

# 情 報 I

(全 問 必 答)

第 1 問 次の問い(問 1 ～ 4)に答えよ。(配点 20)

問 1 次の問い(a・b)に答えよ。

a 次の文章中の空欄  ・  に入れるのに最も適当なものを、後の解答群のうちから一つずつ選べ。

コンピュータの記憶装置は主記憶装置と補助記憶装置に大別される。一般に、主記憶装置に比べ、補助記憶装置はデータを読み書きする速度が  であり、  のために用いられる。

の解答群

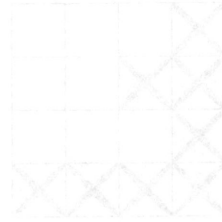
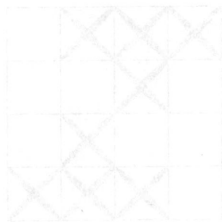
|      |       |      |
|------|-------|------|
| ① 低速 | ② 同程度 | ③ 高速 |
|------|-------|------|

の解答群

|                     |
|---------------------|
| ① 容量が小さく、データの短期的な保存 |
| ② 容量が小さく、データの長期的な保存 |
| ③ 容量が大きく、データの短期的な保存 |
| ④ 容量が大きく、データの長期的な保存 |

b 情報セキュリティに関する記述として最も適当なものを、次の①～④のうちから一つ選べ。 ウ

- ① 犯罪につながる情報が公開されている Web サイトに日本国内からアクセスすると、不正アクセス禁止法違反になる。
- ② ネットワークの通信速度を高速にすることにより、マルウェアによる被害を最小限に抑えることができる。
- ③ ソーシャルエンジニアリングによる被害を防ぐには、コンピュータの主記憶装置(メモリ)を増設することが有効である。
- ④ 保有するデータをバックアップしておくことで、情報セキュリティの要素である「可用性」を高めることができる。
- ⑤ ファイアウォールを利用することによって、無線 LAN での通信内容が盗聴されることを防ぐことができる。



問 2 次の文章を読み、空欄 **エ** ~ **ケ** に当てはまる数字または文字をマークせよ。なお、当てはまる文字が A ~ F の場合には、**a** ~ **f** をマークすること。

M さんは家庭科の授業で、クロスステッチの刺繍<sup>ししゅう</sup>を作成することになった。クロスステッチは、布上の格子の 1 マスに対して糸を「×」の形に縫う刺繍の方法である。刺繍の模様は、図 1 に示すような「図案」で表現される。

M さんは図案の作成にコンピュータを利用しようと考え、縫うマスを 1、縫わないマスを 0 で表すことにした。図 1 の一番上の 4 マス(1 行)は、2 進法で表現すると 4 桁の **1000**<sup>(2)</sup> になり、これを 16 進法で表現すると 1 桁の **8**<sup>(16)</sup> になる。図 1 の 4 行分を続けて書くと、図 1 の図案は **8C8F**<sup>(16)</sup> という 16 進法での 4 桁で表現できる。この方式に基づいて、図 2 の図案を 16 進法で表現すると、**エ** **オ** **27**<sup>(16)</sup> となる。

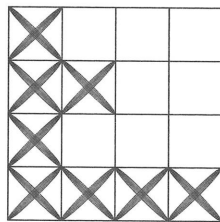


図 1 クロスステッチの図案の例(1)

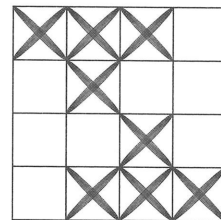


図 2 クロスステッチの図案の例(2)

Mさんは、図案の種類を増やすために、 $4 \times 4$ マスの一つの図案に対する次のような変換方法を考えた。

【変換方法】

変換A：図案の各行を2進法で表して桁を逆順にする。

例) 図1の上から1行目と3行目の4マスは、いずれも  $1000_{(2)}$  から  $0001_{(2)}$  になる。

変換B：図案の全体を16進法で表して桁を逆順にする。

例) 図1の図案全体は、 $8C8F_{(16)}$  から  $F8C8_{(16)}$  になる。

変換C：変換Aを行った後、その結果に対して変換Bを行う。

続いて、小さな図案を組み合わせえてより大きな図案を作るため、図3に示すように  $4 \times 4$  マスの図案を4枚配置して  $8 \times 8$  マスの図案を作ることを考えた。図3中の、変換A、変換B、変換Cはいずれも元の図案に対して変換を行った結果とする。 $35AD_{(16)}$  で表される図案を元の図案として、図3の方法で  $8 \times 8$  マスの図案に拡張したとき、その中央の  $4 \times 4$  マス(図3の網掛け部分)の図案を16進法で表現すると、**カ****キ****ク****ケ**<sub>(16)</sub> となる。

