解答 °

問58 **ア** 問59 **エ**  
問60 **ア** 問61 **ア**

- ☐ **問 62** 改善の効果を定量的に評価するとき、複数の項目で評価した結果を統合し、定量化する方法として重み付け総合評価法がある。表の中で優先すべき改善案はどれか。

評価項目	評価項目の重み	改善案			
		案 1	案 2	案 3	案 4
省力化	4	6	8	2	5
期間短縮	3	5	5	9	5
資源削減	3	6	4	7	6

**ア** 案 1

**イ** 案 2

**ウ** 案 3

**エ** 案 4

- ☐ **問 63** SOA を説明したものはどれか。

- ア** 業務体系、データ体系、適用処理体系、技術体系の四つの主要概念から構成され、業務とシステムの最適化を図る。
- イ** サービスというコンポーネントからソフトウェアを構築することによって、ビジネス変化に対応しやすくする。
- ウ** データフローダイアグラムを用い、情報に関するモデルと機能に関するモデルを同時に作成する。
- エ** 接続、選択、反復の三つの論理構造の組合せで、コンポーネントレベルの設計を行う。

- ☐ **問 64** デジタルディバイドの解消のために取り組むべきことはどれか。

- ア** IT 投資額の見積りを行い、投資目的に基づいて効果目標を設定して、効果目標ごとに目標達成の可能性を事前に評価すること
- イ** IT を活用した家電や設備などの省エネルギー化やテレワークなどによる業務の効率向上によって、エネルギー消費を削減すること
- ウ** 情報リテラシの習得機会を増やしたり、情報通信機器や情報サービスが一層利用しやすい環境を整備したりすること
- エ** 製品や食料品などの生産段階から最終消費段階又は廃棄段階までの全工程について、IC タグを活用して流通情報を追跡可能にすること

- ☐ **問 65** “システム管理基準”において、情報システムの費用、スケジュール、開発体制、投資効果などを明確にする計画はどれか。

**ア** 開発計画

**イ** 事業継続計画

**ウ** 全体最適化計画

**エ** 年間運用計画

## 問62 重み付け総合評価表

案1～案4の点数を、重み付けを考慮して合計すると次のようになります。  
合計点のもっとも大きい案を優先します。

省力化	期間短縮	資源削減
案1: $6 \times 4 + 5 \times 3 + 6 \times 3 = 57$		
案2: $8 \times 4 + 5 \times 3 + 4 \times 3 = 59$		
案3: $2 \times 4 + 9 \times 3 + 7 \times 3 = 56$		
案4: $5 \times 4 + 5 \times 3 + 6 \times 3 = 53$		
↑ 重み	↑ 重み	↑ 重み

合計点のもっとも大きいのは案2です。正解はイです。

## 問63 SOA よく出る!

SOA (Service-Oriented Architecture) は、ソフトウェアの機能をサービスととらえ、複数のサービスを部品のように組み合わせてシステムを構築していく設計手法です。

- × ア エンタープライズアーキテクチャ (EA) の説明です。
- イ 正解です。
- × ウ 機能情報関連図 (DFD: データフローダイアグラム) の説明です。
- × エ NS チャートなどの構造化設計手法の説明です。

## 問64 デジタルディバイド

デジタルディバイドとは、IT を活用できる人とできない人との間に生じる、社会的・経済的な格差のことです。デジタルディバイドは、情報リテラシの不足や、IT の利用環境の低さから生じるため、IT 教育の機会を増やしたり、利用環境を整備することで解消していくのが適切です。以上から、正解はウです。

## 問65 システム管理基準

情報システムの費用やスケジュール、開発体制、投資効果の明確化は、いずれもシステムを新たに開発する際に必要なもので、開発計画で行います。

- ア 正解です。
- × イ 事業継続計画 (BCP) は、災害などで情報システムが損害を受けた場合に備え、システムを早期に復旧して事業を再開させるために立てておく計画です。
- × ウ 全体最適化計画では、情報戦略における情報システム全体の最適化目標を設定し、組織全体の情報システムのあるべき姿を明確にします。
- × エ 年間運用計画は、年間を通じた情報システムの運用計画です。

## 覚えよう! 問63

## SOA とはいば

- 複数のサービスを組み合わせてシステムを構成する設計手法

## デジタルディバイド 問64

情報リテラシの有無やIT の利用環境の相違などによって生じる、社会的または経済的格差のこと。

## システム管理基準 問65

経済産業省が策定したシステム管理のためのガイドライン。

## 解答

問62 イ 問63 イ  
問64 ウ 問65 ア



**問 66** SCMの目的はどれか。

- ア 顧客情報や購買履歴、クレームなどを一元管理し、きめ細かな顧客対応を行うことによって、良好な顧客関係の構築を目的とする。
- イ 顧客情報や商談スケジュール、進捗状況などの商談状況を一元管理することによって、営業活動の効率向上を目的とする。
- ウ 生産や販売、在庫、会計など基幹業務のあらゆる情報を統合管理することによって、経営効率の向上を目的とする。
- エ 調達から販売までの複数の企業や組織にまたがる情報を統合的に管理することによって、コスト低減や納期短縮などを目的とする。



**問 67** コトラーの競争戦略によると、業界でのシェアは高くないが、特定の製品・サービスに経営資源を集中することで、収益を高め、独自の地位を獲得することを戦略目標とする企業はどれか。

- |   |             |   |           |
|---|-------------|---|-----------|
| ア | マーケットチャレンジャ | イ | マーケットニッチャ |
| ウ | マーケットフォロワ   | エ | マーケットリーダー |



**問 68** プロダクトライフサイクルにおける成長期を説明したものはどれか。

- ア 売上が急激に増加する時期である。市場が活性化し、新規参入企業によって競争が激化してくる。
- イ 売上と利益が徐々に減少する時期である。追加投資を控えて市場から撤退することが検討される。
- ウ 需要の伸びが鈍化してくる時期である。製品の品質改良などによって、シェアの維持、利益の確保が行われる。
- エ 先進的な消費者に対して製品を販売する時期である。製品の認知度を高める戦略が採られる。



**問 69** ある製品の設定価格と需要との関係が1次式で表せるとき、aに入る適切な数値はどれか。

- (1) 設定価格を3,000円にすると、需要は0個になる。
- (2) 設定価格を1,000円にすると、需要は60,000個になる。
- (3) 設定価格を1,500円にすると、需要は  個になる。

- |   |        |   |        |   |        |   |        |
|---|--------|---|--------|---|--------|---|--------|
| ア | 30,000 | イ | 35,000 | ウ | 40,000 | エ | 45,000 |
|---|--------|---|--------|---|--------|---|--------|



## 問66 SCMの目的 **よく出る!**

SCM (Supply Chain Management) は、生産から購買、販売、物流に至る一連の情報を統合管理して、商品の流れ（サプライチェーン）全体の効率を向上させる手法です。

- × **ア** CRM (Customer Relationship Management) の目的です。
- × **イ** SFA (Sales Force Automation) の目的です。
- × **ウ** ERP (Enterprise Resource Planning) の目的です。
- **エ** 正解です。

## 問67 コトラーの競争戦略

経営学者のコトラー (Kotler) は、競争上の戦略によって企業を4つのタイプに分類しています。このうち、特定の製品・サービスに特化して独自の地位を獲得する企業を**ニッチャ**（マーケットニッチャ）といいます。

- × **ア** チャレンジャ（マーケットチャレンジャ）は、差別化によってシェア獲得を目指す企業です。
- **イ** 正解です。
- × **ウ** フォロワ（マーケットフォロワ）は、市場にすばやく追随し、リーダ企業を模倣する企業です。
- × **エ** リーダ（マーケットリーダ）は、市場全体をカバーして最大シェアを確保しようとする全方位戦略をとる企業です。

## 問68 プロダクトライフサイクル **よく出る!**

プロダクトライフサイクルとは、製品の売上の変遷を、①**導入期**→②**成長期**→③**成熟期**→④**衰退期**の4段階にモデル化したものです。このうちの成長期は、製品が市場に認知され、売上が伸びていく段階です。

- **ア** 正解です。
- × **イ** 衰退期の説明です。
- × **ウ** 成熟期の説明です。
- × **エ** 導入期の説明です。

## 問69 設定価格と需要の関係 **キホン!**

設定価格を  $x$ 、需要を  $y$  とすれば、両者の関係は1次式  $y = ax + b$  で表すことができます。また、問題文 (1)、(2) より、次の式が成り立ちます。

$$(1) x = 3000 \text{ のとき } y = 0 \quad \rightarrow 3000a + b = 0 \quad \cdots \textcircled{1}$$

$$(2) x = 1000 \text{ のとき } y = 60000 \rightarrow 1000a + b = 60000 \quad \cdots \textcircled{2}$$

式①、②を連立方程式として、 $a$ 、 $b$  を求めると、 $a = -30$ 、 $b = 90000$  となります。これを  $y = ax + b$  に当てはめると、1次式  $y = -30x + 90000$  を得ます。設定価格  $x = 1500$  をこの1次式に代入すれば、

$$y = -30 \times 1500 + 90000 = 45000$$

以上より、設定価格を1,500円にすると、需要は45,000個になると期待できます。正解は**エ**です。

## 午前のカギ

### 覚えよう! 問66

#### SCM といえば

- サプライチェーン全体を最適化し、リードタイムを短縮

### 覚えよう! 問67

#### 企業の競争戦略といえば

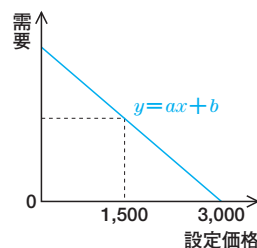
- **リーダ**：すべての顧客をターゲットにした全方位戦略
- **フォロワ**：リーダ企業に追随する模倣戦略
- **ニッチャ**：専門市場に特化する戦略
- **チャレンジャ**：市場シェア獲得をねらう差別化戦略

### 覚えよう! 問68

#### プロダクトライフサイクル といえば

- 導入期→成長期→成熟期→衰退期

### 1次式の関係 問69



### 連立方程式を解く 問69

$$\begin{array}{rcl}
 \textcircled{1} - \textcircled{2}: & 3000a + b = 0 & \\
 & - 1000a + b = 60000 & \\
 \hline
 & 2000a = -60000 & \\
 & \therefore a = -30 & \\
 a = -30 \text{ を } \textcircled{2} \text{ に代入:} & & \\
 & 1000 \times -30 + b = 60000 & \\
 & \therefore b = 90000 & 
 \end{array}$$

#### 解答

- 問66 **エ** 問67 **イ**  
問68 **ア** 問69 **エ**

問70 他の技法では答えが得られにくい、未来予測のような問題に多く用いられ、(1)～(3)の手順に従って行われる予測技法はどれか。

- (1) 複数の専門家を回答者として選定する。  
 (2) 質問に対する回答結果を集約してフィードバックし、再度質問を行う。  
 (3) 回答結果を統計的に処理し、分布とともに回答結果を示す。

ア クロスセクション法  
 ウ 親和図法

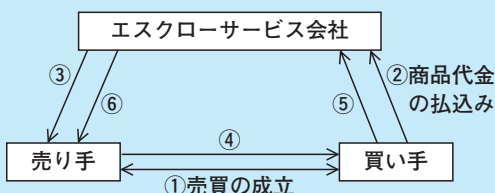
イ シナリオライティング法  
 エ デルファイ法

問71 ある工場では表に示す3製品を製造している。実現可能な最大利益は何円か。ここで、各製品の月間需要量には上限があり、また、組立て工程に使える工場の時間は月間200時間までで、複数種類の製品を同時に並行して組み立てることはできないものとする。

	製品X	製品Y	製品Z
1個当たりの利益(円)	1,800	2,500	3,000
1個当たりの組立て所要時間(分)	6	10	15
月間需要量上限(個)	1,000	900	500

ア 2,625,000  
 イ 3,000,000  
 ウ 3,150,000  
 エ 3,300,000

問72 インターネットオークションなどで利用されるエスクローサービスの取引モデルの⑤に当てはまる行為はどれか。ここで、①～⑥は取引の順序を示し、③～⑥はア～エのいずれかに対応する。



ア 商品受領の通知  
 イ 商品の発送  
 ウ 代金の支払  
 エ 代金の入金通知

## 解説

### 問70 未来予測の手法

複数の専門家に質問し、その回答を集約して専門家に見せて再度質問する、ということを何度か繰り返し、最終的な回答結果を統計的に処理する未来予測の手法を、**デルファイ法**といいます。

- × **ア クロスセクション法**は、現時点の状況が今後どのように推移するかを、先行している他の事例との比較で予測する手法です。
- × **イ シナリオライティング法**は、将来起こりうる状況をいくつかの仮説として記述する未来予測の手法です。

## 午前のカギ



- × **ウ** 親和図法は、多数の情報をカードに記入して、類似したものをグループ化し、問題点を明確にする手法です。KJ 法とも呼ばれます。
- **エ** 正解です。

## 問71 最大利益の計算 **よく出る!**

製造時間が限られているので、単位時間当たりの利益が大きい製品を優先して作ったほうが、全体の利益を大きくできます。そこでまず、製造時間1分あたりに生じる利益を、製品ごとに計算します。

$$\text{製品 X : } 1,800 \div 6 = 300 \text{ 円 / 分}$$

$$\text{製品 Y : } 2,500 \div 10 = 250 \text{ 円 / 分}$$

$$\text{製品 Z : } 3,000 \div 15 = 200 \text{ 円 / 分}$$

以上から、まず製品 X を優先的に作り、余った時間で製品 Y を作り、最後に製品 Z を作る時が、最も多くの利益を得られます。

組立て工程に使える時間を分に換算すると、 $200 \text{ 時間} \times 60 = 12,000 \text{ 分}$ です。この時間を、まず製品 X の製造に割り当てます。製品 X を需要上限である1,000 個作る場合、所要時間は  $6 \times 1,000 = 6,000 \text{ 分}$ 。これで、工場を使える時間は残り  $12,000 - 6,000 = 6,000 \text{ 分}$  になります。

次に、製品 Y を製造します。製品 Y の需要上限は900 個ですが、製品 Y を900 個作る場合、所要時間は  $10 \times 900 = 9,000 \text{ 分}$  で、残り時間6,000 分をオーバーしてしまいます。残り時間をめいっぱい使うと、製品 Y は  $6,000 \div 10 = 600 \text{ 個}$  作れます。これで残り時間はゼロになるので、製品 Z を製造する時間はありません。

以上から、利益が最も大きくなるのは、製品 X を1,000 個、製品 Y を600 個、製品 Z を0 個作ったときで、その金額は、

$$1,800 \times 1,000 + 2,500 \times 600 = 1,800,000 + 1,500,000 = 3,300,000 \text{ 円}$$

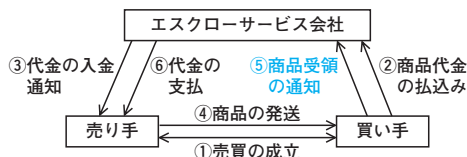
となります。正解は **エ** です。

## 問72 エスクローサービス **初モノ!**

**エスクローサービス**とは、インターネットオークションの売り手と買い手の間に入り、代金の支払いを仲介するサービスです。エスクローサービス会社が仲介する場合、取引の手順は次のようになります。

- ① 売り手と買い手の間で売買が成立
- ② 買い手は商品の代金をエスクロー業者にいったん預ける
- ③ エスクローサービス会社は代金の入金を売り手に通知する
- ④ 売り手が商品を発送する
- ⑤ 買い手は商品を受け取ったことをエスクローサービス会社に通知する
- ⑥ エスクローサービス会社は預かっていた代金を売り手に支払う

以上から、⑤に当てはまるのは **ア** の「商品受領の通知」です。



## 午前のカギ

### 覚えよう! 問70

**デルファイ法**といえば

- ① 複数の専門家を選び、アンケートを送付する
- ② アンケート結果を専門家にフィードバックして、再度質問する
- ③ ①、②を何度か繰り返し、結果を統計的に処理して示す

**参考** エスクローサービスによって、代金を支払ったのに商品が届かなかったり、商品を送ったのに代金が支払われないといった、売り手と買い手のトラブルを避けることができるんだ。



### 解答

- 問70 **エ** 問71 **エ**  
問72 **ア**



### 問 73 ロングテールの説明はどれか。

- ア Web コンテンツを構成するテキストや画像などのデジタルコンテンツに、統合的・体系的な管理、配信などの必要な処理を行うこと
- イ インターネットショッピングで、売上の全体に対して、あまり売れない商品群の売上合計が無視できない割合になっていること
- ウ 自分の Web サイトやブログに企業へのリンクを掲載し、他者がこれらのリンクを経由して商品を購入したときに、企業が紹介料を支払うこと
- エ メーカーや卸売業者から商品を直接発送することによって、在庫リスクを負うことなく自分の Web サイトで商品が販売できること



### 問 74 携帯電話端末の機能の一つであるテザリングの説明として、適切なものはどれか。

- ア 携帯電話端末に、異なる通信事業者の SIM カードを挿して使用すること
- イ 携帯電話端末をモデム又はアクセスポイントのように用いて、PC、ゲーム機などから、インターネットなどを利用したデータ通信をすること
- ウ 契約している通信事業者のサービスエリア外でも、他の事業者のサービスによって携帯電話端末を使用すること
- エ 通信事業者に申し込むことによって、青少年に有害なサイトなどを携帯電話端末に表示しないようにすること



### 問 75 特徴(1)～(4)をもつ組織形態はどれか。

- (1) 戦略的目標を達成するために、必要な専門家を各部門から集めて編成する。
- (2) 環境の変化に適応する戦略的組織であり、職能部門などから独立している。
- (3) 所期の目的を達成すれば解散する流動性をもつ。
- (4) タスクフォースは、この組織形態に属す。

- |            |                |
|------------|----------------|
| ア 事業部制組織   | イ プロジェクト組織     |
| ウ マトリックス組織 | エ ラインアンドスタッフ組織 |

## 問73 ロングテール

商品の売上高を大きい順に並べたグラフを作ると、あまり売れない商品の列が長く伸びたグラフになります。この部分が長い尻尾のように見えることから、**ロングテール**といいます。インターネットショッピングでは店舗スペースに制限がなく、売れ筋以外の商品も数多く扱うことができるため、こうした商品群の売上合計が無視できない割合を占めることがあります。

- × **A** CMS (Content Management System) の説明です。
- **I** 正解です。
- × **U** アフィリエイトの説明です。
- × **E** ドロップshippingの説明です。

## 問74 テザリング 初モノ!

**テザリング**とは、PC やゲーム機からインターネットを利用するために、スマートフォンなどの携帯電話端末を中継機器として利用する機能です。PC、ゲーム機と携帯電話端末との接続には、無線 LAN や Bluetooth、USB ケーブルなどが使われます。



- × **A** 特定の通信事業者の SIM カードしか使えない携帯電話端末で、異なる通信事業者の SIM カードを使えるようにすることを **SIM ロック解除** といいます。
- **I** 正解です。
- × **U** ローミングの説明です。
- × **E** フィルタリング (有害サイトアクセス制限) の説明です。

## 問75 組織形態 キホン!

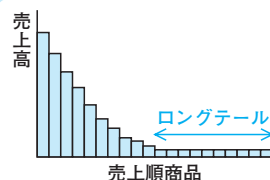
特定の目的のために各部門から必要な人材を集めて編成され、所定の目的を達成すれば解散する組織形態は、**プロジェクト組織**です。

- × **A** **事業部制組織**は、製品や地域ごとに事業部を構成し、各事業部単位に利益責任をもたせる組織形態です。
- **I** 正解です。
- × **U** **マトリックス組織**は、メンバーが同時に複数の指揮命令系統に所属する組織形態です。
- × **E** **ラインアンドスタッフ組織**は、基幹業務を担う製造部、販売部などのライン部門と、それらを支援する経理部、総務部などのスタッフ部門から構成される組織形態です。

## 午前のカギ

### ロングテール

問 73



問 74

**参考** テザリング (tethering) のもとの動詞 tether は、「繋ぎ止める」という意味。



### SIM カード

問 74

携帯電話端末で、電話番号を特定するために通信事業者が発行する IC カード。携帯電話端末を、特定の通信事業者の SIM カードしか使えない状態にすることを SIM ロックという。

### タスクフォース

問 75

特定の目的のために、一時的に編成される部署のこと。

### 解答

問 73 **I** 問 74 **I**  
問 75 **I**



**問 76** 六つの部署に合計 30 台の PC がある。その全ての PC で使用するソフトウェアを購入したい。表に示す購入方法がある場合、最も安く購入すると何円になるか。ここで、各部署には最低 1 冊のマニュアルが必要であるものとする。

購入方法	使用権	マニュアル	価格 (円)
単体で 1 本	1	1	15,000
1 ライセンス	1	0	12,000
5 ライセンス	5	0	45,000

**ア** 270,000

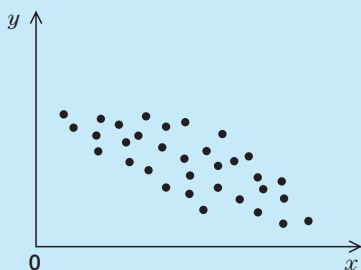
**イ** 306,000

**ウ** 315,000

**エ** 318,000



**問 77** 図は、製品の製造上のある要因の値  $x$  と品質特性の値  $y$  との関係をプロットしたものである。この図から読み取れることはどれか。



**ア**  $x$  から  $y$  を推定するためには、2 次回帰係数の計算が必要である。

**イ**  $x$  から  $y$  を推定するための回帰式は、 $y$  から  $x$  を推定する回帰式と同じである。

**ウ**  $x$  と  $y$  の相関係数は正である。

**エ**  $x$  と  $y$  の相関係数は負である。



**問 78** 表は、ある企業の損益計算書である。損益分岐点は何百万円か。

単位 百万円

項目	内訳	金額
売上高		700
売上原価	変動費 100 固定費 200	300
売上総利益		400
販売費・一般管理費	変動費 40 固定費 300	340
営業利益		60

**ア** 250

**イ** 490

**ウ** 500

**エ** 625

## 問76 ソフトウェアライセンス

6つの部署に最低1冊のマニュアルが必要なので、単体のライセンスが最低6本必要です。まず、これらを6台分のPCに割り当てます。

残り24台分のPC用には、「5ライセンス」を4本と「1ライセンス」を4本購入すればぴったりですが、「1ライセンス」4本の価格は $12,000 \times 4 = 48,000$ 円で、「5ライセンス」1本より高くなってしまいます。したがって、「5ライセンス」をもう1本追加したほうが安上がりです。以上から、最も安い購入価格は、

$$15,000 \times 6 + 45,000 \times 5 = 90,000 + 225,000 = 315,000 \text{ 円}$$

単体      5ライセンス

になります。正解は **ウ** です。

## 問77 散布図 キホン!

図のように、2つの項目を  $x$  軸と  $y$  軸にとり、値の組合せを点でプロットしたグラフを**散布図**といいます。この散布図では、点の集まりが右下がりになっており、 $x$ の値が増加するほど  $y$ の値が減少する傾向にあることがわかります。このような場合、 $x$ と  $y$ には「**負の相関**」関係があるといえます。

また、相関の度合いは**相関係数**と呼ばれる数値で表されます。相関係数は-1から1の範囲の数値で、「正の相関」のとき正、「負の相関」のとき負、「無相関」のとき0になります。

× **ア** 回帰式は1次式（直線のグラフ）で表すことができます。

× **イ** 2つの回帰式は一般に同じにはなりません。

× **ウ** 相関係数が正になるのは「正の相関」の場合です。

○ **エ** 正解です。

## 問78 損益分岐点の計算 キホン!

**損益分岐点**は、売上高と総費用（変動費と固定費の合計）が一致する利益ゼロのポイントで、次の式で求めることができます。

$$\text{損益分岐点} = \frac{\text{固定費}}{1 - \text{変動費率}}$$

変動費率とは、売上高に対する変動費の割合です。問題文の損益計算書から、売上高は700百万円、変動費は $100 + 40 = 140$ 百万円。したがって、変動費率は $140 / 700 = 1 / 5$ となります。

また、固定費は $200 + 300 = 500$ 百万円なので、損益分岐点は次のように計算できます。

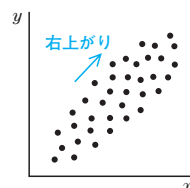
$$\text{損益分岐点} = \frac{500}{1 - 1/5} = 500 \times \frac{5}{4} = 625 \text{ 百万円}$$

以上から、正解は **エ** です。

## 覚えよう! 問77

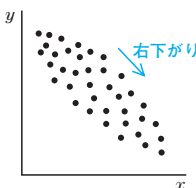
## 正の相関といえば

- $x$ が増加すると  $y$ も増加する
- 相関係数は正になる



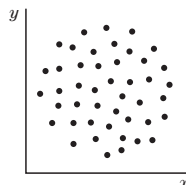
## 負の相関といえば

- $x$ が増加すると  $y$ が減少する
- 相関係数は負になる



## 無相関といえば

- 散布図は特定の傾向をもたない
- 相関係数はゼロになる



問78

**参考** 固定費の合計が500百万円なので、損益分岐点が500以下になることはありえない、と考えれば、すぐに正解がわかるね。



## 解答

問76 **ウ**    問77 **エ**  
問78 **エ**



**問 79** ソフトウェア開発を外部業者へ委託する際に、納品後一定の期間内に発見された不具合を無償で修復してもらう根拠となる項目として、契約書に記載するものはどれか。

- |                                |                 |
|--------------------------------|-----------------|
| <b>ア</b> 瑕疵 <sup>かし</sup> 担保責任 | <b>イ</b> 善管注意義務 |
| <b>ウ</b> 損害賠償責任                | <b>エ</b> 秘密保持義務 |



**問 80** 労働者派遣における派遣元の責任はどれか。

- ア** 派遣先での時間外労働に関する法令上の届出
- イ** 派遣労働者に指示する業務の遂行状況の管理
- ウ** 派遣労働者の休日や休憩時間の適切な取得に関する管理
- エ** 派遣労働者の日々の就業に必要な職場環境の整備

## 問79 瑕疵担保責任

すでに納品した製品に、隠れた欠陥（瑕疵）が後から見つかった場合、その欠陥について納品した側が責任を負うことを**瑕疵担保責任**といいます。瑕疵担保責任の期間や修復について契約書に記載されていれば、納品後に発見された不具合を無償で修復してもらう根拠になります。正解は**ア**です。

- **ア** 正解です。
- × **イ** **善管注意義務**とは、業務を任された人が通常払うべき注意義務のことです。したがって注意義務の範囲で生じた不具合であれば、責任を負わない場合もあります。
- × **ウ** **損害賠償責任**では、損害を金銭で賠償してもらうことになるので、不具合を修復してもらう根拠とはなりません。
- × **エ** **秘密保持義務**は、不具合の修復とは関係ありません。

## 問80 労働者派遣

派遣労働者は派遣先企業の指揮命令の下で働くので、派遣労働者に時間外労働を指示するのは派遣先です。ただし、労働者に時間外労働をさせるには、法令上、労使間で協定を結び、これを所轄の労働基準監督署に届け出なければなりません（36 協定）。派遣労働者の雇用主は派遣元なので、この届出は派遣元が行います。

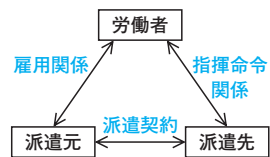
- **ア** 正解です。
- × **イ** 業務の遂行上の管理は派遣先が行います。
- × **ウ** 休日や休憩時間の取得については、就業先である派遣先が管理します。
- × **エ** 職場環境の整備は派遣先の責任です。



## 午前のカギ

### 覚えよう！ 問80

労働者派遣といえば



解答

問79 **ア** 問80 **ア**



# 平成 26 年度 春 午後問題

次の問 1 は必須問題です。必ず解答してください。

**問 1** 情報資産についてのリスクアセスメントに関する次の記述を読んで、設問 1 ～ 3 に答えよ。

Z 社は、従業員数が 500 の中堅 SI ベンダである。Z 社では、プロジェクト開始前に、プロジェクトで扱う情報資産について、図 1 に示す自社で定めた手順に従って、リスクアセスメントを実施している。このたび、新規に受注したプロジェクト Y に対して、リスクアセスメントを実施することになった。

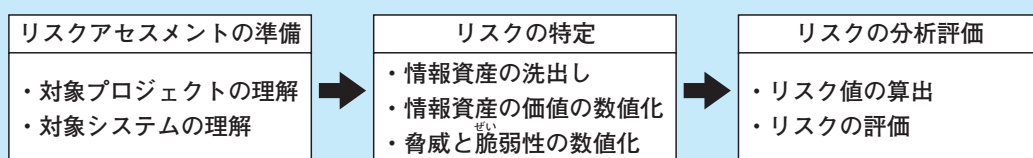


図 1 Z 社のリスクアセスメントの手順

〔プロジェクト Y の説明（抜粋）〕

- 顧客が利用する購買システムを開発する。
- 開発で利用するテストデータは顧客から提供される。
- 顧客のテストデータを格納した顧客の USB メモリを、プロジェクトメンバが顧客から受け取って自社に持ち帰り、顧客のテストデータを開発用サーバに複写後、USB メモリから削除する。
- Z 社から顧客の事務所を訪問するのに、電車で 1 時間 30 分ほど要する。
- 開発用 PC でプログラムを開発し、適宜、開発用サーバにアップロードする。

〔Z 社の開発環境（抜粋）〕

- プログラムの開発には、開発用サーバと開発用 PC を利用する。
- 開発用サーバは、施錠されたサーバールームに設置されている。
- 開発用サーバは、アクセス管理がされており、プロジェクトメンバとシステム管理者だけがアクセスできる。
- 開発用 PC は、プロジェクト開始時にシステム部から各プロジェクトメンバに貸与され、プロジェクト終了時に返却される。

〔Z 社の開発標準（抜粋）〕

- 開発時、プロジェクトメンバは顧客のテストデータのうち必要なものだけを、開発用サーバから自分の開発用 PC にダウンロードし、不要になったら削除する。
- プロジェクト終了時に、プロジェクトマネージャは開発用サーバの顧客のテストデータを削

除し、全ての開発用 PC から顧客のテストデータが削除されていることを確認する。

### 〔Z 社のリスク値算出方法〕

Z 社では、各情報資産のリスク値を、次の式で算出する。

$$\text{リスク値} = \text{情報資産の価値} \times \text{脅威} \times \text{脆弱性}$$

ここで、“情報資産の価値”とは情報資産が損なわれたときの影響の大きさを意味し、機密性（以下、C という）、完全性（以下、I という）、可用性（以下、A という）の観点に対して、影響の大きさをそれぞれ 1～3 の値で評価する。“脅威”は、発生の可能性の大きさを 1～3 の値で評価する。“脆弱性”は、脅威が発生した場合に被害が顕在化する度合いの大きさを 1～3 の値で評価する。ここで、各 1～3 の値は大きい場合を 3、小さい場合を 1 とする。

C, I, A ごとに算出したリスク値が全て 12 以下ならばリスクを受容し、そうでないならば追加のリスク対策を実施することになっている。

### 〔リスクの特定〕

#### ① 情報資産の洗出し

プロジェクト Y で扱う情報資産の洗出しを行った。その結果を、表 1 に示す。

表 1 情報資産の洗出し結果

No.	情報資産	作成又は取得	保管場所	廃棄
⋮				
3	開発用サーバ上の開発中のプログラム	プロジェクトメンバーが開発用サーバにアップロードする	開発用サーバ	プロジェクト終了時に削除する
4	顧客のテストデータ	顧客のUSBメモリで受領して、開発用サーバに複写する	顧客のUSBメモリ、開発用サーバ及び開発用PC	
⋮				

注記 網掛けの部分は表示していない。“…”は表示の省略を示している。

#### ② 情報資産の価値の数値化

表 1 の各情報資産に対して、C, I, A のそれぞれについてその価値を評価した値と評価理由を、表 2 に示す。

表 2 情報資産の価値と評価理由

No.	情報資産	C	I	A	価値の評価理由
⋮					
3	開発用サーバ上の開発中のプログラム	3	3	3	(i) 開発中のプログラムが利用できない場合、プロジェクトの進捗に影響を与える (ii) 社外に漏れた場合、顧客からの信頼を失う (iii) 版管理が行われない場合、不整合によって、プロジェクトの進捗に影響を与える
4	顧客のテストデータ	3	2	1	
⋮					

注記 網掛けの部分は表示していない。“…”は表示の省略を示している。

### ③ 脅威の数値化

表2の情報資産のうち、情報資産 No.4（顧客のテストデータ）について、脅威の内容と脅威の値を、表3に示す。

表3 情報資産 No.4 の脅威の内容と値

No.	脅威ID	脅威の内容	値
4	T1	顧客のテストデータを格納した顧客のUSBメモリを自社に持ち帰る途中で紛失する	3
	T2	開発用サーバが外部から不正アクセスされて顧客のテストデータが盗み出される	1
	T3	ウイルス感染によって顧客のテストデータの破壊又は漏えいが発生する	2
	T4	開発用サーバに複写後、顧客のUSBメモリから顧客のテストデータが漏えいする	3
	T5	テスト終了後、不要になった顧客のテストデータが開発用PCから漏えいする	2
	T6	プロジェクトメンバ又はシステム管理者が顧客のテストデータを開発用サーバから取り出してサーバールームから持ち出す	1
	T7	開発用サーバから顧客のテストデータが滅失する	1

### ④ 脅威に対する脆弱性の数値化

表3の各脅威に対する脆弱性の低減策と脆弱性の値を、表4に示す。脆弱性の値は、システム、規則又は運用で、二つ以上対策済みなら1、一つだけなら2、未対策は3とする。

表4 表3の脅威に対する脆弱性の低減策と値

脅威ID	脆弱性ID	脆弱性の低減策	値
T1	Z1	・顧客のUSBメモリに顧客のテストデータを保存するときに暗号化してもらう	2
T2	Z2	・脆弱性に対する対策なし	3
T3	Z3	・開発用サーバと開発用PCにウイルス対策ソフトを導入し、ウイルス定義ファイルを自動更新する ・顧客のUSBメモリをウイルスチェックした後に顧客のテストデータを開発用サーバに複写する	1
T4	Z4	・顧客のUSBメモリから顧客のテストデータが削除されていることをプロジェクトマネージャが確認する	2
T5	Z5	・開発用PCから顧客のテストデータが削除されていることをプロジェクトマネージャが確認する	2
T6	Z6	・社員証による入退室管理を行う ・サーバールームに監視カメラを設置する	1
T7	Z7	・脆弱性に対する対策なし	3

### 〔リスクの分析評価〕

表2～4を基に情報資産 No.4（顧客のテストデータ）のリスクの分析評価を行い、リスク値を算出した結果を、表5に示す。

表5 情報資産 No.4 のリスク値

No.	情報資産の価値			脅威		脆弱性		リスク値			
	C	I	A	脅威ID	値	脆弱性ID	値	リスク値ID	C	I	A
4	3	2	1	T1	3	Z1	2	R1	18	12	6
				T2	1	Z2	3	R2	9	6	3
				T3	2	Z3	1	R3			
				T4	3	Z4	2	R4			
				T5	2	Z5	2	R5			
				T6	1	Z6	1	R6			
				T7	1	Z7	3	R7			

注記 網掛けの部分は表示していない。

プロジェクト Y のプロジェクトマネージャは、リスクの分析評価の結果からリスク対応計画を作成した。その後、リスク対策を実施した。

**設問 1** 表 2 中の (ii), (iii) は、C, I, A のいずれかの観点から“情報資産の価値”を評価した際の評価理由である。(ii), (iii) に対応する C, I, A の組合せとして適切な答えを、解答群の中から選べ。

解答群

	(ii)	(iii)
<b>ア</b>	A	C
<b>イ</b>	A	I
<b>ウ</b>	C	A
<b>エ</b>	C	I
<b>オ</b>	I	A
<b>カ</b>	I	C

**設問 2** 情報資産 No.4（顧客のテストデータ）に対するリスクの分析評価の結果、追加のリスク対策が必要になる脅威の数として正しい答えを、解答群の中から選べ。

解答群

**ア** 1      **イ** 2      **ウ** 3      **エ** 4

**設問 3** 社内でのセキュリティ事故の発生と対策に関する次の記述中の  に入る適切な答えを、解答群の中から選べ。

プロジェクト Y の終了後、新たに発足したプロジェクト X で利用している開発用 PC に、プロジェクト Y の顧客のテストデータが格納されている、とシステム部に連絡があった。調査した結果、この PC は、プロジェクト Y で利用していた開発用 PC であり、システム部に返却された後に、システム部からプロジェクト X に貸与されたものであることが判明した。そこで、Z 社では、顧

客のテストデータの漏えいというリスクに対処するために、，という対策を追加することにした。

### 解答群

- ア** 開発用サーバのアクセスログをシステム部が定期的に確認する
- イ** 顧客のテストデータを開発用 PC にダウンロードして利用する場合は、管理台帳にダウンロード日、削除日、実施者を記入する
- ウ** 顧客のテストデータを開発用 PC に保存する際に、警告メッセージが表示されるようにする
- エ** プロジェクトごとに新たに開発用サーバを用意する
- オ** プロジェクトメンバーが開発用サーバ上の顧客のテストデータにアクセスする権限を参照だけに設定する
- カ** 返却された開発用 PC は、システム部が全データを完全消去する工程を追加する

## 問 1 午後のカギ

平成 26 年春期試験から、情報セキュリティの問題が必須問題になりました。ただし、本問ではあまりセキュリティの専門知識は要求されません。問題文を読み込んで、論理的に考えて解答を導くことが重要です。

設問 1 情報セキュリティの CIA は、基礎知識として覚えておきましょう。

設問 2 問題文をよく読んで、表 5 の空欄を埋めます。

設問 3 削除の確認をしやすくする対策や、削除漏れがあった場合の対策を解答群から選びます。

### 設問 1 情報セキュリティの CIA

**機密性** (Confidentiality)、**完全性** (Integrity)、**可用性** (Availability) は、「情報セキュリティの CIA」とも呼ばれる三大要素です。

機密性 (C)	秘密が保たれること
完全性 (I)	情報が正確なこと
可用性 (A)	いつでも利用できること

表 2 中の (ii) と (iii) が、上記のどの観点から評価しているかを考えましょう。

(ii) 社外に漏れた場合、顧客からの信頼を失う

(ii) は、開発中のプログラムが「社外に漏れた場合」を心配しています。これは機密性が破られる事態ですから、「C」に該当します。

(iii) 版管理が行われない場合、不整合によって、プロジェクトの進捗に影響を与える

(iii) は、開発中のプログラムに「不整合」が生じた場合を心配しています。これはデータの完全性が破られる事態ですから、「I」に該当します。

以上から、(ii) が「C」、(iii) が「I」の組合せの **エ** が正解です。

### 設問 2 リスク値の算出

まず、情報資産 No.4 (顧客のテストデータ) のリスク値を、C、I、A ごとに算出します。問題文の表 5 にはその一部がすでにまとめられていて、一部が網掛けになっています。

リスク値の算出方法については、問題文の〔Z 社のリスク値算出方法〕に次のように書いてあります。

リスク値＝情報資産の価値 × 脅威 × 脆弱性

表5をみると、たとえば脅威 T1 のリスク値 R1 は、C が 18、I が 12、A が 6 になっています。これらはそれぞれ、

$$\begin{array}{l} \text{情報資産} \quad \text{脅威} \quad \text{脆弱性} \\ \text{C: } 3 \times 3 \times 2 = 18 \\ \text{I: } 2 \times 3 \times 2 = 12 \\ \text{A: } 1 \times 3 \times 2 = 6 \end{array}$$

のように算出したもの、ということがわかります。これを参考に、網掛けになっているリスク値を埋めていきましょう。結果は次のようになります。

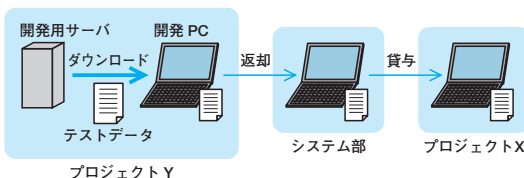
	C	I	A
R1	18	12	6
R2	9	6	3
R3	$3 \times 2 \times 1 = 6$	$2 \times 2 \times 1 = 4$	$1 \times 2 \times 1 = 2$
R4	$3 \times 3 \times 2 = 18$	$2 \times 3 \times 2 = 12$	$1 \times 3 \times 2 = 6$
R5	$3 \times 2 \times 2 = 12$	$2 \times 2 \times 2 = 8$	$1 \times 2 \times 2 = 4$
R6	$3 \times 1 \times 1 = 3$	$2 \times 1 \times 1 = 2$	$1 \times 1 \times 1 = 1$
R7	$3 \times 1 \times 3 = 9$	$2 \times 1 \times 3 = 6$	$1 \times 1 \times 3 = 3$

〔Z社のリスク値算出方法〕によれば、「C、I、Aごとに算出したリスク値が全て12以下ならばリスクを受容し、そうでないならば追加のリスク対策を実施」します。したがって、リスク値が12より大きい項目のあるR1とR4には、追加のリスク対策が必要とわかります。

以上から、追加のリスク対策が必要な脅威の数は「2」です。正解はイです。

### 設問3 顧客のテストデータの漏えい対策

空欄 a, b：今回のセキュリティ事故は、プロジェクトYで利用していた開発用PCが、顧客のテストデータが消去されないまま返却され、それをシステム部がプロジェクトXにそのまま貸与したために生じたものです。



〔Z社の開発標準（抜粋）〕によれば、顧客のテスト

データは開発用サーバから、プロジェクトメンバ各自が自分の開発用PCにダウンロードし、不要になったら削除します。また、プロジェクト終了時には、プロジェクトマネージャが全ての開発用PCから顧客のテストデータが削除されていることを確認します。

今回の事故は、プロジェクトYのプロジェクトマネージャが、この確認を怠ったことが直接の原因です。したがって追加の対策としては、

- ・プロジェクトマネージャが確認作業を確実に行えるようにする
- ・万一確認漏れが生じた場合でも、確実に削除されるような手順にする

といった対策が有効と考えられます。以上を踏まえて、解答群を順番に検討しましょう。空欄は2つあるので、解答群から該当するものを2つ選びます。

- × ア 開発用サーバのアクセスログを確認しても、開発用PCのデータが削除されたかどうかは確認できません。
- イ 正解です。管理台帳に削除日を記入するにすれば、削除日の記入がないPCをチェックすることで削除の確認ができます。
- × ウ 保存の際に警告メッセージを表示しても、後で確実に削除される保証はあいません。
- × エ 開発用サーバを新しくしても、開発用PCにデータが残る可能性はあります。
- × オ 参照権限があれば、開発用PCに保存することは可能です。
- カ 正解です。返却された開発用PCのデータを、システム部が消去するようにすれば、今回のような事故は発生しません。

以上から、空欄 a, b には「イ」と「カ」が順不同で入ります。

### 解答

- 設問1 イ  
設問2 イ  
設問3 a, b — イ, カ（順不同）

次の問 2 から問 7 までの 6 問については、この中から 4 問を選択し、選択した問題については、答案用紙の選択欄の(選)をマークして解答してください。

なお、5 問以上マークした場合には、はじめの 4 問について採点します。

## 問 2 機械語命令に関する次の記述を読んで、設問 1, 2 に答えよ。

命令語の形式を、図 1 に示す。

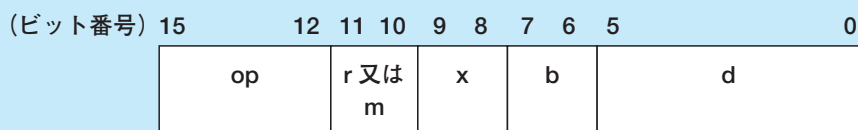


図 1 命令語の形式

図 1 で使用している記号の説明を、表 1 に示す。数字の末尾に h が付いているものは 16 進数表記である。

表 1 記号の説明

記号	説明
op	命令コード
r	レジスタ番号
m	分岐命令で分岐の判定に使用する値
x	指標レジスタとして使用するレジスタの番号
b	ベースレジスタとして使用するレジスタの番号
d	00h~3Fhで示されるアドレスの変位

- (1) この命令語を実行するコンピュータの 1 語は 16 ビットであり、1 語長のレジスタを 4 個（レジスタ番号 0～3）と、命令の実行結果によって値が設定される 2 ビットの条件コードレジスタ（以下、CC という）をもつ。
- (2) 実効アドレスは、表 2 に示す式で算出される。ここで、(x) と (b) は、それぞれ x と b で指定されるレジスタに設定されている内容（以下、レジスタの内容という）を示す。

表 2 実効アドレスの算出式

x	b	実効アドレス
0	0	d
0	0以外	(b) + d
0以外	0	(x) + d
0以外	0以外	(x) + (b) + d

- (3) 命令コード（一部）を表 3 に示す。



表3 命令コード（一部）

命令コード	機能	CCの設定
1	実効アドレスに格納されている内容とrで指定されるレジスタの内容の論理和を、rで指定されるレジスタに設定する。	○
2	実効アドレスに格納されている内容とrで指定されるレジスタの内容の論理積を、rで指定されるレジスタに設定する。	○
3	実効アドレスに格納されている内容とrで指定されるレジスタの内容の排他的論理和を、rで指定されるレジスタに設定する。	○
4	mとCCの論理積を求め、結果が00でなければ実効アドレスに分岐する。結果が00であれば、何もしない。	—

注記 ○：論理演算の結果が0（全てのビットが0）のときは2進数の10、それ以外のときは2進数の01が設定される。  
—：実行前の値が保持される。

**設問1** レジスタの内容が図2に示す値のとき、次の命令の実効アドレスとして正しい答えを、解答群の中から選べ。

命令： 1983h

レジスタ番号	内容
0	0004h
1	0003h
2	0002h
3	0001h

図2 レジスタの内容

解答群

ア 0001h    イ 0002h    ウ 0003h    エ 0004h    オ 0008h    カ 000Ah

**設問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。ここで、レジスタと主記憶装置の内容は、図3と図4のとおりとする。

なお、解答は重複して選んでもよい。

レジスタ番号	内容
0	0004h
1	0003h
2	0002h
3	0001h

図3 レジスタの内容

番地	内容
0001h	0001h
0002h	000Fh
0003h	0003h
0004h	0004h
⋮	⋮
0010h	12C0h
0011h	24C0h
0012h	38C2h
0013h	4815h
0014h	4C16h
0015h	18C3h
0016h	28C1h
⋮	⋮

図4 主記憶装置の内容

図 3, 4 に示した状態で、主記憶装置に格納されているプログラムを 0010h 番地から実行する。  
0011h 番地の命令を実行した直後のレジスタ番号 0 の内容は  になり、レジスタ番号 1 の内容は  になる。  
0013h 番地の分岐命令では 。0016h 番地の命令を実行した直後のレジスタ番号 2 の内容は  になる。

a, b, d に関する解答群

☐ ア 0001h

☐ イ 0002h

☐ ウ 0003h

☐ エ 0004h

☐ オ 0005h

☐ カ 0006h

☐ キ 0007h

☐ ク 0008h

☐ ケ 0009h

c に関する解答群

☐ ア 分岐しない

☐ イ 分岐する

## 問 2 午後のカギ

機械語については、最近あまり馴染みのない人が多いかも知れませんが、その仕組みを理解しておくことは重要です。機械語の解析は少し手間がかかるので、本問で慣れておくといよいでしょう。

設問 1 16 進数の命令語を、2 進数に分解して命令語の形式に当てはめます。実効アドレスの算出方法をよく理解してください。

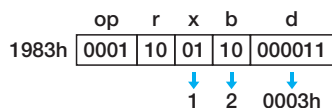
設問 2 命令を 1 つずつ解析して、レジスタの内容の変化をトレースします。

### 設問 1 実効アドレスの算出

命令「1983h」を 2 進数に変換し、図 1 の命令語の形式に当てはめてみましょう。

16 進数： 1 9 8 3 h

2 進数： 0001 1001 1000 0011



実効アドレスを求めるには、命令語の x, b, d の値に注目します。上図のように、x = 1, b = 2, d = 0003h です。表 2 より、x と b が 0 以外のときの実効アドレスは、次に示す式で算出されます。

(x) + (b) + d

(x) と (b) は、それぞれ x と b で指定されるレジスタの内容を示します。つまり、実効アドレスは

レジスタ 1 の内容 + レジスタ 2 の内容 + 0003h

で求められます。図 2 より、レジスタ 1 の内容は 0003h、レジスタ 2 の内容は 0002h ですから、実効アドレスは次のようになります。

0003h + 0002h + 0003h = 0008h

以上から、正解は ☐ オ です。

### 設問 2 機械語命令の実行

主記憶 0010h 番地以降に記憶されている命令を順に実行していきます。

① はじめに 0010h 番地の命令「12C0h」を実行します。  
12C0h を 2 進数に変換し、命令語の形式に当てはめましょう。

	op	r	x	b	d
12C0h	0001	00	10	11	000000
	↓	↓	↓	↓	↓
	1	0	2	3	0000h

**命令コード (op) :** 表 3 より、命令コード 1 番は、実効アドレスに格納されている内容と、レジスタ r の内容の論理和を、レジスタ r に設定します。

**レジスタ番号 (r) :** r は 0 なので、レジスタ 0 の内容 0004h を取り出します。

**実効アドレス :** x = 2, b = 3, d = 0000h なので、実効アドレスはレジスタ 2 の内容 + レジスタ 3 の内容 + 0000h となります。図 3 より、レジスタ 2 の内容は 0002h、レジスタ 3 の内容は 0001h なので、実効アドレスは

$$0002h + 0001h + 0000h = 0003h$$

となります。

**実行結果 :** 主記憶の 0003h 番地の内容は、図 4 より 0003h です。この値と、レジスタ 0 の内容 0004h との論理和を求めます。結果は次のようになります。

$$\begin{array}{r} 0000\ 0000\ 0000\ 0011 \leftarrow 0003h \\ \text{OR } 0000\ 0000\ 0000\ 0100 \leftarrow 0004h \\ \hline 0000\ 0000\ 0000\ 0111 \leftarrow 0007h \end{array}$$

以上から、レジスタ 0 に「0007h」が設定されます。

レジスタ番号	内容
0	0007h
1	0003h
2	0002h
3	0001h

②次に、0011h 番地の命令「24C0h」を実行します。

	op	r	x	b	d
24C0h	0010	01	00	11	000000
	↓	↓	↓	↓	↓
	2	1	0	3	0000h

**命令コード (op) :** 命令コード 2 番は、実効アドレスに格納されている内容と、レジスタ r の内容の論理積を、レジスタ r に設定します。

**レジスタ番号 (r) :** r は 1 なので、レジスタ 1 の内容 0003h を取り出します。

**実効アドレス :** x = 0, b = 3, d = 0000h なので、実効アドレスはレジスタ 3 の内容 + 0000h となります。図 3 より、レジスタ 3 の内容は 0001h なので、実効アドレスは

$$0001h + 0000h = 0001h$$

となります。

**実行結果 :** 主記憶の 0001h 番地の内容は、図 4 より 0001h です。この値と、レジスタ 1 の内容 0003h との論理積を求めます。結果は次のようになります。

$$\begin{array}{r} 0000\ 0000\ 0000\ 0001 \leftarrow 0001h \\ \text{AND } 0000\ 0000\ 0000\ 0011 \leftarrow 0003h \\ \hline 0000\ 0000\ 0000\ 0001 \leftarrow 0001h \end{array}$$

以上から、レジスタ 1 に「0001h」が設定されます。

レジスタ番号	内容
0	0007h
1	0001h
2	0002h
3	0001h

ここまでの結果で、空欄 a には「キ」の「0007h」、空欄 b には「ア」の「0001h」が入るとわかります。

③次に、0012h 番地の命令「38C2h」を実行します。

	op	r	x	b	d
38C2h	0011	10	00	11	000010
	↓	↓	↓	↓	↓
	3	2	0	3	0002h

**命令コード (op) :** 命令コード 3 番は、実効アドレスに格納されている内容と、レジスタ r の内容の排他的論理和を、レジスタ r に設定します。

**レジスタ番号 (r) :** r は 2 なので、レジスタ 2 の内容 0002h を取り出します。

**実効アドレス :** x = 0, b = 3, d = 0002h なので、実効アドレスはレジスタ 3 の内容 + 0002h となります。図 3 より、レジスタ 3 の内容は 0001h なので、実効アドレスは

$$0001h + 0002h = 0003h$$

となります。

**実行結果 :** 主記憶の 0003h 番地の内容は、図 4 より 0003h です。この値と、レジスタ 2 の内容 0002h との排他的論理和（入力が 2 つとも同じ値のとき 0、異なる値のとき 1 になる）を求めます。結果は次のようになります。

$$\begin{array}{r} 0000\ 0000\ 0000\ 0011 \leftarrow 0003h \\ \text{XOR } 0000\ 0000\ 0000\ 0010 \leftarrow 0002h \\ \hline 0000\ 0000\ 0000\ 0001 \leftarrow 0001h \end{array}$$

以上から、レジスタ 2 に「0001h」が設定されます。

レジスタ番号	内容
0	0007h
1	0001h
2	0001h
3	0001h

④ 0013h 番地の命令「4815h」を実行します。

	op	m	x	b	d
4815h	0100	10	00	00	010101
	↓		↓	↓	↓
	4		0	0	0015h

**命令コード (op) :** 命令コード 4 番は、m と CC の論理積を求め、結果が 00 でなければ実効アドレスに分岐します。

**実効アドレス :** x = 0, b = 0, d = 0015h なので、実効アドレスは 0015h となります。

**実行結果 :** m の値は 2 進数で 10 です。また、CC の値は 1 つ前の命令の結果が 0 のとき 10、それ以外のとき 01 となります (表 3 の注記)。0012h 番地の命令の実行結果は「0001h」でしたから、CC の値は 01 となります。したがって、論理積の結果は

```

10
AND 01
----
00

```

となり、分岐はしません。空欄 c は **ア** です。

⑤ 0014h 番地の命令「4C16h」を実行します。

	op	m	x	b	d
4C16h	0100	11	00	00	010110
	↓		↓	↓	↓
	4		0	0	0016h

**命令コード (op) :** 命令コード 4 番は、m と CC の論理積を求め、結果が 00 でなければ実効アドレスに分岐します。

**実効アドレス :** x = 0, b = 0, d = 0016h なので、実効アドレスは 0016h となります。

**実行結果 :** m の値は 2 進数で 11 です。また、CC の値は前と同じ 01 のままです。したがって、

```

11
AND 01
----
01

```

となり、0016h 番地に分岐します。

⑥ 0016h 番地の命令「28C1h」を実行します。

	op	r	x	b	d
28C1h	0010	10	00	11	000001
	↓	↓	↓	↓	↓
	2	2	0	3	0001h

**命令コード (op) :** 命令コード 2 番は、実効アドレスに格納されている内容と、レジスタ r の内容の論理積を、レジスタ r に設定します。

**レジスタ番号 (r) :** r は 2 なので、レジスタ 2 の現在の内容 0001h を取り出します (解説中の③参照)。

**実効アドレス :** x = 0, b = 3, d = 0001h なので、実効アドレスはレジスタ 3 の内容 + 0001h となります。レジスタ 3 の内容は 0001h なので、実効アドレスは

0001h + 0001h = 0002h

となります。

**実行結果 :** 主記憶の 0002h 番地の内容は、図 4 より 000Fh です。この値と、レジスタ 2 の現在の内容 0001h との論理積を求めます。結果は次のようになります。

```

0000 0000 0000 1111 ← 000Fh
AND 0000 0000 0000 0001 ← 0001h
-----
0000 0000 0000 0001 ← 0001h

```

以上から、レジスタ 2 に「0001h」が設定されます。空欄 d は **ア** です。

レジスタ番号	内容
0	0007h
1	0001h
2	0001h
3	0001h

解答

設問 1 **オ**

設問 2 a - **キ**, b - **ア**, c - **ア**,  
d - **ア**

### 問 3

プログラムの並列実行に関する次の記述を読んで、設問 1 ～ 3 に答えよ。

プログラムを並列に実行する方法として、スレッドを使用した並列実行がある。スレッドを使用した並列実行では、プログラムの中で並列実行が可能な部分を抽出して複数の処理に分割し、分割した処理を別々のスレッドとして同時に実行する。

スレッドによる並列実行の例を、図 1 に示す。プログラム A は、一つのプロセスとして実行され、データ作成、計算処理、結果出力の順に処理が行われる。ここで、計算処理は  $n$  個の処理に分割して並列実行が可能であり、分割した処理を異なるスレッドで並列に実行した結果は、スレッドを使わずに計算処理した実行結果と同じであるとする。計算処理では、スレッドの生成と実行を行い、全てのスレッドの終了を同期処理で待つ。各スレッドでは、分割した計算処理（部分計算処理）を行う。

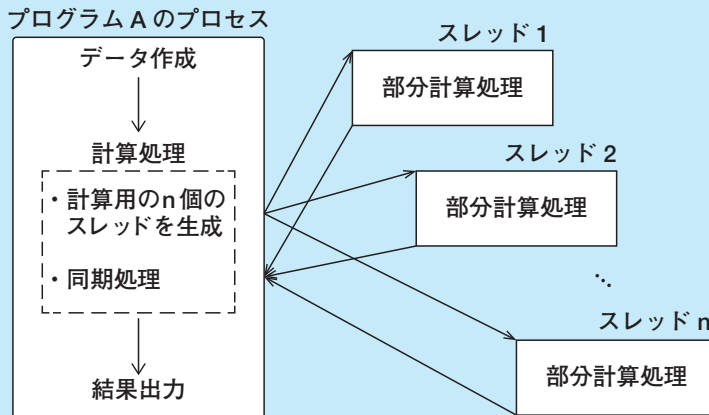


図 1 スレッドによる並列実行の例

プログラム A を実行するコンピュータの構成は、性能が同じである複数の CPU と各 CPU からアクセス可能な共有メモリで構成されているマルチプロセッサとする。生成された複数のスレッドは、異なる CPU に割り当てられて同時に実行され、各スレッドは共有メモリでデータを共有する。マルチプロセッサにおけるスレッドと CPU、共有メモリの関係を、図 2 に示す。ここで、データを転送するバスの競合は無いものとする。

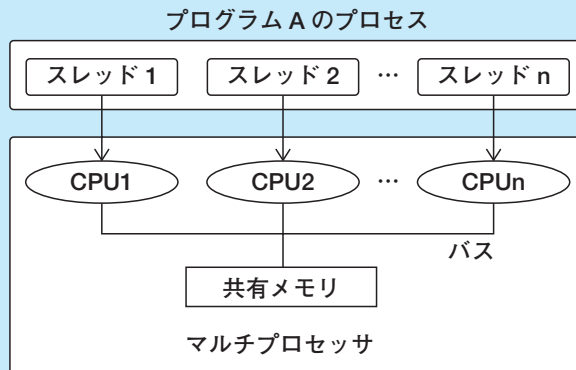


図 2 マルチプロセッサにおけるスレッド実行

**設問 1** 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。

マルチプロセッサによる並列実行で得られる理想的な高速化率  $E$  は、次の式で求められる。

$$E = \frac{1}{1 - r + \left(\frac{r}{n}\right)}$$

$n$  : CPU の数 ( $n \geq 1$ )

$r$  : 対象とする処理のうち、並列実行が可能な部分の処理時間の割合 ( $0 \leq r \leq 1$ )

図 1 のプログラム A において、データ作成、計算処理、結果出力の処理時間の割合が 7 : 90 : 3 の場合、単一の CPU で実行したときと比べた高速化率を 5 以上にするには、CPU が最低  a  個必要である。ここで、スレッドの生成処理などの並列実行に伴うオーバーヘッドは考慮しない。

a に関する解答群

- |                              |                              |                               |
|------------------------------|------------------------------|-------------------------------|
| <input type="checkbox"/> ア 5 | <input type="checkbox"/> イ 6 | <input type="checkbox"/> ウ 7  |
| <input type="checkbox"/> エ 8 | <input type="checkbox"/> オ 9 | <input type="checkbox"/> カ 10 |

**設問 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。

プログラムの一部を複数の処理に分割して並列に実行するためには、プログラムの中から並列実行が可能な部分を抽出する必要がある。並列実行が可能なループの例を、図 3 に示す。図 3 は、 $i$  のループに関してループを四つに分割し、分割したそれぞれのループの処理をスレッドとして並列実行する場合である。ここで、配列のデータはスレッド間で共有され、変数  $i$  はスレッドごとに確保されるものとする。

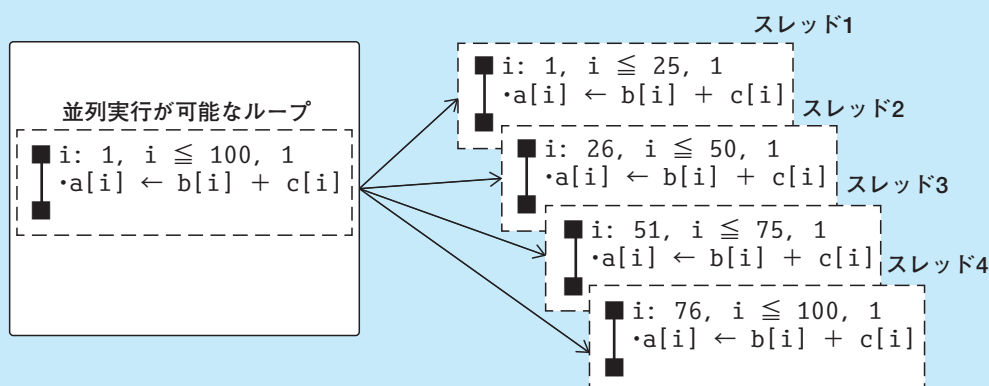


図 3 並列実行が可能なループの例

プログラムの中から並列実行が可能な部分を抽出する場合、並列に実行してもデータの更新と参照の順序が変化しないことを保証する必要がある。図 4 に示すプログラム 1 ~ 3 を、 $i$  のループに関して複数のループに分割し、分割したそれぞれのループの処理を並列に実行する場合の並列実行可能性について考える。ここで、配列は十分に大きいものとする。

プログラム 1 は、ループの中で b，並列実行できない。プログラム 2 は、ループの中で c，並列実行できない。プログラム 3 は、 $m$  の値が不明の場合には並列実行できないが、d であることが保証されていれば並列実行は可能である。

プログラム1

```

■ i: 2, i ≤ n, 1
  |
  · a[i] ← a[i-1] + b[i]
■

```

プログラム2

```

■ i: 1, i ≤ n, 1
  |
  · a[i] ← a[i+1] + b[i]
■

```

プログラム3

```

■ i: 1, i ≤ n, 1
  |
  · a[i] ← a[i+m] + b[i]
■

```

図 4 ループのプログラム 1～3

#### b, c に関する解答群

- ア 更新した値が次の繰返しで参照されるので
- イ 更新した値が次の繰返しで再び更新されるので
- ウ 参照した値が次の繰返しで更新されるので
- エ 参照した値が次の繰返しで再び参照されるので

#### d に関する解答群

- ア  $m \geq 0$     イ  $m \geq n$     ウ  $m \leq 0$
- エ  $m \leq n$

**設問 3** 図 5 に示すプログラム 4 は、配列  $a$  で更新する要素を示すインデックスの値が配列  $ip$  で間接的に決定される。この配列  $a$  のような更新を含むプログラムは、配列  $ip$  の値によっては並列実行できない場合があるので注意が必要である。プログラム 4 を、図 5 のように  $i$  のループに関して複数のループに分割し、分割したそれぞれのループの処理をスレッドで並列実行するとき、並列実行可能な  $ip[i]$  ( $i = 1, 2, \dots, 20$ ) の値として適切な答えを、解答群の中から選べ。



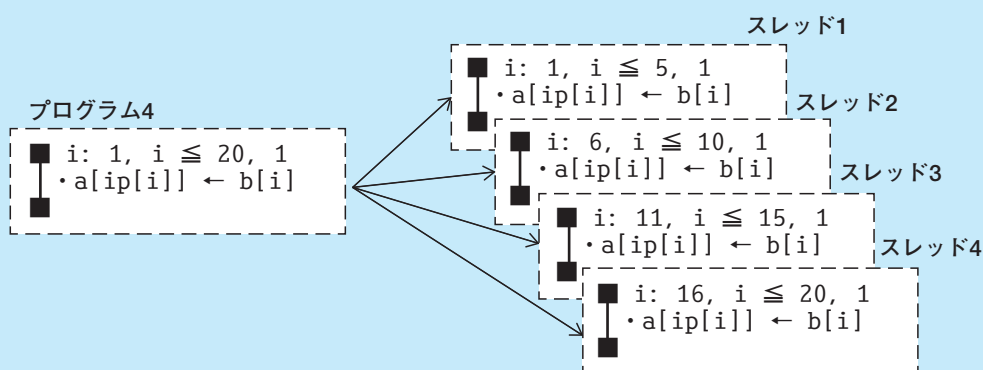


図5 プログラム4の並列実行

解答群

	ip																			
i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
ア	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
イ	2	17	14	10	7	19	16	13	9	6	20	15	12	8	5	18	1	11	3	4
ウ	10	5	8	13	4	8	3	17	2	3	15	13	15	13	1	6	11	18	16	3
エ	20	17	14	10	7	19	16	13	9	6	18	15	12	8	5	17	14	11	7	4

### 問3 午後のカギ

プログラムの並列実行に関する問題です。一連の処理を複数に分割して並列に実行すると、実行時間は短縮されますが、分割した各部分の実行順序は不定になります。そのため、処理によっては不整合が生じる場合があることを理解しましょう。

設問1 問題文の高速化率の式に数値を代入します。

設問2 繰返し処理を分解して考えてみましょう。

設問3 更新する配列の要素が、各スレッド間で重複しないものを選びます。

#### 設問1 高速化率とCPU数

空欄 a：プログラム A において、並列実行が可能なのは「計算処理」の部分だけです。プログラム全体に占める計算処理の処理時間の割合  $r$  は、

$$r = \frac{90}{7 + 90 + 3} = 0.9$$

です。問題文にある高速化率の式に、 $r = 0.9$ 、 $E = 5$  を当てはめると、次のようになります。

$$\frac{1}{1 - 0.9 + \left(\frac{0.9}{n}\right)} = 5$$

上の式から、CPU の個数  $n$  を求めます。

$$1 - 0.9 + \left(\frac{0.9}{n}\right) = \frac{1}{5}$$

$$\frac{0.9}{n} = 0.1 \rightarrow n = 9$$

以上から、高速化率を5以上にするために、CPU の個数は最低9個必要です。空欄 a は **オ** です。

## 設問2 並列実行可能性

空欄 b：プログラム 1 では、

```
i = 2 のとき : a[2] ← a[1] + b[2]
i = 3 のとき : a[3] ← a[2] + b[3]
i = 4 のとき : a[4] ← a[3] + b[4]
      ⋮           ⋮
```

のように更新が行われます。たとえば  $i = 3$  のときは、 $a[3]$  の値を更新するために  $a[2]$  の値を参照しますが、 $a[2]$  の値は  $i = 2$  のときに更新されたものです。このように、前回更新した値が次の繰返しで参照されます。空欄 b は **ア** です。

空欄 c：プログラム 2 では、

```
i = 1 のとき : a[1] ← a[2] + b[1]
i = 2 のとき : a[2] ← a[3] + b[2]
i = 3 のとき : a[3] ← a[4] + b[3]
      ⋮           ⋮
```

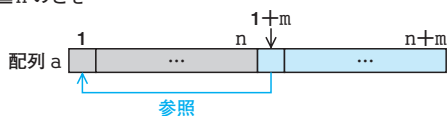
のように更新が行われます。たとえば  $i = 1$  のときに参照した  $a[2]$  の値は、 $i = 2$  のときに更新されます。このように、前回参照した値が次の繰返しで更新されます。空欄 c は **ウ** です。

空欄 d：プログラム 3 では、

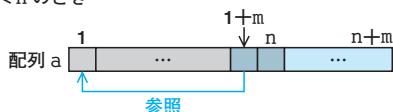
```
i = 1 のとき : a[1] ← a[1 + m] + b[1]
i = 2 のとき : a[2] ← a[2 + m] + b[2]
i = 3 のとき : a[3] ← a[3 + m] + b[3]
      ⋮           ⋮
i = n のとき : a[n] ← a[n + m] + b[n]
```

のように更新が行われます。 $i$  の最大値は  $n$  なので、 $m$  の値が  $n$  より小さいと、一度参照した値が後から更新されてしまいます。ただし、 $m$  の値が  $n$  以上であれば、更新される範囲と参照される範囲は重なりません。

○  $m \geq n$  のとき



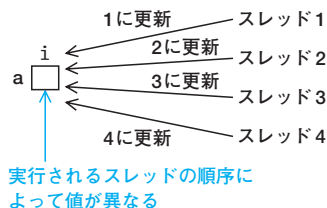
×  $m < n$  のとき



以上から、空欄 d には **イ** の「 $m \geq n$ 」が入ります。

## 設問3 プログラム 4 の並列実行

並列実行される異なるスレッドが同じ要素を更新する場合、スレッドの実行順序によって結果が異なってしまいます。



ただし、更新する要素がスレッド同士で重複しなければ、並列実行が可能です。

プログラム 4 では、配列  $a$  の要素が、

```
i = 1 のとき : a[ip[1]] ← b[1]
i = 2 のとき : a[ip[2]] ← b[2]
i = 3 のとき : a[ip[3]] ← b[3]
      ⋮           ⋮
i = n のとき : a[ip[20]] ← b[n]
```

のように更新されます。したがって、配列  $ip$  の内容が各スレッドで重複しなければ、並列実行が可能です。

解答群の選択肢をひとつずつ検討しましょう。

- × **ア** 各スレッドに 1～5 の値があるので、4 つのスレッドすべてで  $a[1] \sim a[5]$  が更新されます。並行実行はできません。
- **イ** 4 つのスレッドで更新する要素が重なっていないので、並列実行が可能です。
- × **ウ** 8, 13, 3 が複数のスレッドにあるため、 $a[8]$ ,  $a[13]$ ,  $a[3]$  の更新が重複します。並行実行はできません。
- × **エ** 17, 14, 7 がスレッド 1 とスレッド 4 にあるため、 $a[17]$ ,  $a[14]$ ,  $a[7]$  の更新が重複します。並行実行はできません。

○ 解答 ○

設問 1 a－**オ**

設問 2 b－**ア**, c－**ウ**, d－**イ**

設問 3 **イ**

**問 4** ネットワークにおけるスループットの改善に関する次の記述を読んで、設問 1, 2 に答えよ。

X 社は、東京に本社を、札幌、大阪、広島に営業所をもっている。X 社は、広域 Ethernet を利用した企業ネットワークを構築済みである。X 社のネットワーク構成を図 1 に示す。ここで、本社と営業所との間のファイル転送時間に注目し、本社内及び営業所内のファイル転送時間は考慮しない。

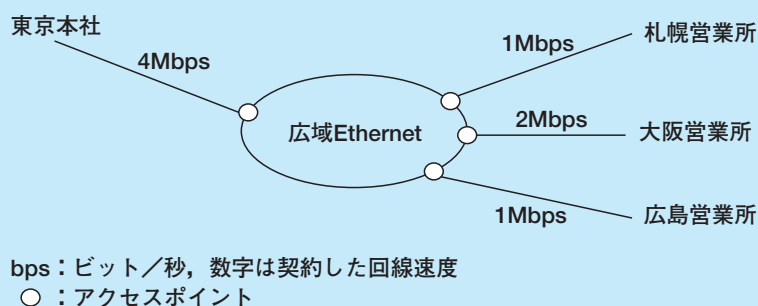


図 1 X 社のネットワーク構成

〔X 社のファイル転送〕

X 社では、各業務システムにおいて、東京本社内にあるファイルサーバ上のファイルを各営業所内の利用者 PC（以下、PC という）から利用している。各営業所からファイルサーバには広域 Ethernet 経由でアクセスしている。①広域 Ethernet はデータリンク層で接続するサービスであり、各営業所内の LAN は広域 Ethernet にアクセスポイントで接続している。

ファイルサーバと PC との間でファイル転送を行う際には、図 2 に示すように、PC からファイルサーバに対して読出し要求を行う。ファイルサーバは、ファイルを固定サイズのプロック（以下、データブロックという）に区切って、PC へ送信することを繰り返す。2 回目以降は、PC からの読出し要求に確認応答が含まれており、ファイルサーバは、PC から確認応答が届くまで次のデータブロックを送信することができない。PC が読出し要求をファイルサーバへ送出し始めてから、ファイルサーバの送信したデータブロックが届き始めるまでを応答時間と呼ぶ。この応答時間は、札幌又は広島営業所と東京本社との間が 45 ミリ秒、大阪営業所と東京本社との間が 24 ミリ秒であった。ここで、通信障害による再送はないものとする。

このネットワーク構成では、広島営業所内の PC と東京本社のファイルサーバ間で、1M バイトのファイルをブロック長が 4k バイトのデータブロックで転送するには、 秒掛かる。

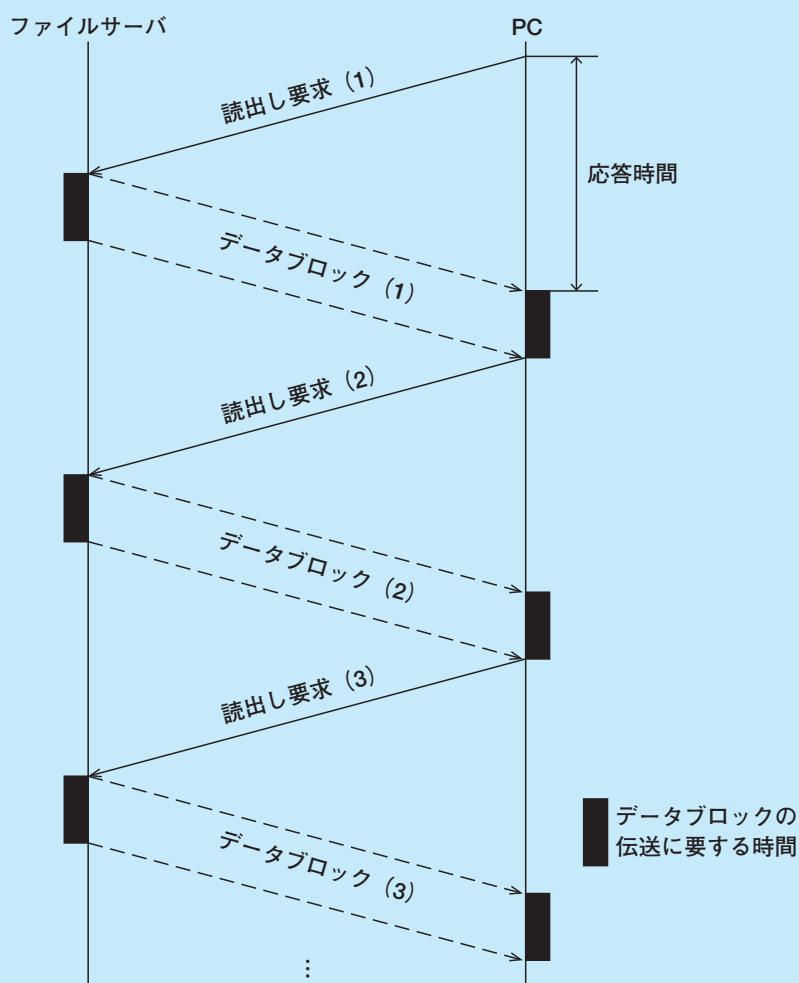


図2 ファイルサーバとPC間の通信シーケンス

#### 〔接続装置の検討〕

図3に示すように、ファイル転送の高速化のために、広域Ethernetへの接続装置の導入を検討することになった。接続装置は、本社と各営業所にそれぞれ設置され、広域Ethernetを挟んで対向して使用される。

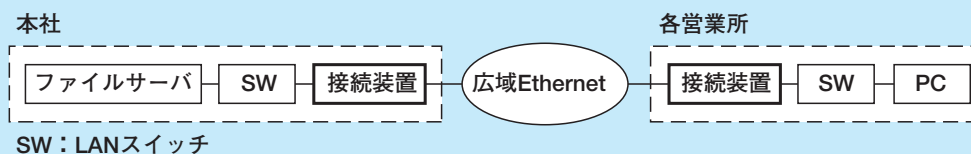


図3 本社と営業所間における接続装置の設置

この接続装置は、PCに代わって読み出し要求をファイルサーバへ送ることができ、ファイルサーバは連続してデータブロックを送信できるようになる。この接続装置を利用した場合のファイル転送の通信シーケンスを図4に示す。

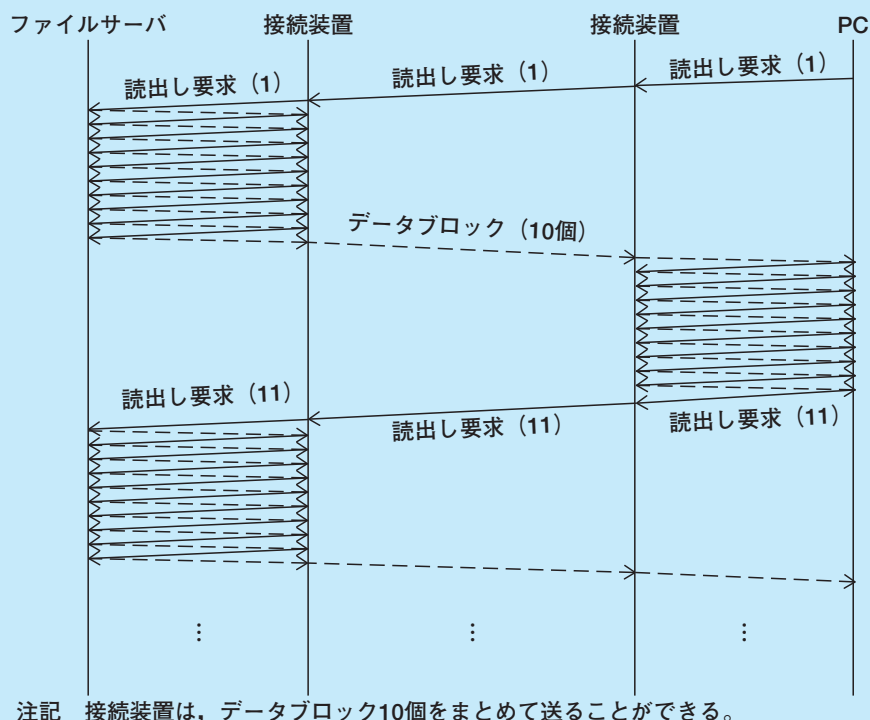


図4 接続装置を利用したファイル転送の通信シーケンス

接続装置を導入すると、札幌営業所内のPCと東京本社のファイルサーバとの間で、1Mバイトのファイルをブロック長が4kバイトのデータブロックで転送するには、秒掛かる。ここで、ファイルサーバと接続装置との間及びPCと接続装置との間の、読出し要求とデータブロックの転送時間は考慮しない。

#### 〔アクセス回線速度の検討〕

X社では、業務の都合から、大阪営業所内のPCにおいて1Mバイトのファイルをブロック長が4kバイトのデータブロックで、8秒以内に転送する必要がある。接続装置を導入しない場合、大阪営業所のアクセス回線速度は最低でも、Mbpsで広域Ethernetの契約をする必要があることが分かった。

**設問 1** X社のネットワーク構成において、本文中の下線①の説明文として適切な答えを、解答群の中から選べ。

#### 解答群

- ア** 広域Ethernetのアクセスポイントへは、各営業所及び本社からネットワーク層の機能をもつ装置を経由して接続しなければならない。
- イ** サーバやPCなど、全ての接続機器のネットワーク層のアドレス設定を同一にする必要がある。
- ウ** 社内の業務システムにおいて、様々な通信プロトコルをネットワーク層で利用することができる。

エ X 社では、TCP/IP 以外のプロトコルを使うことができない。

**設問 2** 本文中の   に入れる正しい答えを、解答群の中から選べ。ここで、1M バイト = 1,000k バイトとし、小数第 3 位を四捨五入する。

a の解答群

ア 8.00      イ 10.00      ウ 11.25      エ 19.25

b の解答群

ア 3.14      イ 5.50      ウ 9.13      エ 13.40

c の解答群

ア 3      イ 4      ウ 8      エ 10

## 問 4 午後のカギ

通信プロトコルの仕組みと、データ転送時間の計算に関する問題です。OSI 基本参照モデルに関する知識は午前試験でもよく出題されるので、覚えておきましょう。計算問題は問題文から計算方法を読み取ります。

**設問 1** 一般的な LAN で使われている Ethernet や無線 LAN はデータリンク層、インターネットで使われる IP はネットワーク層の通信プロトコルです。

**設問 2** ネットワーク全体の回線速度は、最も低速な部分の回線速度によって決まります。

### 設問 1 広域 Ethernet

下線①の内容は次のとおりです。

① 広域 Ethernet はデータリンク層で接続するサービスであり、各営業所内の LAN は広域 Ethernet にアクセスポイントで接続している。

**データリンク層**とは、OSI 基本参照モデルの第 2 層で、同一 LAN 内の隣り合ったノード間を接続する層です。ネットワーク全体をデータリンク層の広域 Ethernet で接続するのは、いわば全体で 1 つの LAN を構成しているようなものです。

データリンク層で接続されたネットワークから、インターネットなどの外部ネットワークに接続するには、ネットワーク間を**ネットワーク層**で接続する必要があります。ネットワーク層の代表的なプロトコルと

して、IP があります。

- × **ア** 広域 Ethernet のアクセスポイントへは、ブリッジやスイッチングハブなどのデータリンク層の装置で接続します。ネットワーク層の機能をもつ装置（ルータなど）は、外部ネットワークに接続する場合に必要となります。
- × **イ** IP アドレスなどのネットワーク層のアドレスは、同一 LAN 内にある各端末で重複しないように設定しなければなりません。
- **ウ** 正しい記述です。データリンク層とネットワーク層は独立しているので、データリンク層で接続されたネットワーク上で、様々なネットワーク層の通信プロトコルを利用できます。たとえば、次世代 IP である IPv6 を使うこともできるでしょう。
- × **エ** TCP/IP はデータリンク層より上位で利用さ

れるプロトコルです。上述のとおり、データリンク層とネットワーク層は独立しているので、TCP/IP 以外のプロトコルも利用できます。

**ウ**と**エ**は、一方が誤りならもう一方が正解になるね。



## 設問 2 転送時間の計算

**空欄 a**：問題文の図 2 のように、データブロックを 1 個送るごとに読出し要求を送るので、その度に応答時間に相当する時間がかかります。データのサイズは 1M = 1,000k バイト、ブロック長は 4k バイトなので、データブロックの数は、

$$1,000k \text{ バイト} \div 4k \text{ バイト} = 250 \text{ 個}$$

です。東京・広島間の応答時間は 45 ミリ秒なので、合計では

$$45 \text{ ミリ秒} \times 250 = 11,250 \text{ ミリ秒} = 11.25 \text{ 秒}$$

となります。これに、データの転送にかかる時間を加えます。

図 1 より、ネットワークの回線速度は、広島営業所からアクセスポイントまでが 1Mbps、東京本社までが 4Mbps になっています。転送速度は回線速度の低い経路に従うので、1Mbps になります。

ファイルのサイズは 1M バイト = 8M ビットなので、ファイルの転送にかかる時間は

$$8M \text{ ビット} \div 1Mbps = 8 \text{ 秒}$$

となります。以上から、転送時間は合計で  $11.25 + 8 = 19.25$  秒になります。**空欄 a** は **エ** です。

**空欄 b**：ファイルサイズとブロック長は先ほどと同じなので、データブロック数は 250 個です。ただし、今回はデータブロックを 10 個ずつまとめて送信するので（図 4 の注記）、読出し要求は  $250 \div 10 = 25$  回で済みます。したがって応答時間の合計は、

$$45 \text{ ミリ秒} \times 25 = 1,125 \text{ ミリ秒} = 1.125 \text{ 秒} \rightarrow 1.13 \text{ 秒}$$

となります（小数第 3 位を四捨五入）。

一方、札幌営業所の回線速度は広島営業所と同じ 1Mbps なので、ファイルの転送にかかる時間は前回と同じく 8 秒です。したがって、転送時間は合計で  $1.13 + 8 = 9.13$  秒となります。**空欄 b** は **ウ** です。

**空欄 c**：東京・大阪間の応答時間は 24 ミリ秒なので、データブロック 250 個を送る場合の応答時間の合計は、

$$24 \text{ ミリ秒} \times 250 = 6,000 \text{ ミリ秒} = 6 \text{ 秒}$$

です。ファイルは 8 秒以内に転送する必要があるので、データ転送にかかる時間は  $8 - 6 = 2$  秒以内にしなければなりません。

ファイルサイズは 1M バイト = 8M ビットなので、回線速度を

$$8M \text{ ビット} \div 2 \text{ 秒} = 4Mbps$$

にすれば、応答時間と合わせて 8 秒でファイルを転送できます。以上から、**空欄 c** は **イ** です。

### 解答

設問 1 **ウ**

設問 2 a - **エ**, b - **ウ**, c - **イ**

## 問 5 システム統合に伴うソフトウェア設計に関する次の記述を読んで、設問 1, 2 に答えよ。

C 社は大型の電子工具を製造する中堅メーカーである。このたび、競争力の強化を図るために、小型の電子工具を製造する D 社を吸収合併することにした。

これに伴い、両社の基幹システムを統合することとなった。早期にシステムを稼働させるために、C 社の基幹システム（以下、C 社システムという）に、D 社の基幹システム（以下、D 社システムという）の機能を追加するというシステム統合の方針を決めた。



なお、C社は受注生産方式、D社は見込生産方式と、異なる生産方式を採っているが、D社の吸収合併後も、C社製品、D社製品それぞれの生産方式は変更しない。

### 〔システム統合の方針〕

- (1) C社とD社の両方に存在するシステムについては、C社システムに、D社システムの固有機能を追加する。この固有機能の追加では、D社システムの既存のプログラムをできるだけ再利用する。
- (2) D社システムを構成する各システム間で行っているデータの連携は、統合後のシステムにおいても、システム統合前の仕様を踏襲する。
- (3) D社システムだけにある在庫管理システムはそのまま利用し、在庫管理システムと、他システムの連携に関しては、新規にインタフェースを開発する。

### 〔C社システムに関する説明〕

図1にC社システムの構成を示す。

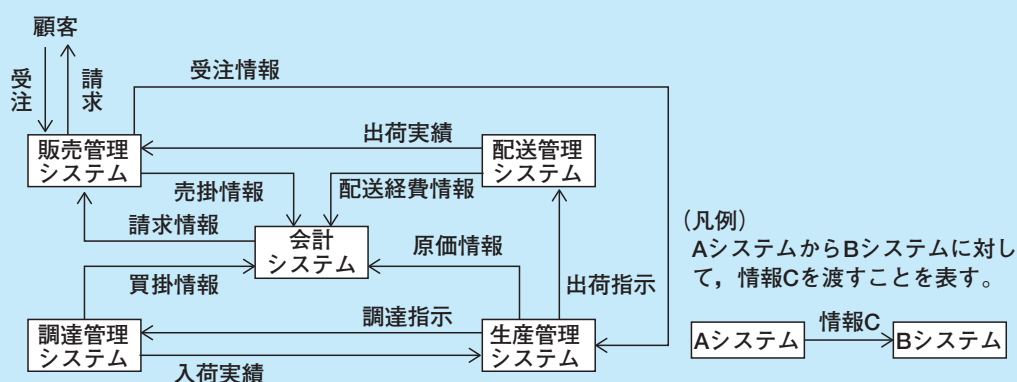


図1 C社システムの構成

#### (1) 販売管理システム

- ① 顧客からEDI（電子データ交換）で受注する。受注の都度、受注情報を生産管理システムに渡す。
- ② 出荷実績に基づき、日次で売上計上し、売掛情報を会計システムに渡す。
- ③ 月初に、顧客に対してEDIで前月の納品分を請求する。

#### (2) 生産管理システム

- ① 受注情報に基づき、週次で製品ごとの生産計画を作成する。生産計画作成時点で、製造に必要な部品の調達指示を調達管理システムに渡す。また、生産計画に基づき、製品の出荷指示を配送管理システムに渡す。
- ② その他、生産工程の作業管理を行い、発生した作業実績（原価情報）を週次で会計システムに渡す。

#### (3) 調達管理システム

- ① 調達指示に基づき、部品を発注する。部品の納品（入荷）時に、入荷実績を生産管理システムに渡す。
- ② 買掛情報を月次で会計システムに渡す。

(4) 配送管理システム

- ① 出荷指示に基づき、週次で配送計画を作成する。製品の出荷後は、出荷実績を日次で販売管理システムに渡す。
- ② 配送経費情報を月次で会計システムに渡す。

(5) 会計システム

- ① 顧客別の請求情報を月次で販売管理システムに渡す。
- ② 原価計算、売掛・買掛管理、一般会計処理などを行う。

〔D 社システムに関する説明〕

図 2 に D 社システムの構成の一部を示す。ここでは、D 社システムのうち、システム統合に関係するものだけを記載している。

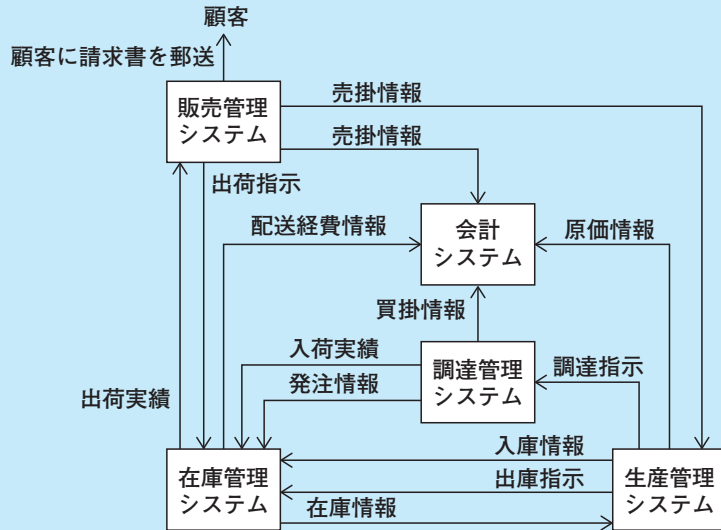


図 2 D 社システムの構成 (一部)

(1) 販売管理システム

- ① 電話、メール及びファックスで注文を受け付け、D 社の社員が端末から受注情報を入力する。この時点で出荷指示を在庫管理システムに渡す。
- ② 月初に、顧客別に前月納品分の請求書を一括で発行し、顧客に郵送する。

(2) 在庫管理システム

- ① 製品及び部品の在庫を管理し、入出庫の都度、在庫情報を更新する。
- ② 出荷指示に基づいて在庫を引き当てる。
- ③ 出荷実績を日次で販売管理システムに渡す。
- ④ 配送経費情報を月次で会計システムに渡す。

(3) 生産管理システム

- ① 販売管理システムの売掛情報と、在庫管理システムの在庫情報を参照し、製品ごとの生産計画を日次で作成する。
- ② 生産計画に基づき、部品の出庫指示を在庫管理システムに、調達指示を調達管理システムに渡す。また、製品の計画時点の入庫情報を在庫管理システムに渡す。

③ 製品の完成時の入庫情報を在庫管理システムに渡す。

(4) 調達管理システム

- ① 調達指示に基づき、部品を発注し、発注情報を在庫管理システムに渡す。
- ② 部品の納品（入荷）時に、入荷実績を在庫管理システムに渡す。

**設問 1** 表 1 は、システム統合に当たり C 社システムに追加開発が必要となる機能を整理した表である。表中の  に入れる正しい答えを、解答群の中から選べ。

表 1 追加開発が必要となる機能

対象のシステム	追加開発が必要となる機能
販売管理システム	・ 端末から受注情報を入力する機能 ・ <input type="text"/>
<input type="text"/> システム	・ <input type="text"/> b <input type="text"/>

注記 網掛けの部分は表示していない。

a に関する解答群

- ☐ ア 会計      ☐ イ 生産管理      ☐ ウ 調達管理      ☐ エ 配送管理

b に関する解答群

- ☐ ア 在庫情報と売掛情報から、製品ごとの生産計画を日次で作成する機能  
☐ イ 作業実績（原価情報）から、月次で原価計算をする機能  
☐ ウ 受注情報から、月別の販売予測を作成する機能  
☐ エ 出荷指示の情報から、週次の配送計画を作成する機能  
☐ オ 請求情報から、月別、顧客別に請求書を作成する機能

**設問 2** システム統合の完了後、C 社が経営課題の一つとして抱えていた納期短縮を目的として、C 社の製品及び部品も、一定量の在庫を保有してリードタイムを短縮することにした。

これに伴い、システム間の情報の渡し方を見直して、在庫管理システムが他システムから在庫の更新要求を受け取るシステム間連携のインタフェースを新たに開発する。

〔在庫管理に関する説明〕

- (1) 在庫は、製品と部品のどちらも、同一体系で一意となる品番コードで実在庫と有効在庫を管理する。
- (2) 実在庫は、実際に保有する製品及び部品の在庫の数量である。
- (3) 製品の有効在庫は、実在庫に翌日までに完成する製品数を加算し、受注に対して引当て済みの製品数を減算した数量である。
- (4) 部品の有効在庫は、実在庫に調達中（未納品）の発注数を加算し、生産計画に対して引当て済みの部品数を減算した数量である。
- (5) 受注時は、有効在庫から受注数を引き当てる。
- (6) 生産計画時に、翌日までに生産予定の製品数を有効在庫として加算する。また、同時に生産

に必要となる部品を有効在庫から引き当てる。

- (7) 部品が不足し、調達先へ発注したときは、発注分を有効在庫として加算する。部品が調達先から納品されたら実在庫として加算する。
- (8) 部品は、生産工程に投入されたら実在庫を減算する。
- (9) 製品の完成時に、製品の実在庫として加算する。
- (10) 製品の出荷時に、製品の実在庫を減算する。

表2は、在庫管理システムが他システムから受け取る在庫更新のインタフェース仕様の説明であり、表3は、他システムと在庫管理システムのシステム間連携について、在庫を更新するタイミングとインタフェースの設定内容を整理した表である。表3の  に入れる正しい答えを、解答群の中から選べ。

表2 在庫更新のインタフェース仕様の説明

設定項目	説明
品番コード	更新を行う在庫品の品番コードを設定する
在庫区分	“有効在庫” か “実在庫” のいずれかを設定する
入出庫区分	“入庫” か “出庫” のいずれかを設定する
数量	在庫を増減させる数量

表3 在庫を更新するタイミングとインタフェースの設定内容

在庫を更新するタイミング		インタフェースの設定内容			
		品番コード	在庫区分	入出庫区分	数量
生産管理システム	生産計画時	更新を行う 在庫品の品 番コード	c		
	生産時				
	製品完成時				
調達管理システム	部品発注時				
	部品納品時		d		
販売管理システム	受注時		e		
配送管理システム	製品出荷時		f		

注記 網掛けの部分は表示していない。

c～fに関する解答群

ア	実在庫	出庫	受注数
イ	実在庫	出庫	出荷数
ウ	実在庫	入庫	生産予定数
エ	実在庫	入庫	納品数
オ	有効在庫	出庫	完成数

力	有効在庫	出庫	受注数
キ	有効在庫	入庫	生産予定数
ク	有効在庫	入庫	納品数

## 問5 午後のカギ

ソフトウェア設計の問題は毎回出題されますが、文章で記述されているシステムの仕様を読み取る能力が問われます。問題文をよく読めば、設問を解くための記述が必ず見つかります。

設問1 C社とD社の経営方針の違いに着目して、システムの相違点を探します。

設問2 〔在庫管理に関する説明〕から、該当する部分を探します。

### 設問1 追加開発が必要となる機能

空欄 a, b: 新たに追加する機能は、D社システムにあって、C社システムにない機能です。両社のシステムの説明は問題文にあります。範囲が広いのでもう少し焦点を絞りましょう。

問題文を読むと、C社は受注生産方式、D社は見込生産方式と、生産方式に大きな違いがあります。したがって、両者の生産管理システムにも大きな違いがありそうです。

そこで、C社の生産管理システムとD社の生産管理システムの機能の説明を、問題文から抜き出して比較してみましょう。

#### ・C社の生産管理システムの機能

- ① 受注情報に基づき、週次で製品ごとの生産計画を作成する。生産計画作成時点で、製造に必要な部品の調達指示を調達管理システムに渡す。また、生産計画に基づき、製品の出荷指示を配送管理システムに渡す。
- ② その他、生産工程の作業管理を行い、発生した作業実績（原価情報）を週次で会計システムに渡す。

#### ・D社の生産管理システムの機能

- ① 販売管理システムの売掛情報と、在庫管理システムの在庫情報を参照し、製品ごとの生産計画を日次で作成する。

- ② 生産計画に基づき、部品の出庫指示を在庫管理システムに、調達指示を調達管理システムに渡す。また、製品の計画時点の在庫情報を在庫管理システムに渡す。
- ③ 製品の完成時の在庫情報を在庫管理システムに渡す。

C社では週次で生産計画を作成するのにに対し、D社では日次で生産計画を作成しています。後者の機能はC社の生産管理システムにはないので、C社の生産管理システムに追加する必要があります。

見込生産方式のほうが、きめ細かい生産計画が必要だからね。



D社の生産管理システムのその他の機能については、いずれも単にデータを渡すだけなので、あえて追加する必要はありません。

以上から、空欄 a には「イ」の「生産管理」、空欄 b には「ア」の「在庫情報と売掛情報から、製品ごとの生産計画を日次で作成する機能」が入ります。

### 設問2 在庫更新の内容

問題文の〔在庫管理に関する説明〕から、在庫更新のタイミングごとの在庫更新の内容を調べます。それぞれの在庫更新の内容については、

- ①在庫区分（有効在庫か実在庫か）
- ②入出庫区分（在庫数を加算する場合は“入庫”，減算または引き当ての場合は“出庫”）
- ③数量（何の数量か）

の3点に注目します

空欄 c：生産計画時の在庫管理については、次のように記述されています。

- (6) 生産計画時に、翌日までに生産予定の製品数を有効在庫として加算する。また、同時に生産に必要な部品を有効在庫から引き当てる。

生産計画時には在庫更新が2回行われます。ただし、「生産に必要な部品を有効在庫から引き当てる」については該当するものは解答群にありません。一方、「生産予定の製品数（＝生産予定数）を有効在庫として加算」については、解答群の **キ** に

有効在庫	入庫	生産予定数
------	----	-------

があります。以上から、空欄 c には **キ** が入ります。

空欄 d：部品納入時の在庫管理については、次のように記述されています。

- (7) 部品が不足し、調達先へ発注したときは、発注分を有効在庫として加算する。部品が調達先から納品されたら実在庫として加算する。

部品納品時には「実在庫として加算する」とあるので、空欄 d は次のようになります。

実在庫	入庫	納品数
-----	----	-----

以上から、空欄 d には **エ** が入ります。

空欄 e：受注時の在庫管理については、次のように記述されています。

- (5) 受注時は、有効在庫から受注数を引き当てる。

「有効在庫から受注数を引き当て」とあるので、空欄 e は次のようになります。

有効在庫	出庫	受注数
------	----	-----

以上から、空欄 e には **力** が入ります。

空欄 f：製品出荷時の在庫管理については、次のように記述されています。

- (10) 製品の出荷時に、製品の実在庫を減算する。

「実在庫を減算」とあるので、空欄 f は次のようになります。

実在庫	出庫	出荷数
-----	----	-----

以上から、空欄 f には **イ** が入ります。

○ 解答 ○

設問 1 a－**イ**，b－**ア**

設問 2 c－**キ**，d－**エ**，e－**力**，  
f－**イ**

問 6

ファンクションポイント法を用いた工数見積りに関する次の記述を読んで、設問 1，2 に答えよ。

A 社では、名簿データ管理システム開発プロジェクトの計画を策定中であり、アプリケーション設計チームで作成した図 1 に示す処理フロー図を基に、工数見積りを行う段階に入った。工数見積りの結果がコスト見積り時の根拠となるので、正確性の確保はもとより、プロジェクトにおける各種の制約条件を考慮したものとする必要がある。

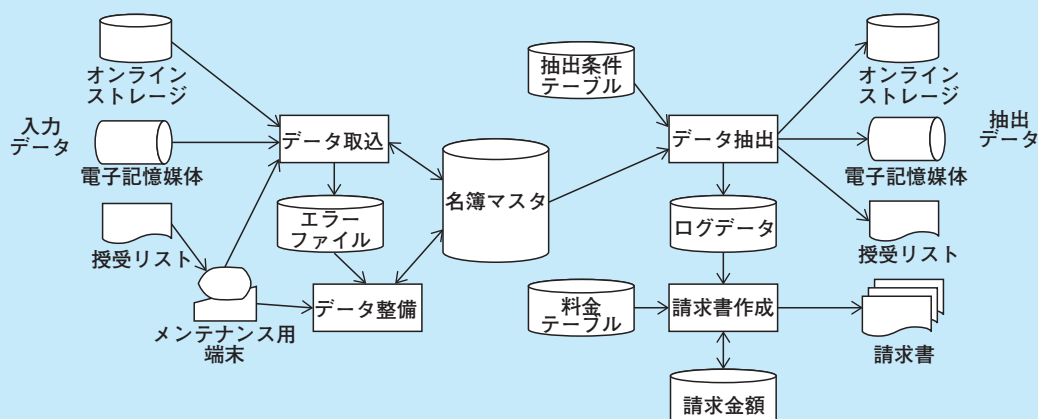


図1 名簿データ管理システムの処理フロー図

#### 〔処理フロー図の説明〕

名簿データ管理システムは、名簿の提供者から受領する名簿データを取り込み、データを整備し、名簿データの利用者からの要求に応じて、データを抽出する。また、利用者から、抽出件数に応じた利用料金を徴収するための請求書を作成する。システムの処理ごとに、その説明を示す。

- (1) データ取込処理では、入力データを基にレコードを編集し、名簿マスタを更新する。入力インタフェースとして、“オンライン接続（オンラインストレージサービスの利用）”、“電子記憶媒体授受”及び“授受リストを基にしたメンテナンス用端末からの入力”の3種類が用意されている。名簿マスタの更新ができなかったデータは、エラーファイルに出力する。
- (2) データ整備処理では、名簿マスタ更新時のエラーファイルを基に、メンテナンス用端末を使用して、レコードを編集し、名簿マスタを更新する。
- (3) データ抽出処理では、名簿マスタ及び抽出条件テーブルを参照し、目的のレコードを抽出して出力する。出力インタフェースとして、“オンライン接続（オンラインストレージサービスの利用）”、“電子記憶媒体授受”及び“授受リスト”の3種類が用意されている。また、利用者ごとのデータ抽出処理の実行と抽出件数に関するログデータを出力する。
- (4) 請求書作成処理では、データ抽出処理で出力されたログデータ及び使用した出力インタフェースとデータ抽出件数ごとに設定された料金テーブルを参照して、請求金額を算出する。算出された請求金額を、請求レコード用フォーマットに編集して、請求書を出力する。

#### 〔工数見積方法の説明〕

- (1) 工数見積りには、A社でアレンジした簡易ファンクションポイント法を用いる。簡易ファンクションポイント法は、ソフトウェアがもつ機能及び入出力インタフェースの数を基に、そのソフトウェアの開発規模を推定する手法である。図1で示す各処理の内容を、機能及び入出力インタフェースごとに分類し、それぞれをファンクションとして定義する。図1の処理フロー図から見積対象の処理ごとに、各ファンクションの個数を求める。
- (2) 求められた各ファンクションの個数を基に、処理ごとの開発規模を求める。名簿データ管理システムのファンクションとそれに対応する開発規模を表1に示す。



表1 ファンクション一覧

項番	ファンクション名	ファンクションに対応する 開発規模 (k ステップ)
1	ファイル読込	0.1
2	ファイル書込	0.3
3	帳票作成	0.6
4	端末画面	1.0
5	レコード編集	0.5

注記 開発規模の単位 1k ステップ= 1,000 ステップ

- (3) 見積対象処理ごとに算出された開発規模と表2に示す開発生産性の計画値から、工数を算出する。

表2 開発生産性の計画値

開発生産性計画値	0.5 (k ステップ/人月)
----------	-----------------

**設問 1** プロジェクト管理グループに属する社員Bは、プロジェクトマネージャの指導の下、工数見積方法に従い、図1の処理フロー図に示された各処理の開発に要する工数を算出して、表3にまとめた。表3中の  に入れる正しい答えを、解答群の中から選べ。

表3 処理ごとのファンクション数と開発規模一覧

ファンクション名 処理名	ファイル 読込	ファイル 書込	帳票作成	端末画面	レコード 編集	処理別開発 規模合計	処理別工数 合計 (人月)
データ取込	3	2	0	1	1		4.8
	0.3	0.6	0.0	1.0	0.5	2.4	
データ整備	2	1	0	1	1		<input type="text" value="b"/>
	0.2	0.3	0.0	1.0	0.5	2.0	
データ抽出	2	<input type="text" value="a"/>	1	0	0		
	0.2		0.6	0.0	0.0		
請求書作成	3	1	1	0	1		3.4
	0.3	0.3	0.6	0.0	0.5	1.7	
ファンクション別開発規模 合計	1.0		1.2	2.0	1.5		

注記 表中の処理ごとに、上段がファンクション数、下段がファンクション数を基に算出された開発規模(単位:kステップ)を表す。網掛けの部分は表示していない。

aに関する解答群

☐ ア 1      ☐ イ 2      ☐ ウ 3      ☐ エ 4

bに関する解答群

☐ ア 1.0      ☐ イ 2.0      ☐ ウ 2.5      ☐ エ 4.0

**設問 2** 設問 1 で求めた工数を基に算出した開発コストは、プロジェクトで想定している開発コストの計画値を上回った。社員 B はプロジェクトマネージャから、開発工数を削減する方法の検討を行うよう指示を受けた。

社員 B は、処理別工程ごとの工数を表 4 にまとめた。表 3 及び表 4 を検証して、工数削減方法の検討を進めた。検討を行った作業に関する次の記述中の  に入れる適切な答えを、解答群の中から選べ。ここで、工程ごとの工数は、処理ごとの工数に工程比率を乗じて算出したものである。

表 4 処理別工程ごとの工数一覧

工程名	外部設計	詳細設計	製造	総合試験	運用試験	処理別工数 合計 (人月)
工程比率 処理名	15%	25%	40%	15%	5%	
データ取込	0.72	1.20	1.92	0.72	0.24	4.8
データ整備	0.60		1.60	0.60	0.20	
データ抽出	0.51	0.85	1.36		0.17	
請求書作成	0.51	0.85	1.36	0.51	0.17	3.4

注記 網掛け部分は表示していない。

全ての処理に利用され、ファンクション別開発規模合計が最も大きいファンクションに対して対策を講じることが、工数削減の効果が最も大きいと想定し、表 3 から対象ファンクションを  c と判断した。

工数削減施策として、“対象ファンクションを共通部品として作成し、各処理に適用することによって、工数を削減する施策”を検討した。共通部品化によって、全工程の工数合計の 80% に相当する  d 工程における工数の一部を削減することができる。

例えば、データ抽出処理では、 d 工程の工数のうち  e 人月が  c の工数なので、本施策によって工数削減を見込むことができる。

#### c に関する解答群

- ☐ ア 端末画面                      ☐ イ 帳票作成                      ☐ ウ ファイル書込  
☐ エ ファイル読込                      ☐ オ レコード編集

#### d に関する解答群

- ☐ ア 外部設計  
☐ イ 外部設計，詳細設計  
☐ ウ 外部設計，詳細設計，製造  
☐ エ 外部設計，詳細設計，製造，総合試験  
☐ オ 外部設計，詳細設計，製造，総合試験，運用試験

#### e に関する解答群

- ☐ ア 1.088                      ☐ イ 1.44                      ☐ ウ 2.176                      ☐ エ 2.72

## 問 6 午後のカギ

工数見積の計算問題です。前の空欄で求めた値を後の空欄でも使うので、計算ミスに注意しましょう。

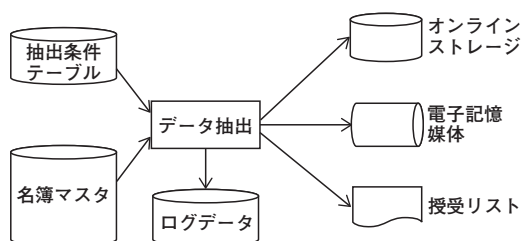
設問 1 図 1 の入っていく矢印が入力、出ていく矢印が出力です。ファイルへの出力と帳票への出力を一緒にしないようにしましょう。

設問 2 表 3 の網掛け部分に入る数値を計算します。

### 設問 1 工数の見積

空欄 a：データ抽出処理のファンクション数を求めます。

図 1 の処理フロー図から、「データ抽出」に関する部分を抜き出してみましょう。



上図を見ると、「データ抽出」に入っている矢印が 2 本、「データ抽出」から出ていく矢印が 4 本あります。このうち、入っている矢印はファイルからデータが入力されることを示すので、**ファイル読込**のファンクション数は 2 になります。

また、出ていく矢印は、データ抽出処理によってデータが出力されることを示します。出力先のうち、「オンラインストレージ」「電子記憶媒体」「ログデータ」の 3 つはファイル、「授受リスト」は帳票ですから、**ファイル書込**のファンクション数は 3 になります。

以上から、空欄 a は **ウ** です。

空欄 b：「データ整備」処理の工数を求めます。〔工数見積方法の説明〕の (3) に、

見積対象処理ごとに算出された開発規模と表 2 に示す開發生産性の計画値から、工数を算出する。

とあります。開發生産性の計画値は、表 2 より 0.5k ステップ／人月です。

工数の算出式は問題文に記述されていませんが、たとえばデータ取込処理の場合、開発規模 2.4k ステップに対し、工数が 4.8 人月となっているので、 $2.4 \div$

$0.5 = 4.8$  のように計算すればよいとわかります。算出式は次のようになります。

$$\text{工数 (人月)} = \frac{\text{開発規模 (kステップ)}}{\text{開發生産性計画値 (kステップ／人月)}}$$

データ整備処理の開発規模合計は 2.0k ステップなので、処理別工数は

$$2.0 \div 0.5 = 4.0 \text{ 人月}$$

となります。空欄 b は **エ** です。

### 設問 2 開発工数の削減

空欄 c：ファンクション別開発規模合計の最も大きいファンクションを選びます。

ファンクション別開発規模合計は表 3 に一覧がありますが、「ファイル書込」の開発規模合計が網掛けになっているので、この値を算出する必要があります。

まず、「データ抽出」処理におけるファイル書込の開発規模を計算します。表 1 より、ファイル書込の開発規模は、**ファンクション数 × 0.3k ステップ**です。「データ抽出」処理におけるファイル書込のファンクション数は 3 (空欄 a) なので、開発規模は  $3 \times 0.3 = 0.9k$  ステップです。

処理名	ファイル書込
データ取込	0.6
データ整備	0.3
データ抽出	0.9
請求書作成	0.3
合計	2.1

以上から、ファイル書込のファンクション別開発規模合計は、 $0.6 + 0.3 + 0.9 + 0.3 = 2.1k$  ステップになります。この値は表 3 のファンクション別開発規模合計の中で最も大きいので、空欄 c には **ウ** の「ファ

イル書込」が入ります。

空欄 d：各工程の比率は表 4 に記載されているので、合計が 80%になるものを解答群から選びます。

- × ア 外部設計：15%
- × イ 外部設計，詳細設計：15 + 25 = 40%
- ウ 外部設計，詳細設計，製造：15 + 25 + 40 = 80%
- × エ 外部設計，詳細設計，製造，総合試験：15 + 25 + 40 + 15 = 95%
- × オ 外部設計，詳細設計，製造，総合試験，運用試験：100%

以上から、空欄 d は **ウ** です。

空欄 e：「データ抽出」処理のファイル書込ファンク

ションの開発規模は、空欄 c で求めた 0.9k ステップです。これを工数に直すと、 $0.9 / 0.5 = 1.8$  人月に相当します。

外部設計，詳細設計，製造工程の工数はこのうちの 80%なので、

$$1.8 \times 0.8 = 1.44 \text{ 人月}$$

となります。空欄 e は **イ** です。

○ 解答 ○

設問 1 a - **ウ**，b - **エ**

設問 2 c - **ウ**，d - **ウ**，e - **イ**

問 7

システム移行の作業計画に関する次の記述を読んで、設問 1～3 に答えよ。

A 社の物流部では、物流費管理システムのバージョンアップを予定している。物流費管理システムは、ソフトウェアパッケージ（以下、パッケージソフトという）と A 社用に開発されたアプリケーションソフトウェア（以下、開発ソフトという）から構成されている。このたび、パッケージソフト及び開発ソフトのバージョンアップ対応版が準備できたので、物流費管理システムのバージョンアップに関する作業計画を策定することになった。表 1 は、バージョンアップに必要な作業の一覧である。

表 1 バージョンアップに必要な作業の一覧

作業	作業内容	作業時間 (時間)	必要要員数 (人)	先行作業
No1	物流費管理システムのサービス停止	1	1	—
No2	物流費管理システムのバックアップ	1	4	No1
No3	パッケージソフトデータの抽出	1	3	No2
No4	開発ソフトデータの抽出	1	3	No2
No5	パッケージソフトデータの変換	1	3	No3
No6	開発ソフトデータの変換	1	3	No4
No7	パッケージソフトの新バージョンの導入	2	4	No3
No8	開発ソフトの新バージョンの導入	1	2	No4
No9	パッケージソフトの新バージョンと開発ソフトの新バージョンの単体動作及び連結動作の確認	3	6	No7, No8
No10	変換後のパッケージソフトデータのロード	2	2	No5, No9
No11	変換後の開発ソフトデータのロード	1	2	No6, No9
No12	変換後のパッケージソフトデータ及び開発ソフトデータの確認	1	3	No10, No11
No13	物流費管理システム全体の動作確認及び判定	3	6	No9, No12

**設問 1** バージョンアップのスケジュールに関する次の記述中の  に入れる正しい答えを、解答群の中から選べ。

物流部では、表 1 を基にアローダイアグラムを作成することにした。図 1 は、作成途中のアローダイアグラムである。

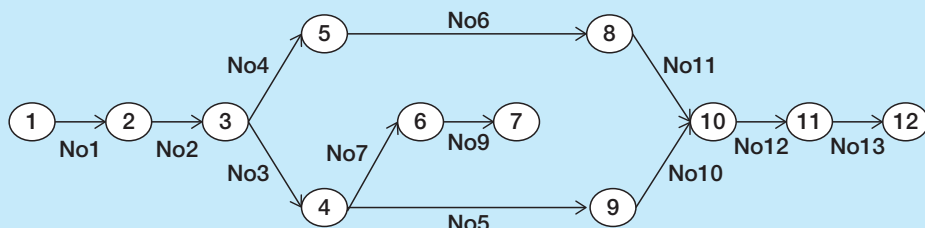


図 1 作成途中のアローダイアグラム

図 1 の作業途中のアローダイアグラムに不足しているのは、ノード  a への作業 No8 を表す矢線と、ノード 7 から  b へのダミー作業を表す矢線である。

バージョンアップに必要な一連の作業の最短所要時間を計算すると、 c 時間となる。

a に関する解答群

- ☐ ア 5 から 6     
 ☐ イ 5 から 7     
 ☐ ウ 6 から 8     
 ☐ エ 6 から 9

b に関する解答群

- ☐ ア 8                     
 ☐ イ 9                     
 ☐ ウ 10  
☐ エ 8 と 9             
 ☐ オ 8 と 10             
 ☐ カ 9 と 10

c に関する解答群

- ☐ ア 9                     
 ☐ イ 10                     
 ☐ ウ 13                     
 ☐ エ 14                     
 ☐ オ 19

**設問 2** 物流部では、バージョンアップ作業に必要なとなる要員計画を策定することにした。要員計画に関する次の記述及び表中の  に入れる適切な答えを、解答群の中から選べ。

物流部では、6 時に作業 No1 を開始し、その後の作業は先行作業が全て完了した時点ですぐに開始できるように作業を配置した。表 2 に、作成した要員計画表の一部を示す。表 2 の“作業”の欄は、該当該時刻に開始する作業を表している。同じ開始時刻に複数の作業が表記されている場合は、それらの作業を並行して行うことを意味している。“必要人数”の欄は、その時間帯に必要な要員数であり、同じ時間帯に複数の作業を行う場合は、それぞれの作業の要員数の合計である。

表2 要員計画表の一部

開始時刻	作業				必要人数
6 時	No1				1
7 時	No2				4
8 時	No3		No4		6
9 時	No5	No7 <sup>1)</sup>	No6	No8	12
10 時					4
11 時					6
12 時					6
13 時					d
14 時					
15 時					
16 時					3

注記 網掛けの部分は表示していない。

注<sup>1)</sup> No7のように作業が1時間よりも長く掛かる場合は、途中の罫線<sup>けい</sup>を省略している。

9～10時の必要人数は12人であるが、eに開始するなど作業の時間帯を工夫することによって、全体の作業完了時刻を遅らせることなく、その時間帯の必要人数を減らすことができる。

物流部では、全体の作業完了時刻を遅らせることなく、かつ、時間帯ごとの必要人数を最少とするように、作業No5、No6及びNo11の開始時刻を見直した。その結果、一連の作業を行うのに必要な最少人数はf人となった。ここで、要員は全ての作業が行えるものとする。

#### dに関する解答群

<span style="background-color: #0070c0; color: white; padding: 2px;">ア</span>	<div style="border: 1px solid black; padding: 5px; text-align: center;">6 2 2</div>	<span style="background-color: #0070c0; color: white; padding: 2px;">イ</span>	<div style="border: 1px solid black; padding: 5px; text-align: center;">6 2 4</div>	<span style="background-color: #0070c0; color: white; padding: 2px;">ウ</span>	<div style="border: 1px solid black; padding: 5px; text-align: center;">6 4 2</div>	<span style="background-color: #0070c0; color: white; padding: 2px;">エ</span>	<div style="border: 1px solid black; padding: 5px; text-align: center;">8 2 2</div>	<span style="background-color: #0070c0; color: white; padding: 2px;">オ</span>	<div style="border: 1px solid black; padding: 5px; text-align: center;">8 4 2</div>
---	---	---	---	---	---	---	---	---	---

#### eに関する解答群

<span style="background-color: #0070c0; color: white; padding: 2px;">ア</span> No5を10時	<span style="background-color: #0070c0; color: white; padding: 2px;">イ</span> No5を14時	<span style="background-color: #0070c0; color: white; padding: 2px;">ウ</span> No6を7時	<span style="background-color: #0070c0; color: white; padding: 2px;">エ</span> No6を8時
---	---	--	--

#### fに関する解答群

<span style="background-color: #0070c0; color: white; padding: 2px;">ア</span> 4	<span style="background-color: #0070c0; color: white; padding: 2px;">イ</span> 6	<span style="background-color: #0070c0; color: white; padding: 2px;">ウ</span> 7	<span style="background-color: #0070c0; color: white; padding: 2px;">エ</span> 8	<span style="background-color: #0070c0; color: white; padding: 2px;">オ</span> 9	<span style="background-color: #0070c0; color: white; padding: 2px;">カ</span> 10
---	---	---	---	---	--

**設問3** 物流部では、全体の作業完了時刻を早めたいと考えている。全体の作業完了時刻を30分早められる記述として正しい答えを、解答群の中から二つ選べ。

#### 解答群

- ア No2とNo9の作業時間を15分ずつ短縮する。
- イ No3とNo4の作業時間を15分ずつ短縮する。
- ウ No5とNo6の作業時間を15分ずつ短縮する。
- エ No10とNo11の作業時間を15分ずつ短縮する。
- オ No12とNo13の作業時間を15分ずつ短縮する。

## 問 7 午後のカギ

作業計画や要員計画に関する問題です。難易度はやや高めですが、アローダイアグラムについては午前試験でも毎回必ず出題されているので、よく理解しておく必要があります。

**設問 1** 表 1 を参照して、アローダイアグラムを完成します。各作業は、先行作業が完了したノードから開始し、自分が先行作業となっている作業の開始ノードで完了します。

**設問 2** 表 2 の網掛け部分を埋めて、必要人数を計算します。作業の時間帯は、先行作業より前や、自分が先行作業になっている作業より後にはずさないことに注意しましょう。

**設問 3** 全体の所要時間を短縮するには、クリティカルパス上にある作業を短縮します。

### 設問 1 アローダイアグラムの完成

**空欄 a:** アローダイアグラムは、プロジェクトを構成する各作業を矢線で表し、作業の結合点を○で表した図です。

図 1 のアローダイアグラムに、作業 No8 の矢線を追加します。

表 1 によると、作業 No8 の先行作業は No4 です。また、作業 No8 は No9 の先行作業になっています。したがって、作業 No8 は No4 が終了しなければ開始できず、作業 No9 は No8 が終了しなければ開始できません。

以上から、No8 の矢線は、No4 の終了したノード 5 から、No9 を開始するノード 6 に向かう方向になります。空欄 a には **ア** の「5 から 6」が入ります。

**空欄 b:** ノード 7 から出るダミー作業を加えます。ダミー作業とは、所要時間 0 の作業を表します。

ノード 7 に到達する作業は No9 ですが、この作業は、作業 No10、No11、No13 の先行作業になっています。したがって、作業 No10、No11、No13 は、いずれも No9 が終了しないと開始できません。作業 No10 はノード 9、作業 No11 はノード 8 から開始するので、ノード 7 からノード 8 と 9 へのダミー作業を加えます。

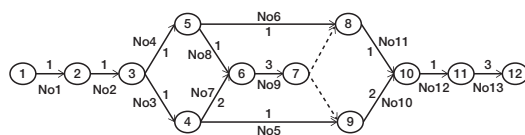
No13 はどうするの？



これらのダミー作業の追加によって、作業 No13 の開始するノード 11 の時点で、作業 No9 は明らかに終了しています。したがって、ノード 7 からノード 11 へのダミー作業を追加する必要はありません。

以上から、空欄 b には **エ** の「8 と 9」が入ります。

**空欄 c:** 完成したアローダイアグラムは次のようになります。なお、表 1 をもとに、各作業の作業時間も図に書き入れました。



この図をもとに、次の作業を開始できるようになる最も早い時刻（最早開始時刻）を、各ノードについて求めます。

ノード 1 : 0 時間

ノード 2 :  $0 + 1 = 1$  時間後

ノード 3 :  $1 + 1 = 2$  時間後

ノード 4 :  $2 + 1 = 3$  時間後

ノード 5 :  $2 + 1 = 3$  時間後

ノード 6 : No8 の終了が  $3 + 1 = 4$  時間後、No7 の終了が  $3 + 2 = 5$  時間後なので、最早開始時刻は 5 時間後になります。

ノード 7 :  $5 + 3 = 8$  時間後

ノード 8 : No6 の終了が  $3 + 1 = 4$  時間後、No9 の終了が 8 時間後なので、最早開始時刻は 8 時間後になります。

ノード 9 : No5 の終了が  $3 + 1 = 4$  時間後、No9 の終了が 8 時間後なので、最早開始時刻は 8 時間後になります。

ノード 10 : No10 の終了が  $8 + 2 = 10$  時間後、No11 の終了が  $8 + 1 = 9$  時間後なので、最早開始時刻は 10 時間後になります。

ノード 11 :  $10 + 1 = 11$  時間後

ノード 12 :  $11 + 3 = 14$  時間後



以上から、一連の作業の所要時間は **14 時間** となります。空欄 c は **エ** です。

## 設問 2 要員計画の策定

空欄 d：表 2 の網掛けになっている部分に作業 No を記入して、表を完成させましょう。

11 時までの時点で作業 No8 が終了しているの、11 時から作業 No9 を開始します。表 1 から、作業 No9 の作業時間は 3 時間です。また、No10 以降の作業は作業 No9 が終了しないと開始できないので、この時間帯にほかの作業は行えません。

作業 No10 と No11 は、作業 No9 が終了する 14 時から開始できます。No10 の作業時間は 2 時間、No11 の作業時間は 1 時間です。

作業 No12 は No10 と No11 の終了後、16 時から開始します。

開始時刻	作業				必要人数
6 時	No1				1
7 時	No2				4
8 時	No3		No4		6
9 時	No5	No7	No6	No8	12
10 時					4
11 時		No9			6
12 時					6
13 時					d
14 時	No10		No11		
15 時					
16 時	No12				3
17 時	No13				6
18 時					6
19 時					6

作業 No を記入したら、表 1 の各作業の必要要員数を参照して、13～16 時までの必要人数を考えます。

- ・13～14 時：作業 No9 だけなので、必要人数は 6 人です。
- ・14～15 時：作業 No10 と No11 を行うので、必要人数は No10 が 2 人、No11 が 2 人の合計 4 人です。
- ・15～16 時：作業 No10 だけなので、必要人数は 2 人です。

以上から、空欄 d には上から 6, 4, 2 が入ります。正解は **ウ** です。

空欄 e：9～10 時の時間帯には、No5～8 が作業に入っています。このうち、No5, No6, No8 は別の時間帯にずらすことができます。ずらすことのできる範

囲は、それぞれ次のとおりです。

No5：No3 の終了後～No10 の開始前

No6：No4 の終了後～No11 の開始前

No8：No4 の終了後～No9 の開始前

開始時刻	作業				必要人数
6 時	No1				1
7 時	No2				4
8 時	No3		No4		6
9 時		No7		No8	12
10 時					4
11 時	No5	No9	No6		6
12 時					6
13 時					6
14 時	No10		No11		4
15 時					2
16 時	No12				3
17 時	No13				6
18 時					6
19 時					6

e に関する解答群のうち、上記の条件に合う選択肢は、**ア** の「No5 を 10 時」だけです。正解は **ア** です。

空欄 f：時間帯の最大必要人数が最も少なくなるように、作業 No5, No6, No11 の開始時刻を見直します。

まず、作業 No5 の開始時刻を 10 時に移動しましょう。すると、9～10 時の必要人数は 9 人、10～11 時の必要人数は 7 人になります（表 1 の「必要要員数」参照）。

次に、作業 No11 の開始時刻を 15 時に移動します。すると、14～15 時の必要人数は 2 人、15～16 時の必要人数は 4 人に変わります。

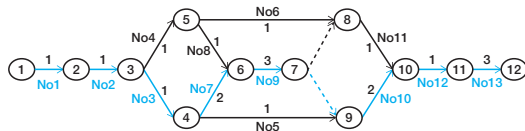
最後に、作業 No6 の開始時刻を 14 時に移動します。すると、9～10 時の必要人数は 6 人、14～15 時の必要人数は 5 人になります。

開始時刻	作業				必要人数
6 時	No1				1
7 時	No2				4
8 時	No3		No4		6
9 時		No7		No8	6
10 時	No5				7
11 時		No9			6
12 時					6
13 時					6
14 時	No10		No6		5
15 時			No11		4
16 時	No12				3
17 時	No13				6
18 時					6
19 時					6

この結果、必要な最少人数は 7 人となります。正解は **ウ** です。

### 設問 3 作業時間の短縮

完成したアローダイアグラムを、ノード 1 からノード 12 までたどる経路のうち、経路上の作業の作業時間の合計が、全体の所要時間（14 時間）と等しい経路を探します。このような経路を、**クリティカルパス**といいます。



上図の色で示した経路がクリティカルパスとなります。

全体の所要時間を短縮するには、クリティカルパス上にある作業の所要時間を短縮します。

そこで、解答群の中から、クリティカルパス上にあ

る作業の所要時間を短縮しているものを選びます。

- **ア** No2 と No9 は、どちらもクリティカルパス上にある作業です。
- × **イ** No4 はクリティカルパス上の作業ではありません。
- × **ウ** No5 と No6 は、どちらもクリティカルパス上にある作業ではありません。
- × **エ** No11 はクリティカルパス上の作業ではありません。
- **オ** No12 と No13 は、どちらもクリティカルパス上にある作業です。

以上から、**ア** と **オ** の 2 つが正解です。

### 解答

設問 1 a - **ア**, b - **エ**, c - **エ**

設問 2 d - **ウ**, e - **ア**, f - **ウ**

設問 3 **ア**, **オ**

次の問 8 は必須問題です。必ず解答してください。

### 問 8 次のアルゴリズムの説明を読んで、設問 1 ～ 3 に答えよ。

セルを 1 列に連続して並べた領域がある。この領域中のセルについて、割当てと解放の処理を行う。

各セルには、セル位置を指定するための連続する整数が対応している。領域のセル数や、対応する整数の範囲には、特に制限がない。

各セルは、“空き” 又は “割当済み” のいずれかの状態にある。現在、領域中のどのセルが “空き” の状態にあるかという情報を、空きリストとして保持している。

関数 Alloc ( 始点, 終点 ) は、引数で指定した始点から終点までの連続した “空き” セルを “割当済み” として、空きリストから取り除く。関数 Free ( 始点, 終点 ) は、引数で指定した始点から終点までの連続した “割当済み” セルを “空き” として、空きリストに戻す。

#### 〔空きリストの説明〕

空きリストの形式を、次に示す。

{ { 始点<sub>1</sub>, 終点<sub>1</sub> }, { 始点<sub>2</sub>, 終点<sub>2</sub> }, ..., { 始点<sub>N</sub>, 終点<sub>N</sub> } }

{ 始点<sub>i</sub>, 終点<sub>i</sub> } ( 始点<sub>i</sub> ≤ 終点<sub>i</sub> ) は、一つの連続した “空き” セルの先頭位置と終端位置の組 ( 以下、組という ) で、始点<sub>1</sub> < 始点<sub>2</sub> < ... < 始点<sub>N</sub> である。

割当て・解放の処理と空きリストの状態の例を、次の (1) ～ (3) に示す。ここで、セル    中の数字は、セル位置を表す。また、   は “空き” を、   は “割当済み” を、それぞれ表す。

- (1) 領域の初期状態は、全セルが空いている。空きリストは  $\{-\infty, +\infty\}$  で表す。

...	0	1	2	3	4	5	6	7	8	9	...
-----	---	---	---	---	---	---	---	---	---	---	-----

初期状態の空きリスト： $\{-\infty, +\infty\}$

ここで、記号“ $-\infty$ ”は、領域中のどのセル位置の値よりも小さい整数を表し、記号“ $+\infty$ ”は、領域中のどのセル位置の値よりも大きい整数を表すものとする。また、セル位置  $-\infty \sim +\infty$  のうち、領域外の部分には“空き”セルが並んでいるものとする。

- (2) 関数 Alloc で“割当済み”としたセルは、空きリストから取り除く。例えば、(1) の初期状態から、Alloc(1, 2) と Alloc(6, 8) を実行すると、次のようになる。

...	0	1	2	3	4	5	6	7	8	9	...
-----	---	---	---	---	---	---	---	---	---	---	-----

実行後の空きリスト： $\{-\infty, 0\}, \{3, 5\}, \{9, +\infty\}$

実行後、空きリスト中の組の個数は3となる。

- (3) 関数 Free で解放したセルは、空きリストに戻す。例えば、(2) の実行後の状態から、Free(6, 7) を実行すると、次のようになる。

...	0	1	2	3	4	5	6	7	8	9	...
-----	---	---	---	---	---	---	---	---	---	---	-----

実行後の空きリスト： $\{-\infty, 0\}, \{3, 7\}, \{9, +\infty\}$

実行後、解放された“空き”セルの組  $\{6, 7\}$  は、実行前の“空き”セルの組  $\{3, 5\}$  とつながって一つの連続した“空き”セルの組  $\{3, 7\}$  となるので、空きリスト中の組の個数は3となる。

#### 〔関数 Alloc の説明〕

関数 Alloc( 始点<sub>p</sub>, 終点<sub>p</sub>) の処理手順は、次のとおりである。

なお、引数の値は、 $-\infty < \text{始点}_p \leq \text{終点}_p < +\infty$  を満たしているものとする。

- (1) 空きリスト中に、始点<sub>i</sub>  $\leq$  始点<sub>p</sub> かつ 終点<sub>p</sub>  $\leq$  終点<sub>i</sub> を満たす組 { 始点<sub>i</sub>, 終点<sub>i</sub> } が存在すれば (2) へ進む。存在しなければ、“一部又は全体が割当済み”を表示して、処理を終了する。
- (2) 割当てが可能であるので、表 1 に従って、引数の状況に対応した空きリストの更新処理を実行して、処理を終了する。

表 1 関数 Alloc の空きリスト更新処理

引数の状況	空きリストの更新処理
始点 <sub>i</sub> = 始点 <sub>p</sub> かつ 終点 <sub>p</sub> = 終点 <sub>i</sub>	組 { 始点 <sub>i</sub> , 終点 <sub>i</sub> } を取り除く。
始点 <sub>i</sub> = 始点 <sub>p</sub> かつ 終点 <sub>p</sub> < 終点 <sub>i</sub>	組 { 始点 <sub>i</sub> , 終点 <sub>i</sub> } を組 <span style="background-color: #cccccc; display: inline-block; width: 100px; height: 1em;"></span> で置き換える。
始点 <sub>i</sub> < 始点 <sub>p</sub> かつ 終点 <sub>p</sub> = 終点 <sub>i</sub>	組 { 始点 <sub>i</sub> , 終点 <sub>i</sub> } を組 <span style="background-color: #cccccc; display: inline-block; width: 100px; height: 1em;"></span> で置き換える。
始点 <sub>i</sub> < 始点 <sub>p</sub> かつ 終点 <sub>p</sub> < 終点 <sub>i</sub>	組 { 始点 <sub>i</sub> , 終点 <sub>i</sub> } を二つの組 { 始点 <sub>i</sub> , 始点 <sub>p</sub> - 1 } と { 終点 <sub>p</sub> + 1, 終点 <sub>i</sub> } で置き換える。

注記 網掛けの部分は表示していない。

〔関数 Free の説明〕

関数 Free( 始点<sub>p</sub>, 終点<sub>p</sub>) の処理手順は, 次のとおりである。

なお, 引数の値は,  $-\infty < \text{始点}_p \leq \text{終点}_p < +\infty$  を満たしているものとする。

- (1) 空きリスト中に, 終点<sub>i</sub> < 始点<sub>p</sub> かつ 終点<sub>p</sub> < 始点<sub>i+1</sub> を満たす連続する二つの組 { 始点<sub>i</sub>, 終点<sub>i</sub> } と { 始点<sub>i+1</sub>, 終点<sub>i+1</sub> } が存在すれば (2) へ進む。存在しなければ, “一部又は全体が割当済みでない” を表示して, 処理を終了する。
- (2) 解放が可能であるので, 表 2 に従って, 引数の状況に対応した空きリストの更新処理を実行して, 処理を終了する。

表 2 関数 Free の空きリスト更新処理

引数の状況	空きリストの更新処理
終点 <sub>i</sub> = 始点 <sub>p</sub> - 1 かつ 終点 <sub>p</sub> + 1 = 始点 <sub>i+1</sub>	二つの組 { 始点 <sub>i</sub> , 終点 <sub>i</sub> } と { 始点 <sub>i+1</sub> , 終点 <sub>i+1</sub> } を一つの組 <input type="text" value="a"/> で置き換える。
終点 <sub>i</sub> = 始点 <sub>p</sub> - 1 かつ 終点 <sub>p</sub> + 1 < 始点 <sub>i+1</sub>	組 { 始点 <sub>i</sub> , 終点 <sub>i</sub> } を組 { 始点 <sub>i</sub> , 終点 <sub>p</sub> } で置き換える。
終点 <sub>i</sub> < 始点 <sub>p</sub> - 1 かつ 終点 <sub>p</sub> + 1 = 始点 <sub>i+1</sub>	組 { 始点 <sub>i+1</sub> , 終点 <sub>i+1</sub> } を組 { 始点 <sub>p</sub> , 終点 <sub>i+1</sub> } で置き換える。
終点 <sub>i</sub> < 始点 <sub>p</sub> - 1 かつ 終点 <sub>p</sub> + 1 < 始点 <sub>i+1</sub>	組 { 始点 <sub>i</sub> , 終点 <sub>i</sub> } の直後に組 <input type="text" value="b"/> を挿入する。

**設問 1** 本文中の  に入れる正しい答えを, 解答群の中から選べ。

a, b に関する解答群

☐ ア { 始点<sub>i</sub>, 終点<sub>i+1</sub> }

☐ イ { 始点<sub>i</sub>, 終点<sub>p</sub> }

☐ ウ { 始点<sub>p</sub>, 終点<sub>i+1</sub> }

☐ エ { 始点<sub>p</sub>, 終点<sub>p</sub> }

**設問 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 Alloc の説明に基づいて, プログラムを作成した。

空きリスト中の現在の組の個数は大域整数型変数 N に格納されている。空きリスト { { 始点<sub>1</sub>, 終点<sub>1</sub> }, { 始点<sub>2</sub>, 終点<sub>2</sub> }, ..., { 始点<sub>N</sub>, 終点<sub>N</sub> } } については, 始点<sub>i</sub> (i : 1, 2, ..., N) の値は大域整数型配列始点の要素始点[i] に, 終点<sub>i</sub> (i : 1, 2, ..., N) の値は大域整数型配列終点の要素終点[i] に, それぞれ格納されている。これらの配列は, 十分に大きいものとする。

〔プログラム〕

○関数: Alloc(整数型: 始点P, 整数型: 終点P)

○整数型: I, L

・ I ← 1

■ 終点P > 終点[I]

・ I ← I + 1

■

```

↑ 始点[I] ≤ 始点P
↑ (始点[I] = 始点P) and (終点P = 終点[I])
■ L: I + 1, L ≤ N, 1
  ・ 始点[L - 1] ← 始点[L]
  ・ 終点[L - 1] ← 終点[L]
■
  ・ N ← N - 1
↓
↑ (始点[I] = 始点P) and (終点P < 終点[I])
  ・ c
↓
↑ (始点[I] < 始点P) and (終点P = 終点[I])
  ・ d
↓
↑ (始点[I] < 始点P) and (終点P < 終点[I])
■ L: e
  ・ 始点[L + 1] ← 始点[L]
  ・ 終点[L + 1] ← 終点[L]
■
  ・ 始点[I + 1] ← 終点P + 1
  ・ 終点[I + 1] ← 終点[I]
  ・ 終点[I] ← 始点P - 1
  ・ N ← N + 1
↓
・ print("一部又は全体が割当済み") /* " " 内の文字列を表示 */
↓

```

c, d に関する解答群

- |   |   |
|---|---|
| <span style="background-color: #00a0e3; color: white; padding: 2px;">ア</span> 始点[I] ← 始点P - 1 | <span style="background-color: #00a0e3; color: white; padding: 2px;">イ</span> 始点[I] ← 終点P + 1 |
| <span style="background-color: #00a0e3; color: white; padding: 2px;">ウ</span> 終点[I] ← 始点P - 1 | <span style="background-color: #00a0e3; color: white; padding: 2px;">エ</span> 終点[I] ← 終点P + 1 |

e に関する解答群

- |  |  |
|--|--|
| <span style="background-color: #00a0e3; color: white; padding: 2px;">ア</span> I + 1, L < N, 1  | <span style="background-color: #00a0e3; color: white; padding: 2px;">イ</span> I + 1, L ≤ N, 1  |
| <span style="background-color: #00a0e3; color: white; padding: 2px;">ウ</span> N, L ≥ I + 1, -1 | <span style="background-color: #00a0e3; color: white; padding: 2px;">エ</span> N, L > I + 1, -1 |

**設問 3** 次の記述中の   に入れる適切な答えを、解答群の中から選べ。

このアルゴリズムでは、空きリスト  $\{\{\text{始点}_1, \text{終点}_1\}, \{\text{始点}_2, \text{終点}_2\}, \dots, \{\text{始点}_N, \text{終点}_N\}\}$  の始点<sub>1</sub>に値  $-\infty$  を、終点<sub>N</sub>に値  $+\infty$  をそれぞれ設定している。このような設定をすることの利点の一つに、f という特徴が挙げられる。

また、このアルゴリズムでは、空きリスト中の組の個数が変化する。領域中のセル数がE個であるとする。このとき、空きリスト中の組の個数は、最大で g となる。また、E個の全てのセルが“割当済み”となったとき、空きリスト中の組の個数は、h となる。ここで、整数同士の除算では、商の小数点以下を切り捨てる。

## fに関する解答群

- ア** 空きリストが空（組の個数が0）にならない
- イ** 関数 Free の実行時に空きリスト中の組の個数が2以上であることが保証される
- ウ** 始点<sub>1</sub>又は終点<sub>N</sub>の値が変わらない限り領域中に“空き”セルが残っている
- エ** 領域中の一つの連続した“空き”セルが幾ら長くても一つの組で表せる

## g, hに関する解答群

- |                               |                  |                         |
|-------------------------------|------------------|-------------------------|
| <b>ア</b> 1                    | <b>イ</b> 2       | <b>ウ</b> $E \div 2 + 1$ |
| <b>エ</b> $(E + 1) \div 2 + 1$ | <b>オ</b> $E + 1$ | <b>カ</b> $E + 2$        |

## 問8 午後のカギ

アルゴリズムの問題では、プログラムの各部分が、問題文にある処理の説明のどの部分に当たるかを考えます。また、文章だけではわかりにくい処理は、問題用紙の余白に図を描いて考えるとわかりやすくなります。

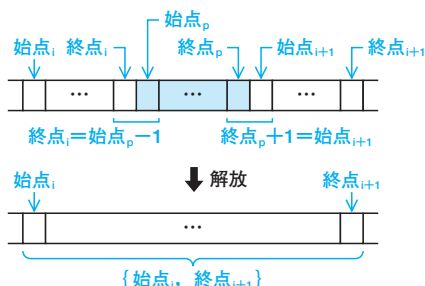
設問1 簡単でいいので、図を描いて考えましょう。

設問2 プログラムと表1の内容を対応付けて考えます。なお、配列をずらす場合は、配列の先頭からずらすのがよいか、後ろからずらすのがよいかを考えましょう。

設問3 空きリスト中の組の個数は、「割り済み領域の個数」+1個です。

### 設問1 関数Freeの処理

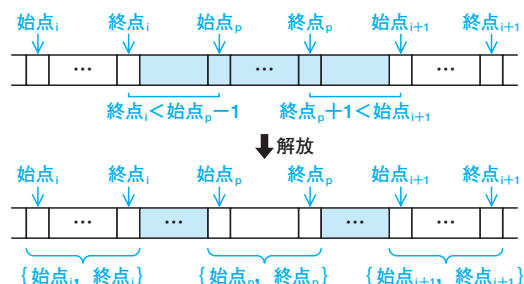
空欄a：関数 Free の引数の状況が「終点<sub>i</sub> = 始点<sub>p</sub> - 1 かつ 終点<sub>p</sub> + 1 = 始点<sub>i+1</sub>」の場合は、2組の空き領域の間にある領域をすべて解放します。その結果は、次のような1組の空き領域になります。



以上から、空欄aには**ア**の{始点<sub>i</sub>, 終点<sub>i+1</sub>}が入ります。

空欄b：関数 Free の引数の状況が「終点<sub>i</sub> < 始点<sub>p</sub> - 1 かつ 終点<sub>p</sub> + 1 < 始点<sub>i+1</sub>」の場合、割り済み領域の中間にある領域を解放します。この結果、空きリス

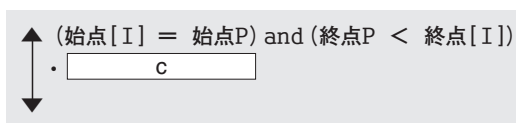
ト中の組は次のような3組に分かれます。



以上から、空欄bには**エ**の{始点<sub>p</sub>, 終点<sub>p</sub>}が入ります。

### 設問2 プログラムの完成

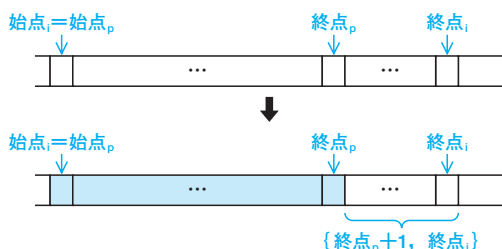
空欄c：空欄の前後は次のとおりです。



条件 (始点[I] = 始点P) and (終点P < 終点

[I]) は、問題文の表 1 の引数の状況が「始点<sub>i</sub> = 始点<sub>p</sub> かつ 終点<sub>p</sub> < 終点<sub>i</sub>」の場合に相当します。

このような場合、関数 Alloc は、空き領域の先頭からの部分を割り済みにします。結果は、次のような 1 組の空き領域になります。

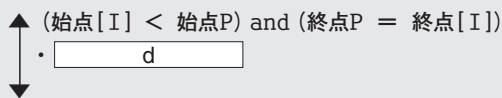


これは、空きリストの組 { 始点<sub>i</sub>, 終点<sub>i</sub> } を組 { 終点<sub>p</sub> + 1, 終点<sub>i</sub> } に置き換える操作です。終点<sub>i</sub> の値は変わらないので、始点<sub>i</sub> の値を終点<sub>p</sub> + 1 に更新すればよいでしょう。擬似言語で書くと、次のようになります。

・ 始点 [I] ← 終点 P + 1

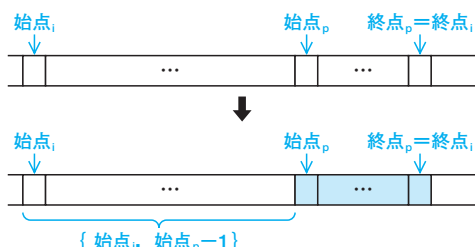
以上から、空欄 c には **1** が入ります。

空欄 d : 空欄の前後は次のとおりです。



条件 (始点 [I] < 始点 P) and (終点 P = 終点 [I]) は、問題文の表 1 の引数の状況が「始点<sub>i</sub> < 始点<sub>p</sub> かつ 終点<sub>p</sub> = 終点<sub>i</sub>」の場合に相当します。

このような場合、関数 Alloc は、空き領域の途中から最後尾までを割り済みにします。結果は、次のような 1 組の空き領域になります。

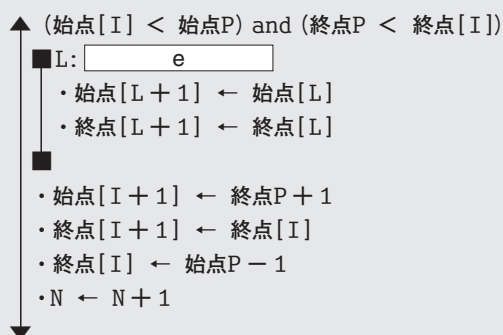


これは、空きリストの組 { 始点<sub>i</sub>, 終点<sub>i</sub> } を組 { 始点<sub>i</sub>, 始点<sub>p</sub> - 1 } に置き換える操作です。始点<sub>i</sub> の値は変わらないので、終点<sub>i</sub> の値を始点<sub>p</sub> - 1 に更新すればよいでしょう。擬似言語で書くと、次のようになります。

・ 終点 [I] ← 始点 P - 1

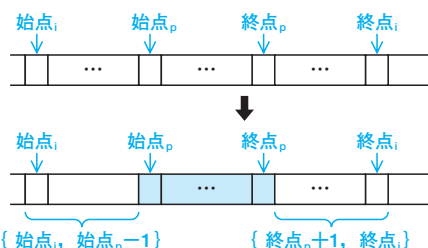
以上から、空欄 d には **2** が入ります。

空欄 e : 空欄の前後は次のとおりです。

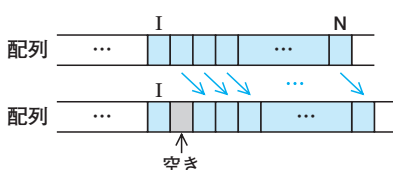


条件 (始点 [I] < 始点 P) and (終点 P < 終点 [I]) は、問題文の表 1 の引数の状況が「始点<sub>i</sub> < 始点<sub>p</sub> かつ 終点<sub>p</sub> < 終点<sub>i</sub>」の場合に相当します。

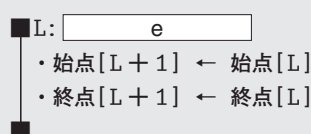
このような場合、空き領域は次のように 2 つに分割されます。



割当て後、組 { 始点<sub>i</sub>, 終点<sub>i</sub> } は { 始点<sub>i</sub>, 始点<sub>p</sub> - 1 } と { 終点<sub>p</sub> + 1, 終点<sub>i</sub> } の 2 つに分割されます。組の数が 1 つ増えるので、配列「始点」と配列「終点」の要素を、1 つずつ後ろにずらしていかなければなりません。



この操作を行っているのが、次の繰返し処理になります。





配列の各要素を後ろにずらしていくときは、配列の最後尾から前方に向かって順に行わなければならないことに注意しましょう。前から後方に向かってずらしていくと、ずらす前の要素を上書きしてしまいます。

したがって、繰返し処理の初期値は  $N$  で、増分は  $-1$  です。また、繰返しは  $I + 1$  番目の要素をずらせば終了するので、条件式は「 $L \geq I + 1$ 」となります。

以上から、空欄 e には **ウ** の「 $N, L \geq I + 1, -1$ 」が入ります。

### 設問 3 アルゴリズムの特徴

空欄 f：始点  $1$  に値  $-\infty$ 、終点  $N$  に値  $+\infty$  を設定すると、割当済み領域が 1 つもない場合でも、空きリストには組  $\{-\infty, +\infty\}$  が 1 個だけ存在します。

関数 Alloc のアルゴリズムは、空きリストに最低でも 1 個の組があることが前提になっているので、このように空きリストが空にならないようにする必要があるのです。以上から、**ア** が正解です。

○ **ア** 正解です。

× **イ** 割当済み領域が存在しなければ、空きリスト中の組は 1 個になります。この場合、関数 Free は「一部又は全体が割当済みでない」を表示して処理を終了します。以上から、関数 Free の実行時に空きリスト中の組が必ず 2 個以上あるとは限りません。

× **ウ** このアルゴリズムでは、始点  $1$  と終点  $N$  の値はそれぞれ  $-\infty, +\infty$  のまま、変化することはありません。

× **エ** 連続した空きセルは、 $-\infty, +\infty$  の設定がなくても 1 つの組で表せます。

空欄 g：空きリスト中の組の個数は、割当済み領域の個数 + 1 個になります。割当済み領域の個数は、次のようにセルを 1 個おきに割当済みにしたとき最も多くなり、その個数は  $(E + 1) \div 2$  個（小数点以下切捨て）と表せます。

E=5 の場合

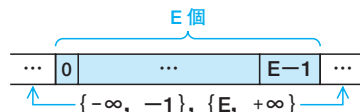
... 0 1 2 3 4 ... ← 割当済み： $(5+1) \div 2 = 3$  個  
 $\{-\infty, -1\}, \{1, 1\}, \{3, 3\}, \{5, +\infty\}$  ←  $3+1=4$  組

E=6 の場合

... 0 1 2 3 4 5 ... ← 割当済み： $(6+1) \div 2 = 3.5 \rightarrow 3$  個  
 $\{-\infty, -1\}, \{1, 1\}, \{3, 3\}, \{5, +\infty\}$  ←  $3+1=4$  組

以上から、空きリスト中の組の個数は、最大で「 $(E + 1) \div 2 + 1$ 」個になります。空欄 g は **エ** です。

空欄 h：領域中のセルがすべて割当済みの場合、空きリスト中の組は次のように 2 個になります。空欄 h は **イ** です。



### 解答

設問 1 a - **ア**, b - **エ**

設問 2 c - **イ**, d - **ウ**, e - **ウ**

設問 3 f - **ア**, g - **エ**, h - **イ**

次の問 9 から問 13 までの 5 問については、この中から 1 問を選択し、選択した問題については、答案用紙の選択欄の (選) をマークして解答してください。  
 なお、2 問以上マークした場合には、はじめの 1 問について採点します。

## 問 9 次の C プログラムの説明及びプログラムを読んで、設問 1, 2 に答えよ。

### 〔プログラムの説明〕

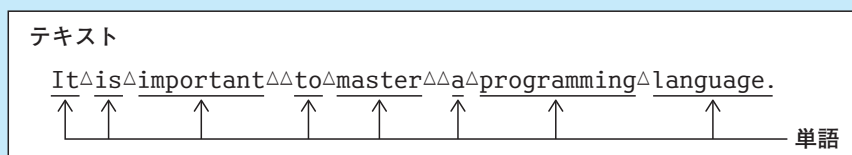
関数 format\_text は、印字したときに単語が行末で切れないようにテキストを編集してファイルに出力するプログラムである。

(1) テキストに含まれる文字は、次のものである。

- ① 英字 A ~ Z, a ~ z
- ② 数字 0 ~ 9

- ③ 記号 ! " # % & ' ( ) \* + , - . / : ; < = > ? [ ] ^ \_ { | } ~
- ④ 空白文字
- ⑤ 改行文字

(2) 単語は、空白文字及び改行文字を含まない文字列であり、空白文字又は改行文字で区切られている。単語の文字数は 20 以下とする。図 1 に、テキストに含まれる単語の例を示す。



注記 “△” は空白文字を表す。

図 1 テキストに含まれる単語の例

- (3) 改行文字を除き、1 行分として印字できる文字数（以下、最大文字数という）をプログラムの引数で与えるが、その値は 40 以上とする。単語の途中で最大文字数を超える場合は、単語の直前に改行文字を出力し、その単語が次の行の先頭に印字されるようにする。
- (4) テキスト中の空白文字及び改行文字は、そのまま出力する。ただし、連続する空白文字を出力している途中で最大文字数を超える場合は改行文字を出力し、最大文字数を超える分は次の行の先頭から印字されるようにする。
- (5) 関数 `format_text` の引数は次のとおりである。ここで、引数の値に誤りはないものとする。

`in_file`      編集前のテキスト（以下、入力テキストという）が格納されているファイル名  
`out_file`    編集後のテキスト（以下、出力テキストという）を格納するファイル名  
`width`        最大文字数

(6) プログラムを実行したときは、図 2 に示すようになる。

(入力テキスト)					
The△△△△Information△△△Technology△△△Engineers ↓					
△△△Examination△△△was△△first△△administered△△in△1969. △In△1970, ↓					
△it△became△△a△national△examination. △↓					
△△△△△Since△its△commencement, △the△examination△has△played△△△↓					
△△△△an△important△role△△△in△the△development△of△IT△△△engineers. △↓					
(出力テキスト)					
The△△△△Information△△△Technology△△△Engineers ↓					
△△△Examination△△△was△△first△△administered△△in△↓					
1969. △In△1970, ↓					
△it△became△△a△national△examination. △↓					
△△△△△Since△its△commencement, △the△examination△has ↓					
△played△△△↓					
△△△△an△important△role△△△in△the△development△of△IT△△↓					
△△engineers. △↓					
1	10	20	30	40	50
印字したときの文字位置					

注記 “△” は空白文字を、“↓” は改行文字を表す。

図 2 最大文字数を 50 とした場合の例

## 〔プログラム〕

(行番号)

```
1 #include <stdio.h>

2 #define WLEN_MAX 20 /*単語の最大長 */

3 void format_text(char *, char *, int);

4 void format_text(char *in_file, char *out_file, int width) {
5     FILE *ifp, *ofp;
6     int ch, /* 入力した文字 */
7         lpos = 0, /* 出力処理をしている行での出力済み文字数 */
8         sp = 0; /* strに格納されている文字数 */
9     char str[WLEN_MAX + 2]; /* 単語を含む出力用文字列 */

10    ifp = fopen(in_file, "r");
11    ofp = fopen(out_file, "w");

12    while ((ch = fgetc(ifp)) != EOF) {
13        if (ch == '\n') {
14            str[sp++] = ch;
15            str[sp] = '\0';
16            fputs(str, ofp);
17            ;
18            sp = 0;
19        } else if (ch == ' ') {
20            lpos += sp;
21            if (lpos >= width) {
22                str[sp++] = ;
23                lpos = 0;
24            }
25            str[sp] = '\0';
26            fputs(str, ofp);
27            fputc(ch, ofp);
28            lpos++;
29            ;
30        } else {
31            if () {
32                fputc('\n', ofp);
33                lpos = 0;
34            }
35            str[sp++] = ch;
36        }
37    }
38    str[sp] = '\0';
```

```

39     fputs(str, ofp);

40     fclose(ifp);
41     fclose(ofp);
42 }

```

**設問 1** 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。

a に関する解答群

- |                                     |                                       |   |
|-------------------------------------|---------------------------------------|---|
| <input type="checkbox"/> ア lpos = 0 | <input type="checkbox"/> イ lpos = sp  | <input type="checkbox"/> ウ lpos = width |
| <input type="checkbox"/> エ lpos++   | <input type="checkbox"/> オ lpos += sp |   |

b に関する解答群

- |                                 |                                 |                                |                               |
|---------------------------------|---------------------------------|--------------------------------|-------------------------------|
| <input type="checkbox"/> ア '\0' | <input type="checkbox"/> イ '\n' | <input type="checkbox"/> ウ ' ' | <input type="checkbox"/> エ ch |
|---------------------------------|---------------------------------|--------------------------------|-------------------------------|

c に関する解答群

- |   |   |                                 |
|---|---|---------------------------------|
| <input type="checkbox"/> ア sp = 0         | <input type="checkbox"/> イ sp = lpos      | <input type="checkbox"/> ウ sp++ |
| <input type="checkbox"/> エ str[lpos] = ch | <input type="checkbox"/> オ str[sp++] = ch |                                 |

d に関する解答群

- |  |   |
|--|---|
| <input type="checkbox"/> ア lpos > 0        | <input type="checkbox"/> イ lpos >= width        |
| <input type="checkbox"/> ウ lpos > sp       | <input type="checkbox"/> エ sp > 0               |
| <input type="checkbox"/> オ (lpos + sp) > 0 | <input type="checkbox"/> カ (lpos + sp) >= width |

**設問 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。

テキストの編集処理のうち、行頭の空白文字に関する処理を変更することになった。これに合わせて関数 `format_text` に、処理を追加する。ここで、プログラム中の  a ～  d には正しい答えが入っているものとする。

(1) テキストの編集処理の変更内容は、次のとおりである。

- ① 入力テキスト中の 1 文字以上の連続する空白文字列で、印字したときに行頭に来る部分は出力しない。
- ② 最初に出力する単語の場合、又は行の先頭に来る単語の場合で直前に出力した単語の最後の文字が “.” であったときは、単語の直前に空白文字を一つ出力する。

(2) 処理を変更したプログラムを実行したときは、図 3 に示すようになる。

(入力テキスト)
The△△△Information△△△Technology△△△Engineers↓ △△△Examination△△△was△△first△△administered△△in△△1969.△In△1970, ↓ △it△became△a△national△examination.△↓ △△△△△Since△its△commencement, △the△examination△has△played△△△↓ △△△an△important△role△△△in△the△development△of△IT△△△△engineers.△↓
(出力テキスト)
△The△△△Information△△△Technology△△△Engineers↓ Examination△△△was△△first△△administered△△in△△1969.△↓ △In△1970, ↓ it△became△a△national△examination.△↓ △Since△its△commencement, △the△examination△has△↓ played△△△↓ an△important△role△△△in△the△development△of△IT△△△△↓ engineers.△↓
-----+-----+-----+-----+-----+ 1          10          20          30          40          50 印字したときの文字位置

注記 “△” は空白文字を, “↓” は改行文字を表す。

図3 処理を変更し, 最大文字数を 50 とした場合の例

(3) 処理の変更に対応するために, プログラムを表1のとおりに変更する。

表1 プログラムの変更内容

処置	変更内容
行番号8と9の間に追加	int lch = '.';
行番号26と27の間に追加	if ( <input type="text" value="e"/> ) {
行番号28と29の間に追加	}
行番号34と35の間に追加	if ((lpos == 0) && (sp == 0) && (lch == '.')) { fputc(' ', ofp); lpos++; } <input type="text" value="f"/> ;

eに関する解答群

- |                                       |                                       |                                      |
|---------------------------------------|---------------------------------------|--------------------------------------|
| <input type="checkbox"/> ア lpos == 0  | <input type="checkbox"/> イ lpos == sp | <input type="checkbox"/> ウ lpos != 0 |
| <input type="checkbox"/> エ lpos != sp | <input type="checkbox"/> オ sp == 0    | <input type="checkbox"/> カ sp != 0   |

fに関する解答群

- |  |  |                                     |
|--|--|-------------------------------------|
| <input type="checkbox"/> ア lch = ' '       | <input type="checkbox"/> イ lch = '.'       | <input type="checkbox"/> ウ lch = ch |
| <input type="checkbox"/> エ str[sp++] = ' ' | <input type="checkbox"/> オ str[sp++] = lch |                                     |

## 問 9

## 午後のカギ

テキストを整形するテキストフォーマッタは、C 言語の初歩的なプログラムとして、よく取り上げられるテーマです。本問は文字列処理に配列を使っており、ポインタを使う場合より難易度は低めでしょう。

**設問 1** 1 文字ずつ読み込んだ文字によって処理を場合分けするので、空欄の部分でどんな処理が必要かを考えます。

**設問 2** プログラムの変更部分で行う処理を、問題文の変更内容の記述から読み取ります。

### 設問 1 プログラムの完成

**空欄 a** : まず、プログラム内で使われている各変数の役割を確認しておきましょう。

ch	入力した文字
lpos	出力処理をしている行での出力済み文字数
sp	str に格納されている文字数
str	単語を含む出力用文字列

プログラムは、入力ファイルから 1 文字ずつ変数 ch に読み込み、文字列 str に格納していきます。str に 1 単語分の文字が格納されたら出力ファイルに書き出しますが、読み込んだ文字の種類によっては、特別な処理が必要になります。

**空欄 a** は、読み込んだ文字が改行文字 ('\\n') だった場合の処理です。

```
13 if (ch == '\\n') {
14     str[sp++] = ch;
15     str[sp] = '\\0';
16     fputs(str, ofp);
17     a;
18     sp = 0;
19 } ...
```

読み込んだ文字が改行文字のときは、単語はその前の文字で終わっているの、str に格納されている単語を出力し、改行を出力します (行番号 14 ~ 16)。

単語と改行を出力したので、次の文字は単語の 1 文字目で、次の行の先頭になります。したがって、str の格納文字数 sp と、出力済み文字数 lpos は、どちらもゼロにリセットする必要があります。

sp については行番号 18 でリセットしているので、**空欄 a** には lpos をゼロにリセットする処理が入ります。

以上から、**空欄 a** は **ア** の「lpos = 0」です。

**空欄 b** : プログラムの前後は次のとおりです。

```
19 } else if (ch == ' ') {
20     lpos += sp;
21     if (lpos >= width) {
22         str[sp++] = b;
23         lpos = 0;
24     }
25     str[sp] = '\\0';
26     fputs(str, ofp);
27     fputc(ch, ofp);
28     lpos++;
29     c;
30 } ...
```

読み込んだ文字が空白文字だった場合は、単語の区切りを意味するので、str に格納されている文字列を出力し (行番号 26)、続いて空白文字を出力します (行番号 27)。

ただし、str を出力したとき、現在の行がちょうど 1 行の最大文字数に達する場合は、単語の末尾で改行を出力し、空白文字は次の行の先頭に出力する必要があります (最大文字数を超える場合は行番号 30 以下に分岐するので、ここでは考慮しません)。

・str を出力しても最大文字数にならない場合 :

.....has△  
 ①strを出力 ↑ ②空白文字を出力 ↑

・str を出力すると最大文字数に達する場合 :

.....has↓  
 △  
 ③空白文字を出力 ↑ ②改行文字を出力 ↓ ①strを出力 ↑

行番号 21 ~ 24 は、str の出力によって 1 行の最大文字数に達する場合の処理になります。行番号 22 では、str の末尾に改行文字を追加し、sp の値を 1 増やします。

```
str[sp++] = '\n';
```

以上から、空欄 b には **イ** の「'\n'」が入ります。

**空欄 c**：単語を出力した後は、str の内容をリセットするため、sp の値をゼロに戻す必要があります。行番号 29 で行うのはこの処理です。

```
sp = 0;
```

以上から、空欄 c には **ア** の「sp = 0」が入ります。

**空欄 d**：プログラムの前後は次のとおりです。

```
30 } else {
31     if (   d   ) {
32         fputc('\n', ofp);
33         lpos = 0;
34     }
35     str[sp++] = ch;
36 }
```

読み込んだ文字が改行でも空白でもなければ、単語の一部として str に格納し、sp の値を 1 増やします (行番号 35)。

ただし、1 行の最大文字数に達してもまだ単語が続く場合には、その単語は次の行の先頭に出力するので、いったん改行します。この処理を行っているのが行番号 31 ~ 34 です。

**空欄 d** の条件式は、現在の出力文字数 (lpos) と、現在の str の文字数 (sp) の合計が、1 行の最大文字数以上になるかどうかを、次のように調べます。

```
(lpos + sp) >= width
```

以上から、空欄 d には **カ** が入ります。

## 設問 2 プログラムの変更

**空欄 e**：プログラムの変更内容に、

- ① 入力テキスト中の 1 文字以上の連続する空白文字列で、印字したときに行頭に来る部分は出力しない。

とあります。この変更に対応して、空白文字を出力する部分 (行番号 27, 28) が、次のように変更されます。

```
if (   e   ) {
    fputc(ch, ofp);
    lpos++;
}
```

変更後のプログラムでは、空欄 e に入る条件式が真のときのみ、空白文字を出力ようになります。「空白文字が行頭に来るときは印字しない」→「行頭でなければ空白文字を印字する」ので、条件式は「行頭ではない」とき真になる式です。これは、変数 lpos を使って、

```
lpos != 0
```

のようにすれば表すことができます。以上から、空欄 e は **ク** の「lpos != 0」です。

**空欄 f**：プログラムのもう 1 つの変更は次のようなものです。

- ② 最初に出力する単語の場合、又は行の先頭に来る単語の場合で直前に出力した単語の最後の文字が "." であったときは、単語の直前に空白文字を一つ出力する。

この変更に対応して、プログラムに次の部分を追加します (行番号 34 と 35 の間に追加)。

```
if((lpos == 0) && (sp == 0) && (lch == '.')) {
    fputc(' ', ofp);
    lpos++;
}
  f   ;
```

次の単語が行の先頭に来る単語で、かつ、直前に出力した単語の最後の文字が "." かどうかを、変数 lpos、変数 sp、変数 lch の値を調べて判断しています。

このうち変数 lch には、直前に出力した文字が格納されていなければなりません。変数 lch に値を代入する文が見当たりません。**空欄 f** にはこの代入文が入ります。

現在の変数 ch の値を lch に代入しておけば、次の文字を読み込んだとき、lch は直前の文字を表すことになります。したがってこの代入文は、



lch = ch;

とすればよいでしょう。空欄 f は **ウ** の「lch = ch」です。

○ 解答 ○

設問 1 a - **ア**, b - **イ**, c - **ア**,  
d - **カ**

設問 2 e - **ウ**, f - **ウ**

## 問 10 次の COBOL プログラムの説明及びプログラムを読んで、設問 1, 2 に答えよ。

### 〔プログラムの説明〕

X 社では、従業員の英語力向上のために、定期的に英語の検定テストを実施している。従業員は誰でも希望すれば受検することができ、直近の 5 回分の検定結果が得点マスタファイルに記録される。このプログラムは、1 回分の検定結果が記録された採点ファイルを読み込み、検定結果を得点マスタファイルに反映する。

- (1) 得点マスタファイルは、**図 1** に示すレコード様式の順ファイルで、従業員の得点履歴を管理する。

従業員番号 6桁	平均得点 3桁	得点履歴	
		受検日 8桁	得点 3桁

5回分繰返し

図 1 得点マスタファイルのレコード様式

- ① 全ての従業員に対するレコードが、従業員番号の昇順に格納されている。
- ② 平均得点には、得点履歴に記録された得点の平均点が小数点以下切捨てで格納される。
- ③ 得点履歴には、直近 5 回分の受検日と得点が、受検日の降順に格納される。受検回数が 5 回に満たない場合、残りの得点履歴の受検日及び得点にはゼロが格納される。
- ④ 受検日には、年、月、日が、それぞれ 4 桁、2 桁、2 桁の西暦で格納される。
- ⑤ 得点には、当該受検日の得点が格納される。得点は 000 ～ 100 である。

- (2) 採点ファイルは、**図 2** に示すレコード様式の順ファイルで、英語の検定テストの都度、作成される。

#### 様式1

受検日 8桁	未使用域 1桁
-----------	------------

#### 様式2

従業員番号 6桁	得点 3桁
-------------	----------

図 2 採点ファイルのレコード様式

- ① ファイルの先頭レコードには、様式1のレコードが1レコードだけ格納される。受検日には、年、月、日が、それぞれ4桁、2桁、2桁の西暦で格納される。
- ② 2レコード目以降には、様式2のレコードが、従業員番号の昇順に格納される。なお、受検は希望者だけなので、この回に受検しなかった従業員のレコードは存在しない。

## 〔プログラム〕

(行番号)

```
1 DATA DIVISION.
2 FILE SECTION.
3 FD MST-FILE.
4 01 MST-REC.
5     02 MST-ENUM      PIC 9(6).
6     02 MST-AVG       PIC 9(3).
7     02 MST-HST.
8         03 MST-ELM    OCCURS 5.
9             04 MST-DATE PIC 9(8).
10            04 MST-SCORE PIC 9(3).
11 FD SCR-FILE.
12 01 FIRST-REC.
13     02 SCR-DATE      PIC 9(8).
14     02 FILLER        PIC X(1).
15 01 SCR-REC.
16     02 SCR-ENUM      PIC 9(6).
17     02 SCR-SCORE     PIC 9(3).
18 WORKING-STORAGE SECTION.
19 77 SCR-FLAG          PIC X(1) VALUE SPACE.
20     88 SCR-EOF        VALUE "E".
21 77 W-CNT              PIC 9(2).
22 77 TEST-DATE          PIC 9(8).
23 77 SCR-TOTAL          PIC 9(4).
24 77 HST-NUM            PIC 9(1).
25 PROCEDURE DIVISION.
26 MAIN-PROC.
27     OPEN INPUT SCR-FILE
28         I-O MST-FILE.
29     READ SCR-FILE.
30     MOVE SCR-DATE TO TEST-DATE.
31     PERFORM MATCHING-PROC.
32     CLOSE MST-FILE SCR-FILE.
33     STOP RUN.
34 MATCHING-PROC.
35     PERFORM UNTIL SCR-EOF
```

```

36      READ SCR-FILE
37      AT END
38      SET SCR-EOF TO TRUE
39      NOT AT END
40      PERFORM [a]
41      READ MST-FILE END-READ
42      END-PERFORM
43      PERFORM UPDATE-PROC
44      END-READ
45      END-PERFORM.
46  UPDATE-PROC.
47      MOVE ZERO TO SCR-TOTAL HST-NUM.
48      PERFORM [b]
49      MOVE MST-ELM(W-CNT) TO MST-ELM(W-CNT + 1)
50      IF [c] THEN
51          ADD 1 TO HST-NUM
52          ADD MST-SCORE(W-CNT) TO SCR-TOTAL
53      END-IF
54      END-PERFORM.
55      MOVE TEST-DATE TO MST-DATE([d]).
56      MOVE SCR-SCORE TO MST-SCORE([d]).
57      COMPUTE MST-AVG = (SCR-TOTAL + SCR-SCORE) / (HST-NUM + 1).
58      REWRITE MST-REC.

```

**設問 1** 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。

a, b に関する解答群

- ☐ ア TEST AFTER UNTIL SCR-ENUM = MST-ENUM
- ☐ イ TEST AFTER VARYING W-CNT FROM ZERO BY 1 UNTIL W-CNT > 5
- ☐ ウ TEST BEFORE UNTIL SCR-ENUM NOT = MST-ENUM
- ☐ エ TEST BEFORE VARYING W-CNT FROM 1 BY 1 UNTIL W-CNT > 4
- ☐ オ TEST BEFORE VARYING W-CNT FROM 4 BY -1 UNTIL W-CNT = ZERO

c に関する解答群

- ☐ ア MST-DATE(W-CNT) = ZERO
- ☐ イ MST-DATE(W-CNT) NOT = ZERO
- ☐ ウ MST-SCORE(W-CNT) = ZERO
- ☐ エ MST-SCORE(W-CNT) NOT = ZERO

d に関する解答群

- ☐ ア 1
- ☐ イ 5
- ☐ ウ HST-NUM
- ☐ エ W-CNT

**設問2** 検定結果の反映時に、今回の検定における得点の上位10名の従業員番号と得点を表示するようプログラムを変更する。このとき、10人目の受検者と同得点の受検者が複数いた場合は、当該受検者を全て表示する。表中の  に入れる正しい答えを、解答群の中から選べ。

表1 プログラムの変更内容

処置	変更内容
行番号2と3の間に追加	SD SRT-FILE. 01 SRT-REC. 02 SRT-ENUM PIC 9(6). 02 SRT-SCORE PIC 9(3).
行番号24と25の間に追加	77 DISP-FLAG PIC X(1) VALUE SPACE. 88 DISP-END VALUE "E". 77 PREV-SCORE PIC 9(3) VALUE ZERO.
行番号31を変更	SORT SRT-FILE DESCENDING KEY SRT-SCORE INPUT PROCEDURE IS MATCHING-PROC OUTPUT PROCEDURE IS DISPLAY-PROC.
<input type="text"/> e に追加	MOVE SCR-REC TO SRT-REC RELEASE SRT-REC
行番号58の後ろに追加	DISPLAY-PROC. PERFORM TEST BEFORE VARYING W-CNT FROM 1 BY 1 UNTIL DISP-END RETURN SRT-FILE AT END SET DISP-END TO TRUE NOT AT END IF <input type="text"/> f <input type="text"/> THEN SET DISP-END TO TRUE ELSE DISPLAY SRT-ENUM ": " SRT-SCORE <input type="text"/> g END-IF END-RETURN END-PERFORM.

eに関する解答群

- ☐ ア 行番号34と35の間      ☐ イ 行番号41と42の間  
☐ ウ 行番号43と44の間      ☐ エ 行番号53と54の間

fに関する解答群

- ☐ ア W-CNT = 10 AND SRT-SCORE = PREV-SCORE  
☐ イ W-CNT = 10 AND SRT-SCORE > PREV-SCORE  
☐ ウ W-CNT > 10 AND SRT-SCORE < PREV-SCORE

エ W-CNT > 10 AND SRT-SCORE = PREV-SCORE

gに関する解答群

- ア ADD 1 TO W-CNT
- イ ADD SRT-SCORE TO PREV-SCORE
- ウ MOVE PREV-SCORE TO SRT-SCORE
- エ MOVE SRT-SCORE TO PREV-SCORE

## 問 10 午後のカギ

テスト結果を集計する COBOL プログラムの問題です。各設問のヒントを参照してください。

設問 1 PERFORM 文で、繰返し処理の終了条件を後判定にする場合は「TEST AFTER」を、前判定にする場合は「TEST BEFORE」を指定します。どちらも省略した場合は前判定になります。

設問 2 SORT 文は COBOL 問題ではよく出題されます。入力処理で使われる RELEASE 文や、出力処理で使われる RETURN 文と合わせて、よく理解しておきましょう。

### 設問 1 プログラムの完成

空欄 a：空欄の前後は次のとおりです。

```
36 READ SCR-FILE
37     AT END
38     SET SCR-EOF TO TRUE
39     NOT AT END
40     PERFORM a
41     READ MST-FILE END-READ
42     END-PERFORM
43     PERFORM UPDATE-PROC
44 END-READ
```

段落 MATCHING-PROC では、採点ファイルから従業員の採点レコードを 1 件ずつ読み取り、そのデータをもとに、得点マスタファイルの該当する従業員のレコードを更新します。

行番号 40～42 は、採点レコードの従業員番号 SCR-ENUM に一致する従業員のレコードを、得点マスタファイル MST-FILE から検索する処理です。MST-FILE のレコードは従業員番号 MST-ENUM の昇順に並んでいるので、MST-ENUM と SCR-ENUM が一致する

まで、MST-FILE を先頭から順に調べていきます。この処理は、PERFORM 文による繰返し処理になります。

空欄には、PERFORM 文に指定する繰返しの終了条件が入ります。

SCR-ENUM と MST-ENUM が一致したら繰返しを終了するので、終了条件は明らかに

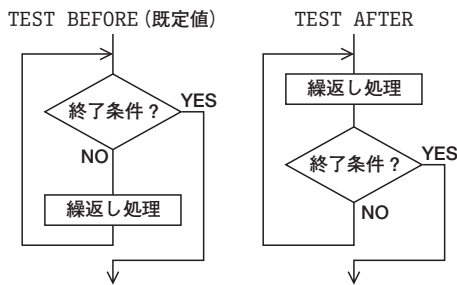
UNTIL SCR-ENUM = MST-ENUM

です。ただしこのプログラムでは、繰返し処理に入る前に、得点マスタファイルのレコードを読み込む命令がありません。そのため、上の条件判定は、得点マスタファイルからレコードを読み込んだ後に行う必要があります。

このような場合は、PERFORM 文に「TEST AFTER」を指定します。

PERFORM TEST AFTER UNTIL 終了条件  
繰返し処理  
END-PERFORM

こうすると、PERFORM 文はとりえず繰返し処理の内容を実行した後に終了条件を判定し、繰返し処理を続行するかどうか判断します。



後判定では「繰返し処理」が最低1回は必ず実行されるんだね。



以上から、空欄 a には **ア** の「TEST AFTER UNTIL SCR-ENUM = MST-ENUM」が入ります。

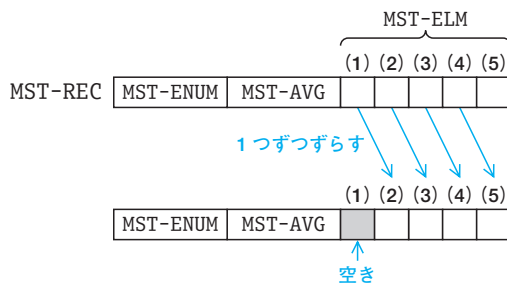
なお、繰返し処理を実行する前に条件判定を行う場合は、PERFORM 文に「TEST BEFORE」を指定します。ただし、「TEST BEFORE」を指定しなければ前判定となるので、この指定は多くの場合省略されます。

空欄 b：空欄の前後は次のとおりです。

```
47 MOVE ZERO TO SCR-TOTAL HST-NUM.
48 PERFORM b
49 MOVE MST-ELM(W-CNT) TO
    MST-ELM(W-CNT + 1)
    :
54 END-PERFORM.
```

段落 UPDATE-PROC では、得点マスタファイルのレコードを、採点レコードにしたがって更新します。

行番号 48～54 の繰返し処理では、レコードに記録されている得点履歴 MST-ELM の内容を、1 つずつ後ろにずらします。



得点履歴は受検日の降順に格納するので、新しい受検結果を先頭に格納するために、後ろにずらす必要があります。この作業は

- ① MST-ELM(4) の内容を MST-ELM(5) へ転記
- ② MST-ELM(3) の内容を MST-ELM(4) へ転記
- ③ MST-ELM(2) の内容を MST-ELM(3) へ転記
- ④ MST-ELM(1) の内容を MST-ELM(2) へ転記

の順番に行うことに注意しましょう。順番を逆に実行すると、更新前の内容を上書きしてしまいます。

この処理は、COBOL プログラムで次のように記述できます。

```
PERFORM TEST BEFORE
    VARYING W-CNT FROM 4 BY -1
    UNTIL W-CNT = ZERO
    MOVE MST-ELM(W-CNT) TO MST-ELM(W-CNT + 1)
    :
END-PERFORM.
```

以上から、空欄 b には、**オ**が入ります。

空欄 c：空欄の前後は次のとおりです。

```
50 IF c THEN
51 ADD 1 TO HST-NUM
52 ADD MST-SCORE(W-CNT) TO SCR-TOTAL
53 END-IF
```

得点履歴 MST-ELM(1)～(4)を後ろにずらすたびに、HST-NUMを1増やし、その得点を SCR-TOTAL に加算していきます。これは、後ほど「SCR-TOTAL / HST-NUM」で平均点を算出するために必要な処理です。

受検履歴が5回に満たない従業員のレコードでは、MST-ELM(W-CNT)の内容が空になる場合があります。その場合は HST-NUM や SCR-TOTAL を加算すると正しい平均点になりませんから、行番号 51, 52 は

「MST-ELM(W-CNT) の内容が空ではない」

場合にのみ実行します。空欄 c には、このことを意味する条件式が入ります。

MST-ELM は受検日 MST-DATE と得点 MST-SCORE から構成され、内容が空の場合はどちらもゼロが格納されます。ただし、MST-SCORE には受検した結果得点が0点だったときにもゼロが格納されるので、内容が空かどうかの判定には使えません。一方、MST-DATE がゼロであれば、内容が空だとわかります。したがって条件式は

MST-DATE(W-CNT) NOT = ZERO

となります。空欄 c は **イ** です。

空欄 d：空欄の前後は次のとおりです。

```
MOVE TEST-DATE TO MST-DATE( d ).
MOVE SCR-SCORE TO MST-SCORE( d ).
```

これまでの得点履歴を1つずつ後ろにずらしたので、先頭の得点履歴に空きができます。ここに、今回の受検結果を格納します。

```
MOVE TEST-DATE TO MST-DATE(1).
MOVE SCR-SCORE TO MST-SCORE(1).
```

以上から、空欄 d は **ア** です。

## 設問2 プログラムの変更

空欄 e：変更後のプログラムでは、SORT 文を使って次のように入力処理と出力処理を行います。

```
SORT SRT-FILE DESCENDING KEY SRT-SCORE
INPUT PROCEDURE IS MATCHING-PROC
OUTPUT PROCEDURE IS DISPLAY-PROC.
```

入力処理で SRT-FILE に入力された各レコードが、出力処理で SRT-SCORE の降順に整列されて出力されます。

入力処理 MATCHING-PROC では、採点ファイルからレコードを1件ずつ読み込んで、整列レコードを作り、整列ファイルに書き出します。この処理を行うために、次の内容をプログラムに追加します。

```
MOVE SCR-REC TO SRT-REC
RELEASE SRC-REC
```

この処理は、採点ファイルからレコードを1件読み込むたびに1回実行します。ただし、ファイル末尾の場合はレコードがありませんから、実行するタイミングは行番号39の「NOT AT END」以降のブロックです。

このうち、行番号40～42のPERFORM文の間は繰返し処理になってしまうので、追加できる位置は行番号39と40の間、行番号42と43の間、行番号43と44の間の3か所だけです。解答群の中にあるのは **ウ** の「行番号43と44の間」だけなので、空欄 e には **ウ** が入ります。

空欄 f：出力処理 DISPLAY-PROC では、整列ファイル SRT-FILE から、RETURN 文で整列レコードを1件ずつ読み込んで表示します。

```
RETURN SRT-FILE
AT END
SET DISP-END TO TRUE
NOT AT END
IF f THEN
SET DISP-END TO TRUE
ELSE
DISPLAY SRT-ENUM ": " SRT-SCORE
g
END-IF
END-RETURN
```

表示するのは得点の上位10名だけなので、W-CNT（1名表示するたびに1増える）が10より大きい場合は表示を終了します。ただし、10人目と同得点の受検者がいればすべて表示するので、表示を終了する条件は、「W-CNT が10より大きく、かつ、得点 SRT-SCORE が前の受検者より低い場合」です。

空欄 f には、この条件を表す条件式が入ります。解答群からこの条件に当てはまる条件式を探すと、

W-CNT > 10 AND SRT-SCORE < PREV-SCORE

が見つかります。以上から、空欄 f は **ウ** です。

なお、PREV-SCORE に「前の受検者の得点」を設定する処理は、空欄 g で実行すると考えられます。

空欄 g：受検者を1人表示したら、その得点を PREV-SCORE に設定します。この後、次のレコードを読み込むと、PREV-SCORE は「前の受検者の得点」を表すことになります。

したがって空欄 g には

MOVE SRT-SCORE TO PREV-SCORE

が入ります。空欄 g は **エ** です。

### ○ 解答 ○

設問1 a - **ア**, b - **オ**, c - **イ**,  
d - **ア**

設問2 e - **ウ**, f - **ウ**, g - **エ**



**問 11** 次の Java プログラムの説明及びプログラムを読んで、設問 1, 2 に答えよ。  
(Java プログラムで使用する API の説明は、この冊子の末尾を参照してください。)

〔プログラムの説明〕

雑誌記事のオンライン購読サイトをモデルとしたプログラムである。記事には、無料記事と有料記事がある。サイトの利用者は、誰でも無料記事を閲覧できる。利用者は、名前を登録することで登録会員となり、無料記事に加え、有料記事を 1 日に 1 本だけ無料で閲覧できる。登録は無料である。登録会員は、購読料を支払うことで有料会員となり、全ての記事が無制限に閲覧できる。

このプログラムは、次のクラスから成る。

(1) クラス `Article` : 記事を表す。

各記事には、記事識別子 (`id`) が割り当てられる。各記事は、見出し (`headline`)、本文 (`body`) 及び無料記事を表すフラグ (`free`) から成る。記事識別子と記事は、1 対 1 に対応し、同一記事に対して `Article` のインスタンスは高々 1 個しか存在しない。

- ① 静的メソッド `create` : 引数で与えられた記事識別子及び記事のデータから `Article` のインスタンスを生成し、登録する。記事識別子に重複はないものとする。
- ② 静的メソッド `getArticle` : 引数で与えられた記事識別子から登録されている記事である `Article` のインスタンスを返す。記事識別子に誤りはないものとする。
- ③ 静的メソッド `getIds` : 登録されている全記事の記事識別子の集合を返す。
- ④ メソッド `isFree` : 無料記事であれば `true` を返す。そうでなければ、`false` を返す。
- ⑤ その他、記事のデータを返すメソッドが用意されている。

(2) 抽象クラス `User` : 利用者を表す。

- ① コンストラクタ : 引数で与えられた利用者名をもつインスタンスを生成する。
- ② メソッド `getName` : 利用者名を返す。
- ③ 抽象メソッド `testAndMark` : 引数で与えられた記事が閲覧可能かどうかを調べ、閲覧可能であれば `true` を返す。そうでなければ、`false` を返す。このメソッドが `true` を返したとき、サイトは利用者に記事を表示するものとし、必要に応じて記事が閲覧済みであることを記録する。

(3) クラス `Guest` : 未登録の利用者 (ゲスト) を表す。

- ① コンストラクタ : 利用者名を “ゲスト” とするインスタンスを生成する。
- ② メソッド `testAndMark` : 引数で与えられた記事が無料記事であれば `true` を返す。そうでなければ、`false` を返す。

(4) クラス `Member` : 登録会員を表す。

- ① コンストラクタ : 引数で与えられた利用者名をもつインスタンスを生成する。
- ② メソッド `testAndMark` : 引数で与えられた記事が無料記事であれば、`true` を返す。有料記事の場合は、今日最初の有料記事の閲覧のとき、又は今日閲覧済みの有料記事と同一であるときは、`true` を返す。それら以外のときは、`false` を返す。加えて、今日最初の有料記事の閲覧のときには、閲覧済みの記事として今日の日付とともに記録する。

- ③ メソッド `today` : 今日の日付 (ローカル時間) を 1970 年 1 月 1 日からの日数で返す。

(5) クラス `PaidMember` : 有料会員を表す。

- ① コンストラクタ : 引数で与えられた利用者名をもつインスタンスを生成する。
- ② メソッド `testAndMark` : 全ての記事に対して `true` を返す。

(6) クラス SubscriptionSite：上記クラスのテスト用プログラムである。メソッド main を実行すると、図 1 の結果が得られる。

なお、Member 及び PaidMember のコンストラクタに与える利用者の衝突は、ないものとする。

```
ゲスト：記事「PC入門」PC初心者…  
ゲスト：記事「スマホ特集」＜閲覧不可＞  
ゲスト：記事「アプリガイド」使えるアプリ…  
登録会員A：記事「PC入門」PC初心者…  
登録会員A：記事「スマホ特集」最新のスマホ…  
登録会員A：記事「アプリガイド」使えるアプリ…  
有料会員B：記事「PC入門」PC初心者…  
有料会員B：記事「スマホ特集」最新のスマホ…  
有料会員B：記事「アプリガイド」使えるアプリ…
```

図 1 メソッド main の実行結果

〔プログラム 1〕

```
import java.util.Map;  
import java.util.Set;  
import java.util.TreeMap;  
  
class Article {  
    private static final Map<String, Article> ARTICLES =  
        new TreeMap<String, Article>();  
    private final String id, headline, body;  
    private final boolean free;  
  
    private Article(String id, String headline,  
                    String body, boolean free) {  
        this.id = id;  
        this.headline = headline;  
        this.body = body;  
        this.free = free;  
    }  
  
    static Article create(String id, String headline,  
                        String body, boolean free) {  
        Article article = new Article(id, headline, body, free);  
        ARTICLES.put(id, article);  
        return article;  
    }  
  
    static Article getArticle(String id) {  
        return ARTICLES.get(id);  
    }  
}
```

```
static Set<String> getIds() {
    return ARTICLES.keySet();
}

String getId() { return id; }
String getHeadline() { return headline; }
String getBody() { return body; }
boolean isFree() { return free; }
}
```

## 〔プログラム 2〕

```
abstract class User {
    private final String name;

    User(String name) {
        this.name = name;
    }

    String getName() {
        return name;
    }

    abstract boolean testAndMark(Article article);
}
```

## 〔プログラム 3〕

```
class Guest extends User {
    Guest() {
        super("ゲスト");
    }

    boolean testAndMark(Article article) {
        return ;
    }
}
```

## 〔プログラム 4〕

```
import java.util.TimeZone;

class Member extends User {
    private static final long MILLIS_PER_DAY = 24 * 60 * 60 * 1000L;
    private Article browsedArticle; // 閲覧済みの有料記事
    private long browseDate = Long.MIN_VALUE; // 記事を閲覧した日付
}
```

```

Member(String name) {
    ;
}

boolean testAndMark(Article article) {
    if (article.isFree()) {
        return true;
    }
    long today = today();
    if (browseDate  today) {
        return browsedArticle == article;
    }
    // 閲覧済みの有料記事と日付を記録
    browsedArticle = article;
    browseDate = today;
    return true;
}

private long today() {
    long time = System.currentTimeMillis(); // 現時刻 (協定世界時)
    TimeZone tz = TimeZone.getDefault();
    time += tz.getOffset(time); // ローカル時間に変換
    return time / MILLIS_PER_DAY;
}
}

```

#### [プログラム 5]

```

class PaidMember extends Member {
    PaidMember(String name) {
        ;
    }

    boolean testAndMark(Article article) {
        return ;
    }
}

```

#### [プログラム 6]

```

public class SubscriptionSite {
    public static void main(String[] args) {
        User[] readers = {
            new Guest(),
            new Member("登録会員A"),
            new PaidMember("有料会員B")
        };
        Article.create("0001", "PC入門", "PC初心者...", true);
    }
}

```

```

Article.create("0002", "スマホ特集", "最新のスマホ...", false);
Article.create("0003", "アプリガイド", "使えるアプリ...", true);
for (  e ) {
    for (  f ) {
        Article article = Article.getArticle(id);
        String body;
        if (reader.testAndMark(article)) {
            body = article.getBody();
        } else {
            body = " <閲覧不可> ";
        }
        System.out.printf("%s: 記事「%s」 %s%n",
            reader.getName(), article.getHeadline(), body);
    }
}

```

**設問 1** 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。

a, d に関する解答群

- |  |  |
|--|--|
| <input type="checkbox"/> ア !article.isFree()       | <input type="checkbox"/> イ article.isFree()        |
| <input type="checkbox"/> ウ false                   | <input type="checkbox"/> エ this.article != article |
| <input type="checkbox"/> オ this.article == article | <input type="checkbox"/> カ this.article == null    |
| <input type="checkbox"/> キ true                    |  |

b に関する解答群

- |  |                                       |  |
|--|---------------------------------------|--|
| <input type="checkbox"/> ア new User(name)    | <input type="checkbox"/> イ super()    | <input type="checkbox"/> ウ super(name) |
| <input type="checkbox"/> エ super.name = name | <input type="checkbox"/> オ this()     | <input type="checkbox"/> カ this(name)  |
| <input type="checkbox"/> キ this.name = name  | <input type="checkbox"/> ク User(name) |  |

c に関する解答群

- |                               |                              |                               |                              |                                       |
|-------------------------------|------------------------------|-------------------------------|------------------------------|---------------------------------------|
| <input type="checkbox"/> ア != | <input type="checkbox"/> イ < | <input type="checkbox"/> ウ == | <input type="checkbox"/> エ > | <input type="checkbox"/> オ instanceof |
|-------------------------------|------------------------------|-------------------------------|------------------------------|---------------------------------------|

e, f に関する解答群

- |  |
|--|
| <input type="checkbox"/> ア int i = 0; i < Article.getIds().size(); i++ |
| <input type="checkbox"/> イ int i = 0; i < readers.length; i++          |
| <input type="checkbox"/> ウ Member reader : readers                     |
| <input type="checkbox"/> エ PaidMember reader : readers                 |
| <input type="checkbox"/> オ String id : Article.articles                |
| <input type="checkbox"/> カ String id : Article.getIds()                |
| <input type="checkbox"/> キ User reader : readers                       |

**設問 2** プログラム 6 で記事識別子 “0003” の記事の無料記事を表すフラグの値を false にしてプログラムを実行したとき、“<閲覧不可>”は何回出力されるか、正しい答えを、解答群の中から選べ。ここで、プログラム中の  ～  には正しい答えが入っているものとし、プログラムは日をまたいで実行しないものとする。

解答群

- |                                |                                |                                |
|--------------------------------|--------------------------------|--------------------------------|
| <input type="checkbox"/> ア 0 回 | <input type="checkbox"/> イ 1 回 | <input type="checkbox"/> ウ 2 回 |
| <input type="checkbox"/> エ 3 回 | <input type="checkbox"/> オ 4 回 | <input type="checkbox"/> カ 5 回 |

## 問 11 午後のカギ

会員の種類を、抽象クラス User から派生した複数のサブクラスとして実装して利用する Java プログラムです。

**設問 1** プログラム 3 ～ 5 は、いずれも抽象クラス User のサブクラスなので、互いに共通する点もあります。空欄の部分、他のクラスではどう処理しているかも参考にしてください。空欄 c はやや難易度が高いですが、プログラムの各部を問題文の説明に対応させて当てはまるロジックを考えます。空欄 e, f は拡張 for 文です。

**設問 2** ゲスト、登録会員、有料会員ごとに、有料記事閲覧可能かどうか考えます。

### 設問 1 プログラムの完成

空欄 a：空欄には、クラス Guest のメソッド testAndMark の戻り値が入ります。

```
boolean testAndMark(Article article) {  
    return ;  
}
```

クラス Guest のメソッド testAndMark は、引数で与えられた記事が無料記事であれば true、そうでなければ false を返します。

引数に指定するクラス Article には、無料記事のとき true、そうでなければ false を返すメソッド isFree が定義されています。このメソッドを利用すれば、引数 article が無料記事のとき true、そうでなければ false となる式は、

article.isFree()

と書くことができます。以上から、空欄 a は ☐ イ です。

空欄 b：空欄は、クラス Member のコンストラクタの

内容です。

```
Member(String name) {  
    ;  
}
```

クラス Member のコンストラクタは、引数で与えられた利用者名をもつインスタンスを生成します。

このクラスは抽象クラス User のサブクラスです。したがって、コンストラクタはスーパークラスのコンストラクタを呼び出し、

```
Member(String name) {  
    super(name);  
}
```

のように書くことができます。このことは、同じく User のサブクラスである Guest のコンストラクタが、[プログラム 3] で

```
Guest() {  
    super(" ゲスト ");  
}
```

と定義されていることからわかります。

以上から、空欄 b は **ウ** です。

空欄 c：空欄の前後は次のとおりです。

```
boolean testAndMark(Article article) {
    if (article.isFree()) {
        return true;
    }
    long today = today();
    if (browseDate  c  today) {
        return browsedArticle == article;
    }
    // 閲覧済みの有料記事と日付を記録
    browsedArticle = article;
    browseDate = today;
    return true;
}
```

クラス Member のメソッド testAndMark は、引数で与えられた記事が①無料記事のとき true を返します。また、有料記事のときは、②その記事が今日最初の有料記事のとき、または③その記事が閲覧済みの有料記事と同一のとき、true を返します。

①～③は、それぞれプログラムの次の部分に対応します。

①無料記事のとき

```
if (article.isFree()) {
    return true;
}
```

②今日最初の有料記事のとき

```
browsedArticle = article;
browseDate = today;
return true;
```

③閲覧済みの有料記事と同一のとき

```
if (browseDate  c  today) {
    return browsedArticle == article;
}
```

フィールド browseDate の値は、Long.MIN\_VALUE（システムが決める最小値）に初期化されており、上記②のケースが1度実行されると、以降は today に設定されます。条件式「browseDate

c  today」は、③のケースでは true にならなければならないので、

browseDate == today

とします。したがって、空欄 c には **ウ** の「==」が入ります。

空欄 d：空欄には、クラス PaidMember のメソッド testAndMark の戻り値が入ります。

```
boolean testAndMark(Article article) {
    return  d ;
}
```

クラス PaidMember のメソッド testAndMark は、すべての記事に対して true を返します。したがって、メソッドの定義は

```
boolean testAndMark(Article article) {
    return true;
}
```

とします。空欄 d は **キ** です。

空欄 e, f：空欄の前後は次のとおりです。

```
for ( e ) {
    for ( f ) {
        :
        System.out.printf(
            "%s: 記事「%s」%s%n",
            reader.getName(),
            article.getHeadline(), body);
    }
}
```

利用者名、記事見出し、記事本文（閲覧できない場合は「<閲覧不可>」と表示）を、2重ループを使って表示します。

図 1 の実行結果をみると、表示は利用者別・記事別に表示されているので、外側の for 文（空欄 e）で利用者の集合、内側の for 文（空欄 f）で記事の集合を指定すればよいと考えられます。

利用者のインスタンスは User 型の配列 readers にまとめられています。for 文の内側では、reader.getName() のように、変数 reader で利用者を参照しているので、外側の for 文は次のように指定します。



```
for (User reader : readers) {
```

以上から、空欄 e は **キ** です。

また、登録されている各記事はクラス Article に登録され、Article.getArticle(id) のように、String 型の識別子 id で参照されています。登録されている全記事の id の集合は静的メソッド getIds で取り出せるので、内側の for 文は次のように指定します。

```
for (String id : Article.getIds()) {
```

以上から、空欄 f は **カ** です。

## 設問 2 テスト用プログラムの変更

記事識別子 "0003" のフラグを false にすると、記事「アプリガイド」は有料記事になります。

この結果、ゲストは有料記事を閲覧できないので、記事「スマホ特集」とこの記事が「閲覧不可」になります。

また、登録会員は 1 日 1 本しか有料記事を閲覧できません。登録会員 A は、記事「スマホ特集」ですでに有料記事を閲覧しているので、この記事は「閲覧不可」になります。

有料会員はすべての有料記事を閲覧できるので、「閲覧

不可」になる記事はありません。

以上から、「閲覧不可」は全部で 3 回表示されます。

正解は **エ** です。

ゲスト：記事「PC入門」PC初心者…

ゲスト：記事「スマホ特集」＜閲覧不可＞

ゲスト：記事「アプリガイド」＜閲覧不可＞

登録会員A：記事「PC入門」PC初心者…

登録会員A：記事「スマホ特集」最新のスマホ…

登録会員A：記事「アプリガイド」＜閲覧不可＞

有料会員B：記事「PC入門」PC初心者…

有料会員B：記事「スマホ特集」最新のスマホ…

有料会員B：記事「アプリガイド」使えるアプリ…

### ○ 解答 ○

設問 1 a - **イ**, b - **ウ**, c - **ウ**,  
d - **キ**, e - **キ**, f - **カ**

設問 2 **エ**

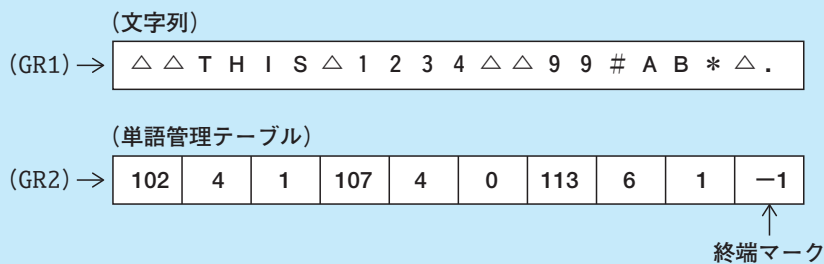
## 問 12 次のアセンブラプログラムの説明及びプログラムを読んで、設問 1～3 に答えよ。

### 〔プログラム 1 の説明〕

文字列中の単語を切り出して、単語管理テーブルを作成する副プログラム GETTKN である。

- (1) 文字列は英字、数字、空白文字の 0 文字以上の並びで、最後にピリオドが置かれる。単語は、1 文字以上の空白文字又はピリオドで区切られた英数字の並びである。英字には、アルファベットの他に、ピリオド以外の記号を含むものとする。文字列表記中の“△”は空白文字を示す。
- (2) 単語管理テーブルには、文字列中に現れる単語ごとに、3 語から成る要素を作成し、単語の先頭アドレス、単語の長さ、単語の種別の順に格納する。種別は、単語が数字だけから成るときは 0、英字を含むときは 1 とする。単語の切出しが終了したとき、単語管理テーブルの終端を示すマークとして -1 を格納する。
- (3) 主プログラムは、文字列の先頭アドレスを GR1 に、単語管理テーブルの先頭アドレスを GR2 に設定して、GETTKN を呼ぶ。
- (4) 副プログラム GETTKN から戻るとき、汎用レジスタ GR1～GR7 の内容は元に戻る。

副プログラム GETTKN の実行例を、図 1 に示す。



注記 文字列の先頭アドレスは100番地とし，数字は10進数表記とする。

図 1 副プログラム GETTKN の実行例

〔プログラム 1〕

```

GETTKN  START
        RPUSH
        LD      GR3,=-1          ; 単語種別の初期化
        LAD     GR1,-1,GR1
LP      LAD     GR1,1,GR1
        LD      GR4,0,GR1       ; 1文字を取り出す
        CPL     GR4,='.'
        JZE     FIN
        CPL     GR4,=' '
        JNZ     ALNUM
        CALL    SETTKN
        

|   |
|---|
| a |
|---|


ALNUM   LD      GR3,GR3          ; 単語の処理中?
        JPL     LP              ; 処理中の単語が英字を含む場合はLPへ
        JZE     ACHK            ; 処理中の単語が数字だけから成る場合はACHKへ
        LD      GR3,=0          ; 次の単語の処理開始
                                   ; 単語種別を“数字だけから成る”に設定
        LD      GR6,GR1         ; 先頭アドレスを退避
ACHK    CPL     GR4,='9'
        JPL     NEXT
        CPL     GR4,='0'
        JMI     NEXT
        JUMP    LP              ; 取り出した文字が数字の場合はLPへ
NEXT    LD      GR3,=1          ; 単語種別を“英字を含む”に設定
        JUMP    LP
FIN     CALL    SETTKN
        LD      GR5,=-1
        ST      GR5,0,GR2       ; 終端マークを格納
        RPOP
        RET
SETTKN  LD      GR3,GR3
        JMI     FIN2            ; 単語処理中でなければ何もしない
    
```

	ST	GR6, 0, GR2	; 単語の先頭アドレスを格納
	LD	GR5, GR1	
		<div style="border: 1px solid black; padding: 2px; display: inline-block;">b</div>	
	ST	GR5, 1, GR2	; 単語の長さを格納
	ST	GR3, 2, GR2	; 単語の種別を格納
	LD	GR3, -=1	; 単語種別を初期化（処理中状態を解除）
	LAD	GR2, 3, GR2	
FIN2	RET		
	END		

**設問 1** プログラム 1 中の   に入れる正しい答えを、解答群の中から選べ。

a に関する解答群

- |                            |      |              |                            |      |             |
|----------------------------|------|--------------|----------------------------|------|-------------|
| <input type="checkbox"/> ア | JUMP | ACHK         | <input type="checkbox"/> イ | JUMP | LP          |
| <input type="checkbox"/> ウ | LAD  | GR1, -1, GR1 | <input type="checkbox"/> エ | LAD  | GR1, 1, GR1 |
| <input type="checkbox"/> オ | LD   | GR4, 0, GR1  | <input type="checkbox"/> カ | LD   | GR4, 1, GR1 |

b に関する解答群

- |                            |      |          |                            |      |             |
|----------------------------|------|----------|----------------------------|------|-------------|
| <input type="checkbox"/> ア | ADDL | GR5, GR6 | <input type="checkbox"/> イ | LAD  | GR3, 1, GR3 |
| <input type="checkbox"/> ウ | SLL  | GR3, 1   | <input type="checkbox"/> エ | SLL  | GR5, 1      |
| <input type="checkbox"/> オ | SRL  | GR5, 1   | <input type="checkbox"/> カ | SUBL | GR5, GR6    |

**設問 2** 次の文字列が与えられ、プログラム 1 のラベル NEXT が付いた命令を 2 度目に実行した直後に GR4 に設定されている文字として、正しい答えを、解答群の中から選べ。

(GR1) → △ 1 A 2 B △ C 3 .

解答群

- |                            |   |                            |   |                            |   |
|----------------------------|---|----------------------------|---|----------------------------|---|
| <input type="checkbox"/> ア | 1 | <input type="checkbox"/> イ | 2 | <input type="checkbox"/> ウ | 3 |
| <input type="checkbox"/> エ | A | <input type="checkbox"/> オ | B | <input type="checkbox"/> カ | C |

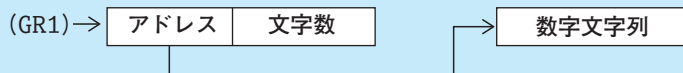
**設問 3** GETTKN を使用して、指示に従い 2 数の和、差、積のいずれかを求める副プログラム CALC を作成した。プログラム 2 中の   に入れる正しい答えを、解答群の中から選べ。

- (1) 主プログラムは、文字列の先頭アドレスを GR1 に設定して、CALC を呼ぶ。CALC は演算結果を GR0 に設定して呼出し元に戻る。文字列の形式を図 2 に示す。

(GR1)→ 数字文字列1△演算子△数字文字列2△.

図2 文字列の形式

- (2) 数字文字列は0～65535の整数(符号なしの数字文字列)で1文字以上の文字列とし、演算子は加算(+), 減算(-), 乗算(\*)を表す記号1文字とする。演算はそれぞれ論理加算, 論理減算, 整数の乗算として実行し, 桁あふれは発生しないものとする。
- (3) CALCは, GETTKNで切り出した数字だけから成る単語を2進数に変換するために, 別に用意された副プログラム DTOB を呼ぶ。
- (4) DTOBは, 数字文字列格納領域の先頭アドレスと文字列の長さが順に格納された2語から成る領域の先頭アドレスが GR1 に設定されて, 呼び出される。DTOBは数字文字列を2進数に変換し, GR0 に設定して呼出し元に戻る。



- (5) 副プログラム CALC, DTOB から戻るとき, 汎用レジスタ GR1～GR7の内容は元に戻る。

#### 〔プログラム2〕

```
CALC    START
        RPUSH
        LAD    GR2, CTBL
        CALL   GETTKN
        

c


        CALL   DTOB                ; 数字文字列1を2進数に変換して
        LD     GR4, GR0             ; GR4に設定
        LD     GR1, 3, GR2
        LD     GR3, 0, GR1         ; GR3 ← 演算子の文字コード
        SUBL   GR3, '=' * '
        LAD    GR1, 6, GR2
        CALL   DTOB                ; 数字文字列2を2進数に変換して
        LD     GR5, GR0            ; GR5に設定
        

d


        JUMP   0, GR3              ; 演算子で指定された処理にジャンプ
MULT    LD     GR0, =0             ; 乗算
        LD     GR5, GR5
LP      JZE    FIN
        LD     GR3, GR5
        AND    GR3, =#0001         ; 乗数の最下位ビットのチェック
        JZE    NEXT
        ADDL   GR0, GR4
NEXT    SLL    GR4, 1              ; 被乗数を1ビット左論理シフト
        

e


        JUMP   LP
```

PLUS	ADDL	GR4, GR5	; 加算
	LD	GR0, GR4	
	JUMP	FIN	
MINUS	SUBL	GR4, GR5	; 減算
	LD	GR0, GR4	
FIN	RPOP		
	RET		
CTBL	DS	10	; GETTKN用単語管理テーブル
LTBL	DC	MULT	; 演算の分岐先アドレステーブル
	DC	PLUS	
	DS	1	; ダミー
	DC	MINUS	
	END		

#### c に関する解答群

- |          |     |             |          |    |             |
|----------|-----|-------------|----------|----|-------------|
| <b>ア</b> | LAD | GR1, 3, GR2 | <b>イ</b> | LD | GR1, 0, GR2 |
| <b>ウ</b> | LD  | GR1, CTBL   | <b>エ</b> | LD | GR1, GR2    |
| <b>オ</b> | LD  | GR2, CTBL   | <b>カ</b> | LD | GR2, GR1    |

#### d に関する解答群

- |          |     |           |          |     |                |
|----------|-----|-----------|----------|-----|----------------|
| <b>ア</b> | LAD | GR3, LTBL | <b>イ</b> | LAD | GR3, LTBL, GR3 |
| <b>ウ</b> | LD  | GR3, LTBL | <b>エ</b> | LD  | GR3, LTBL, GR3 |
| <b>オ</b> | SLL | GR3, 1    |          |     |                |

#### e に関する解答群

- |          |      |          |          |      |          |
|----------|------|----------|----------|------|----------|
| <b>ア</b> | ADDL | GR0, GR5 | <b>イ</b> | ADDL | GR5, GR4 |
| <b>ウ</b> | LD   | GR5, GR4 | <b>エ</b> | SLL  | GR5, 1   |
| <b>オ</b> | SRL  | GR5, 1   |          |      |          |

## 問 12 午後のカギ

単語の切出しや、数式の解析を行うアセンブラプログラムの問題です。空欄を埋める問題では、プログラムの処理の流れをトレースしながら、正しく動作するために欠けている処理を考えます。

設問 1 空欄 b は、単語管理テーブルに格納する単語の文字数を計算する命令です。

設問 2 ラベル NEXT の命令が、どの文字を読み込んだときに実行されるかを考えます。

設問 3 空欄 c は、GETTKN の結果を DT0B に渡すために必要な処理、空欄 d は、分岐先の実効アドレスを設定します。空欄 e は、2 進数同士の乗算のアルゴリズムの一部です。

### 設問 1 プログラムの完成

空欄 a：空欄の前後は次のとおりです。

LD	GR3, -1	…①
LAD	GR1, -1, GR1	…②

LP	LAD	GR1, 1, GR1	…③
	LD	GR4, 0, GR1	…④
	CPL	GR4, = ' . '	
	JZE	FIN	…⑤
	CPL	GR4, = ' '	
	JNZ	ALNUM	…⑥
	CALL	SETTKN	…⑦
		a	
ALNUM	LD	GR3, GR3	
	JPL	LP	…⑧

- ①単語の種別を－1に初期化。
- ②文字列のポインタを1減らす（③で文字列の先頭に戻すため）。
- ③文字列のポインタを1増やす。
- ④GR4に1文字読み込む。
- ⑤ピリオドならラベルFINへ。
- ⑥空白文字でなければ⑧へ。
- ⑦空白文字ならSETTKNを呼び出す。
- ⑧単語種別が1なら③に戻る。

GR1には文字列中の文字を指すポインタ、GR4にはポインタが指すアドレスから読み込んだ文字、GR3には単語の種別（初期値は－1）が格納されます。

読み込んだ文字がピリオドなら文字列の末尾なので、ラベルFINにジャンプします。読み込んだ文字が空白文字のときは単語の区切りを意味するので、ラベルSETTKNを呼び出します。

ラベルSETTKNは、単語管理テーブルに単語情報を書き込み、空欄aに戻ります（先頭の空白文字や連続する空白文字の場合は特に何もせず、空欄aに戻ります）。

1単語分の処理が終わったら、ラベルLPに戻って次の単語の処理をはじめます。したがって空欄aには、

JUMP LP

が入ります。空欄aはイです。

空欄b：空欄の前後は次のとおりです。

SETTKN	LD	GR3, GR3	
	JMI	FIN2	…①
	ST	GR6, 0, GR2	…②
	LD	GR5, GR1	
		b	

	ST	GR5, 1, GR2	…③
	ST	GR3, 2, GR2	…④
	LD	GR3, = -1	…⑤
	LAD	GR2, 3, GR2	…⑥
FIN2	RET		

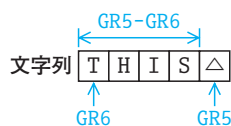
- ①単語種別が－1のときはFIN2へ。
- ②単語の先頭アドレスを単語管理テーブルに格納。
- ③単語の長さを単語管理テーブルに格納。
- ④単語の種別を単語管理テーブルに格納。
- ⑤単語の種別を－1に初期化。
- ⑥単語管理テーブルを次のレコードに設定。

ラベルSETTKNは、単語管理テーブルに、単語の先頭アドレス、単語の長さ、単語の種別を格納します。

ST GR6, 0, GR2 ←単語の先頭アドレス  
ST GR5, 1, GR2 ←単語の長さ  
ST GR3, 2, GR2 ←単語の種別

このうち、GR5に格納されている単語の長さは、「現在のポインタのアドレス－単語の先頭アドレス」で求めます。この処理を行っているのが次の部分です。

LD GR5, GR1 ←ポインタアドレスをGR5に複写  
SUBL GR5, GR6 ←GR5－GR6を計算



以上から、空欄bにはカが入ります。

## 設問2 プログラムのトレース

ラベルNEXTの前後のプログラムは次のとおりです。

ALNUM	LD	GR3, GR3	
	JPL	LP	…①
	JZE	ACHK	…②
	LD	GR3, = 0	…③
	LD	GR6, GR1	…④
ACHK	CPL	GR4, = ' 9 '	
	JPL	NEXT	…⑤
	CPL	GR4, = ' 0 '	
	JMI	NEXT	…⑥
	JUMP	LP	
NEXT	LD	GR3, = 1	…⑦
	JUMP	LP	

- ①単語種別が 1 なら LP へ戻る。
- ②単語種別が 0 なら ACHK へ。
- ③単語種別が -1 なら単語の先頭文字とみなし、単語種別を 0 に設定。
- ④単語の先頭アドレスを GR6 に格納。
- ⑤ '9' より後の文字か、
- ⑥ '0' より後の文字の場合は
- ⑦単語種別を 1 に設定。

ラベル NEXT の命令は、処理中の単語の種別を 1 に設定します。この命令が実行されるのは、以下のどちらかのケースに限ります。

ケース 1：処理中の単語が先頭から連続して数字（文字種別 0 の状態）で、GR4 に読み込んだ文字が英字だった場合→②でラベル ACHK にジャンプし、⑤または⑥でラベル NEXT にジャンプする。

ケース 2：GR4 に読み込んだ文字が単語の先頭文字（文字種別 -1 の状態）で、かつ、英字だった場合→③④を経てラベル ACHK の命令が実行され、⑤または⑥でラベル NEXT にジャンプする。

したがって、与えられた文字列が次のとき、ラベル NEXT の命令が実行されるのは、GR4 に読み込んだ文字が次のときです。

△ 1 A 2 B △ C 3 .  
↑                      ↑  
ケース1                      ケース2

以上から、NEXT の命令を 2 度目に実行した直後、GR4 に設定されている文字は 'C'（力）となります。

### 設問 3 副プログラム CALC

空欄 c：空欄の前後は次のとおりです。

```
CALL GETTKN
  c
CALL DTOB
```

GETTKN の実行を終えると、次のような単語管理テーブルの先頭アドレスが GR2 に格納されます。

単語管理テーブル  
 (GR2) → アドレス 文字数 0 ... → 数字文字列

DTOB は、上記のアドレスと文字数の部分が設定さ

れた領域の先頭アドレスを GR1 に設定して呼び出します。したがって、空欄 c で、GR2 の内容を GR1 に設定する処理が必要になります。

LD GR1, GR2

以上から、空欄 c には エ が入ります。

空欄 d：空欄の前後は次のとおりです。

```
LD GR4, GR0      ...①
LD GR1, 3, GR2
LD GR3, 0, GR1    ...②
SUBL GR3, '*'      ...③
LAD GR1, 6, GR2
CALL DTOB
LD GR5, GR0        ...④
d
JUMP 0, GR3        ...⑤
```

- ① 1 つ目の数字を GR4 に設定。
- ② 演算子の文字コードを GR3 に設定。
- ③ 乗算の場合 0、加算の場合 1、減算の場合 3 を GR3 に設定。
- ④ 2 つ目の数字を GR5 に設定。
- ⑤ GR3 に設定された分岐先アドレスにジャンプ。

⑤で、GR3 に設定されている分岐先アドレスにジャンプしています。分岐先は、演算の種類に応じて、乗算の場合はラベル MULT、加算の場合はラベル PLUS、減算の場合はラベル MINUS です。これらのアドレスは、それぞれラベル LTBL に次のように格納されています。

(LTBL) → MULT PLUS ダミー MINUS  
↑                      ↑                      ↑  
+0                      +1                      +3

LTBL に加える値は、③で GR3 に設定されます。したがって空欄 d には、アドレス (LTBL) + (GR3) に格納されているアドレスを、GR3 に設定する処理が入ります。

LD GR3, LTBL, GR3

以上から、空欄 d は エ です。

なお、命令「LAD GR3, LTBL, GR3」と紛らわしいですが、LAD 命令では、アドレス (LTBL) + (GR3) に格納されている値ではなく、LTBL + GR3 のアドレス自体が GR3 に格納されてしまいます。



空欄 e：空欄の前後は次のとおりです。

MULT	LD	GR0, =0	…①
	LD	GR5, GR5	
LP	JZE	FIN	…②
	LD	GR3, GR5	…③
	AND	GR3, =#0001	…④
	JZE	NEXT	
	ADDL	GR0, GR4	…⑤
NEXT	SLL	GR4, 1	…⑥
		<div>e</div>	…⑦
	JUMP	LP	…⑧

- ①乗算結果を格納する GR0 を 0 に初期化。
- ② GR5 が 0 なら終了。
- ③ GR5 の内容を GR3 にコピー。
- ④ GR3 の下位 1 ビットを取り出す。
- ⑤ GR3 が 1 なら、GR4 を GR0 に加算。
- ⑥ GR4 を 1 ビット左シフト。
- ⑦ 

e
- ⑧②に戻る。

GR4×GR5 を計算し、結果を GR0 に格納する処理です。2 進数の乗算は、次のような手順で計算できます。

- (1) 乗数 (GR5) の下位 1 ビットが 1 なら、結果 (GR0) に被乗数 (GR4) を加える

- (2) 被乗数を 1 ビット左にシフトする
- (3) 乗数を 1 ビット右にシフトする
- (4) 乗数が 0 なら終了、さもなければ (1) に戻る

例として、1101×1010 を計算する場合を次に示します。GR4、GR5、GR0 の内容は、(1) の処理を実行するたびに次のように変化し、最終的には GR0 に 1101×1010 の計算結果が格納されます。

GR4	GR5	GR0
1101	1010	0
11010	101	11010
110100	10	11010
1101000	1	10000010

以上から、空欄 e には「乗数を 1 ビット右にシフトする」処理が入ります。

SRL GR5, 1

空欄 e は **オ** です。

解答

- 設問 1 a - **イ**, b - **カ**
- 設問 2 **カ**
- 設問 3 c - **エ**, d - **エ**, e - **オ**

問 13 次の表計算のワークシート及びマクロの説明を読んで、設問 1、2 に答えよ。

〔表計算の説明〕

F 社では、所有する顧客情報のデータを分析して営業活動に役立てている。このたび、情報漏えいなどの事故に備えるために、必要に応じて匿名性の高いデータに変換してから、分析を担当する部署に渡すことになった。そこで、顧客データに対して匿名化処理を行うプログラムを表計算ソフトで作成した。

〔ワークシート：顧客リスト〕

分析対象の顧客は 5,000 人である。各顧客に対して固有の ID、氏名、郵便番号、住所、電話番号、年齢、職業コードの情報を所有している。顧客情報を格納したワークシート“顧客リスト”の例を図 1 に示す。ここで、郵便番号は 7 桁の数値として取り扱う。

	A	B	C	D	E	F	G
1	ID	氏名	郵便番号	住所	電話番号	年齢	職業コード
2	0001	滝本卓也	1070062	東京都港区南青山…	03xxxxxxxx	66	111
3	0002	中野浩二	1690051	東京都新宿区西早稲田…	03xxxxxxxx	59	132
4	0003	下山健	1110042	東京都台東区寿…	03xxxxxxxx	37	214
5	0004	飯山由美	1830003	東京都府中市朝日町…	042xxxxxxxx	19	312
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
5000	4999	山本礼子	5300042	大阪府大阪市北区天満橋…	06xxxxxxxx	29	112
5001	5000	田中陽子	2400012	神奈川県横浜市保土ヶ谷…	045xxxxxxxx	36	321

図1 ワークシート“顧客リスト”の例

〔ワークシート：匿名化顧客リスト〕

ワークシート“顧客リスト”を基に、匿名化処理を施したワークシート“匿名化顧客リスト”を作成した。ワークシート“匿名化顧客リスト”の例を図2に示す。

なお、列H、列Iは、後述のマクロGenerateListで使用する。

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	ID	氏名	郵便番号	住所	電話番号	年齢	職業コード	マーキング	評価値		表示年齢	職業コード変換	多重度	
2	0001	*	10	*	*	60	100				0	100	*	3
3	0002	*	16	*	*	50	100				20	110		
4	0003	*	11	*	*	35	210				25	120	*	
5	0004	*	18	*	*	0	312				30	200		
6	0005	*	27	*	*	50	100				35	210	*	
7	0006	*	20	*	*	25	312				40	220		
8	0007	*	20	*	*	30	230				50	230	*	
9	0008	*	25	*	*	60	100				60	240		
10	0009	*	26	*	*	40	400				1000	300		
11	0010	*	22	*	*	30	222					310		
12	0011	*	22	*	*	35	225					320	*	
5000	4999	*	53	*	*	25	100							
5001	5000	*	24	*	*	35	320							

図2 ワークシート“匿名化顧客リスト”の例

**設問1** ワークシート“匿名化顧客リスト”に関する次の記述中の□に入れる正しい答えを、解答群の中から選べ。ここで、c1とc2に入れる答えは、cに関する解答群の中から組合せとして正しいものを選ぶものとする。

- (1) 氏名、住所、電話番号の情報は完全に秘匿するために、“\*”にする。そこでセルB2、D2、

E2に“\*”を入力し、それぞれセル B3～B5001, D3～D5001, E3～E5001 に複写する。

- (2) 郵便番号は、7桁の郵便番号のうち上位2桁だけの値に置き換えることで匿名化する。そこで、次の式をセル C2 に入力し、セル C3～C5001 に複写する。

a

- (3) セル F2～F5001 には、図3に示す各年齢区分の最小値である表示年齢を表示することで匿名化する。各年齢区分の最小値である表示年齢を、セル K2～K9 に昇順に入力する。また、セル K10 には、終端を示す 1000 を入力する。

年齢区分	表示年齢
0～19歳	0
20～24歳	20
25～29歳	25
30～34歳	30
35～39歳	35
40～49歳	40
50～59歳	50
60歳以上	60

図3 年齢区分

例えば、顧客の年齢が23歳の場合、“20”と表示する。そこで、次の式をセル F2 に入力し、セル F3～F5001 に複写する。

b

- (4) 職業コードは図4に示す項目から成る3桁の数値である。各桁は、1～9の整数値である。

大分類 1桁	中分類 1桁	小分類 1桁
-----------	-----------	-----------

図4 職業コードの様式

セル G2～G5001 には、必要に応じて大分類化するか中分類化することで匿名化した職業コードを表示する。大分類化とは、中分類、小分類の値をいずれも0に置き換えた数値に変換することであり、中分類化とは、小分類の値を0に置き換えた数値に変換することである。大分類化、中分類化された3桁の数値をそれぞれ大分類コード、中分類コードと呼ぶ。

セル L2～M41 には、職業コードの大分類化、中分類化に関する情報が入力されている。存在し得る大分類コード、中分類コードは合わせて40種類であり、セル L2～L41 に昇順で格納されている。次の規則によって、職業コードを変換する。

- ① 大分類コードが格納されているセルの右側の対応するセルに“\*”が格納されている場合、その大分類に属する職業コードは全て大分類コードに変換する。例えば、セル L2 には大分

類コード 100 が格納されていて、その右隣のセル M2 は “\*” なので、100 番台の全ての職業コードは 100 に変換する。

- ② ①に該当しない場合であって、中分類コードが格納されているセルの右側の対応するセルに “\*” が格納されている場合、その中分類に属する職業コードは全て中分類コードに変換する。例えば、セル L6 には中分類コード 210 が格納されていて、右隣のセル M6 は “\*” なので、210 番台の全ての職業コードは 210 に変換する。
- ③ ①、②のいずれも該当しない場合、職業コードは変換しない。
- そこで、次の式をセル G2 に入力し、セル G3 ～ G5001 に複写する。

IF (垂直照合 (  , L\$2:M\$41, 2, 0) = ' \* ',  ,  
IF (垂直照合 (  , L\$2:M\$41, 2, 0) = ' \* ',  ,  
顧客リスト!G2))

#### a に関する解答群

- ア** 剰余 ( $10^5$ , 顧客リスト!C2)
- イ** 剰余 (顧客リスト!C2,  $10^5$ )
- ウ** 整数部 ((顧客リスト!C2 /  $10^5$ )
- エ** 整数部 (顧客リスト!C2 /  $10^5$ )
- オ** 整数部 (顧客リスト!C2 /  $10^5$ )

#### b に関する解答群

- ア** 垂直照合 (顧客リスト!F2, K\$2:K\$10, 1, 0)
- イ** 垂直照合 (顧客リスト!F2, K\$2:K\$10, 1, 1)
- ウ** 垂直照合 (顧客リスト!F2 + 1, K\$2:K\$10, 1, 0)
- エ** 垂直照合 (顧客リスト!F2 + 1, K\$2:K\$10, 1, 1)
- オ** 表引き (K\$2:K\$10, 照合一致 (顧客リスト!F2, K\$2:K\$10, - 1), 1)
- カ** 表引き (K\$2:K\$10, 照合一致 (顧客リスト!F2 + 1, K\$2:K\$10, 1), 1)

#### c に関する解答群

	c1	c2
<b>ア</b>	切捨て (顧客リスト!G2, - 1)	顧客リスト!G2
<b>イ</b>	切捨て (顧客リスト!G2, - 2)	切捨て (顧客リスト!G2, - 1)
<b>ウ</b>	切捨て (顧客リスト!G2, - 2)	顧客リスト!G2
<b>エ</b>	顧客リスト!G2	切捨て (顧客リスト!G2, - 1)
<b>オ</b>	顧客リスト!G2	切捨て (顧客リスト!G2, - 2)

#### 【ワークシート：提供リスト】

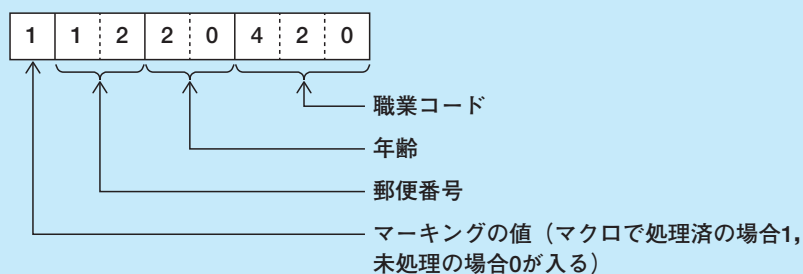
匿名化された後の郵便番号、年齢、職業コードが全て同一となる顧客の数（多重度と呼ぶ）のリストをデータ分析の担当部署に渡すために、**図 5** に示すワークシート “提供リスト” を作成する。

	A	B	C	D
1	郵便番号	年齢	職業コード	多重度
2	00	20	230	4
3	00	60	500	4
4	01	20	241	5
⋮	⋮	⋮	⋮	⋮
206	98	60	500	4
207	99	60	500	3

注記 郵便番号は2桁で表示している。

図5 ワークシート“提供リスト”

- (1) ワークシート“匿名化顧客リスト”にマクロ GenerateList を実装する。
- (2) マクロ GenerateList は、ワークシート“提供リスト”の列 A～D に、ワークシート“匿名化顧客リスト”を郵便番号、年齢、職業コードの優先順で昇順に整列し、多重度とともに表示する。
- (3) ワークシート“匿名化顧客リスト”のセル H2～H5001 は、マクロ GenerateList の実行中に、どの顧客の情報が処理済みであるかを示すのに用いる。処理済みの顧客の場合は1を、そうでない場合は0を表示する。
- (4) (2) の処理を行う際、次に処理する顧客を決定するために、ワークシート“匿名化顧客リスト”中の郵便番号、年齢、職業コードから成る次のような8桁の数値（以下、評価値という）を定義する。



この評価値が最小の顧客を処理対象とし、多重度を順次求めていく。

そこで、次の式をワークシート“匿名化顧客リスト”のセル I2 に入力し、セル I3～I5001 に複写する。

$$H2 * 10000000 + C2 * 100000 + F2 * 1000 + G2$$

- (5) ワークシート“匿名化顧客リスト”のセル N2 には、表示する多重度の下限値が入力されている。ここで、表示する多重度の下限値は2以上の整数とする。多重度がこの値以上の郵便番号、年齢、職業コードの組を、多重度とともにワークシート“提供リスト”に表示する。

**設問 2** マクロGenerateListをワークシート“匿名化顧客リスト”に格納した。マクロGenerateList中の  に入れる正しい答えを、解答群の中から選べ。

[マクロ：GenerateList]

```

○マクロ： GenerateList
○数値型： numCustomer, minID, previousValue, minMultiplicity,
          I, J, K
・numCustomer ← 5000
・previousValue ← 0
・minMultiplicity ← N2
・相対(提供リスト!A1, 0, 0) ← '郵便番号'
・相対(提供リスト!A1, 0, 1) ← '年齢'
・相対(提供リスト!A1, 0, 2) ← '職業コード'
・相対(提供リスト!A1, 0, 3) ← '多重度'

■ I: 1, I ≤ numCustomer, 1
  ↓
  ・相対(H1, I, 0) ← 0
  ■
  ・J ← 1
  ・K ← 0
  ■ I: 1, I ≤ numCustomer, 1
    ↓
    ・minID ←  d
    ・相対(H1, minID, 0) ← 1
    ▲相対(I1, minID, 0) ≠ previousValue
      ▲K ≥ minMultiplicity
        ↓
        ・ e
        ↓
        ・相対(提供リスト!A1, J, 0) ← 相対(A1, minID, 2)
        ・相対(提供リスト!A1, J, 1) ← 相対(A1, minID, 5)
        ・相対(提供リスト!A1, J, 2) ← 相対(A1, minID, 6)
        ・K ← 1
        ・相対(提供リスト!A1, J, 3) ← K
        ・previousValue ← 相対(I1, minID, 0)
        ↓
        ・ f
        ・相対(提供リスト!A1, J, 3) ← K
        ↓
        ■
      ▲相対(提供リスト!A1, J, 3) < minMultiplicity
        ↓
        ・相対(提供リスト!A1, J, 0) ← null
        ・相対(提供リスト!A1, J, 1) ← null
        ・相対(提供リスト!A1, J, 2) ← null
        ・相対(提供リスト!A1, J, 3) ← null
        ↓
        ▲

```

### d に関する解答群

- ア** 条件付個数(I2:I5001, > 相対(I1, I, 0))
- イ** 条件付個数(I2:I5001, < 相対(I1, I, 0))
- ウ** 照合一致(最小(I2:I5001), I2:I5001, 0)
- エ** 照合一致(最小(I2:I5001), I2:I5001, 1)
- オ** 照合一致(最大(I2:I5001), I2:I5001, 0)
- カ** 照合一致(最大(I2:I5001), I2:I5001, 1)
- キ** 相対(I1, I, 0)
- ク** 相対(I1, J, 0)

### e, f に関する解答群

- |                               |                               |                               |
|-------------------------------|-------------------------------|-------------------------------|
| <b>ア</b> $I \leftarrow I + 1$ | <b>イ</b> $I \leftarrow I + J$ | <b>ウ</b> $I \leftarrow I + K$ |
| <b>エ</b> $J \leftarrow I$     | <b>オ</b> $J \leftarrow J + 1$ | <b>カ</b> $J \leftarrow J + K$ |
| <b>キ</b> $J \leftarrow K + 1$ | <b>ク</b> $K \leftarrow J$     | <b>ケ</b> $K \leftarrow K + 1$ |
| <b>コ</b> $K \leftarrow K + J$ |                               |                               |

## 問 13 午後のカギ

表計算問題では、大きく計算式の組立てとマクロプログラムの作成の2つが出題されますが、本問は特にマクロプログラムに複雑な仕事をさせています。問題文の説明をよく読んで、プログラムの内容を把握しましょう。

**設問 1** 垂直照合関数の引数にある、「検索の指定」というオプションの使い方に注意します。

**設問 2** 多重度が下限値より小さい場合は、提供リストに出力しないという仕様がポイントです。

### 設問 1 ワークシート“匿名化顧客リスト”

**空欄 a:** 7桁の郵便番号から、上位2桁だけを取り出す式です。

郵便番号は、ワークシート“顧客リスト”に、9999999のような7桁の数値で入力されています。これを100,000で割ると、結果は99.99999のような数値になります。この数値の小数点以下を切り捨てれば、上位2桁の数値が取り出せます。小数点以下を切り捨てるには整数部関数が利用できます。

以上から、C2入力する式は、

**整数部 (顧客リスト!C2/100000)**

です。解答群の**エ**は、「100000」を「 $10^5$ 」( $10$ の5乗)に置き換え、

**整数部 (顧客リスト!C2/10^5)**

となっています。これが正解です。

**空欄 b:** セルF2に、年齢をもとに検索した表示年齢を表示する式を入力します。表示年齢はK2~K10に入力されています。このような計算式には、次のような垂直照合関数を使います。

**垂直照合 (式, セル範囲, 列位置, 検索の指定)**

① ② ③ ④

①式: 顧客リストのセルF2を指定します。

②セル範囲: 表示年齢が入力されているセルK2~K10を指定します。ただし、式は行方向に複写するので、セル番地がずれないように、K\$2~K\$10とします。

③列位置: セルK2~K10は1列しかないなので、1を指定します。



④検索の指定：ここでは、式の値と一致する値ではなく、式の値以下の最大値を表示年齢とするので、1を指定します。

以上から、セル F2 に入力する式は、

垂直照合 (顧客リスト !F2, K\$2:K\$10, 1, 1)

となります。以上から、空欄 b は **イ** です。

なお、解答群の **オ** や **カ** のように表引き関数と照合一致関数を組み合わせる場合は、

表引き (K\$2:K\$10, 照合一致 (顧客リスト !F2, K\$2:K\$10, 1), 1)

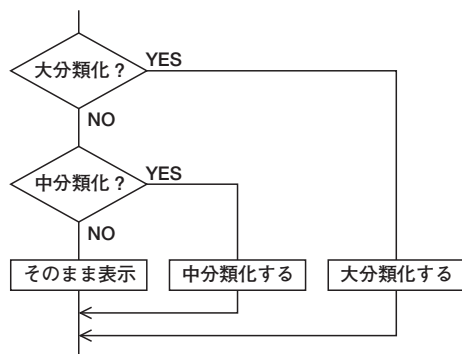
とすれば同じ結果を得られます。

検索の指定に 1 を指定するときは、セル範囲が昇順に整列されてなければならないことに注意！



空欄 c：セル G2 には、必要に応じて大分類化または中分類化した職業コードを表示する式を入力します。

この式は IF 関数を入れ子に使う。①大分類化が必要な場合は大分類化した職業コードを表示し、さもなければ、②中分類化が必要な場合は中分類化した職業コードを表示します。どちらでもないときは、職業コードをそのまま表示します。



IF (大分類化?, 大分類化する,  
IF (中分類化, 中分類化する,  
そのまま表示))

c1 には、ワークシート“顧客リスト”の職業コードを大分類化した数値が入ります。大分類化では、3桁の数値の下2桁を0に置き換えます。切捨て関数の切捨て位置に負の数を指定すると、整数部の桁を切捨てできるので、

切捨て (顧客リスト !G2, -2)

のようにすれば、下2桁が0に置き換わりします。

次に c2 には、ワークシート“顧客リスト”の職業コードを中分類化した数値が入ります。中分類化では、3桁の数値の下1桁を0に置き換えるので、切捨て関数を使い、

切捨て (顧客リスト !G2, -1)

のようにします。

完成した式は次のようになります。

IF (垂直照合 (切捨て (顧客リスト !G2, -2), L\$2: M\$41, 2, 0) = ' \* ', 切捨て (顧客リスト !G2, -2), IF (垂直照合 (切捨て (顧客リスト !G2, -1), L\$2: M\$41, 2, 0) = ' \* ', 切捨て (顧客リスト !G2, -1), 顧客リスト !G2))

以上から、空欄 c は **イ** です。

## 設問2 マクロの完成

空欄 d：空欄の前後は次のとおりです。

```

I: 1, I ≤ numCustomer, 1
  • minID ← d
  • 相対(H1, minID, 0) ← 1
  ▲相対(I1, minID, 0) ≠ previousValue
  
```

変数 minID には、処理対象となる顧客のワークシートの行位置を代入します。処理対象となる顧客は、セル I2 ~ I5001 に入力されている評価値が最小である顧客です。そこでまず、最小関数を使って入力されているうちの最小の評価値を求め、この値が入力されている行の位置を、照合一致関数で求めます。照合一致関数の書式は次のようになります。

照合一致 (式, セル範囲, 検索の指定)

① ② ③

①式：最小の評価値を求めるので、最小関数を使い、最小 (I2: I5001) とします。

②セル範囲：I2 ~ I5001 から検索します。

③検索の指定：最小の評価値と一致する行位置を求めるので、0を指定します。