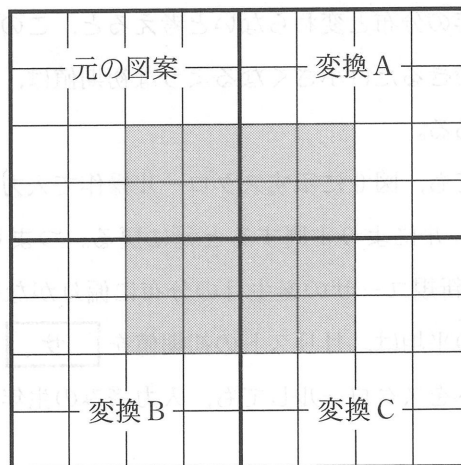


図3  $4 \times 4$  マスの図案を利用して  $8 \times 8$  マスの図案を作る方法

問 3 次の文章を読み、空欄 **コ** ・ **サ** に入れるのに最も適当なものを、後の解答群のうちから一つずつ選べ。

T 社では、あるサービスを展開している。そのサービスのスマートフォン専用のアプリを開発するにあたり、新規ユーザ登録の入力インタフェースとして、図 4 の生年の入力方法を考える。

図 4 では、年リストの中央に表示されている初期値は 2000 になっている。2008 年生まれの人は、スクロール操作によりホイールを回転させ、リストの中央に 2008 を表示させることで入力する。この操作では、スクロールする移動距離(以下、スクロール距離と呼ぶ。)は 8 である。また、1997 年生まれの人のスクロール距離は 3 となる。なお、年リストの上限は 2100、下限は 1900 とし、それより上下にはスクロールしない。

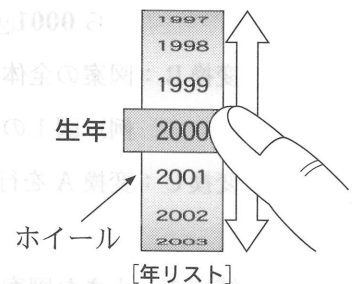


図 4 生年の入力方法

生年の初期値について、直近 1 年間に登録されたユーザの生年データの分布(図 5)をもとに検討する。表 1 に生年データの分布をもとに考えた生年の初期値の候補を示す。新規に登録するユーザの生年の分布は、直近 1 年間に登録されたユーザの生年の分布と変わらないと考えると、この分布に対してスクロール距離の平均ができるだけ小さくなるような初期値は、表 1 の中では生年データの **コ** である。

誕生月についても、図 6 に示すスクロール操作で入力することを考える。月リストはスクロールにより 1 周すると元に戻る。つまり、12 の下は 1、1 の上は 12 である。新規ユーザの誕生月の分布に偏りが無い場合、登録に要するスクロール距離の平均は、月リストの初期値を **サ** と考えられる。

なお、月リストをスクロールしても、入力済みの生年には影響しない。

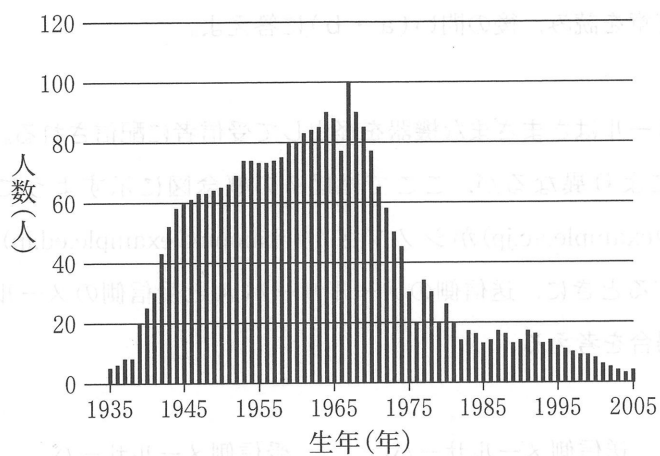


図5 直近1年間に登録されたユーザの生年データの分布

表1 生年の初期値の候補

|                    |      |
|--------------------|------|
| 中央値                | 1961 |
| 最頻値                | 1967 |
| 範囲(1935～2005)の中央の値 | 1970 |
| 最大値                | 2005 |

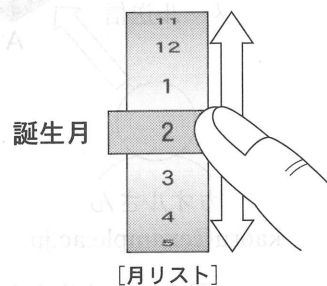


図6 誕生月の入力方法

**コ** の解答群

- |                      |       |
|----------------------|-------|
| ① 中央値                | ① 最頻値 |
| ② 範囲(1935～2005)の中央の値 | ③ 最大値 |

**サ** の解答群

- ① 1 にするとそれ以外にしたときよりも小さくなる
- ① 12 にするとそれ以外にしたときよりも小さくなる
- ② 6 か 7 のいずれかにするとそれら以外にしたときよりも小さくなる
- ③ どの月にしても変わらない

問 4 次の文章を読み、後の問い(a・b)に答えよ。

電子メールはさまざまな機器を経由して受信者に配信される。経由する機器は場合により異なるが、ここでは図7の概念図に示すように、カオルさん(kaoru@example.ac.jp)がシノブさん(shinobu@example.ed.jp)にメールを送ろうとすると、送信側のメールサーバAと受信側のメールサーバBがかわる場合を考える。

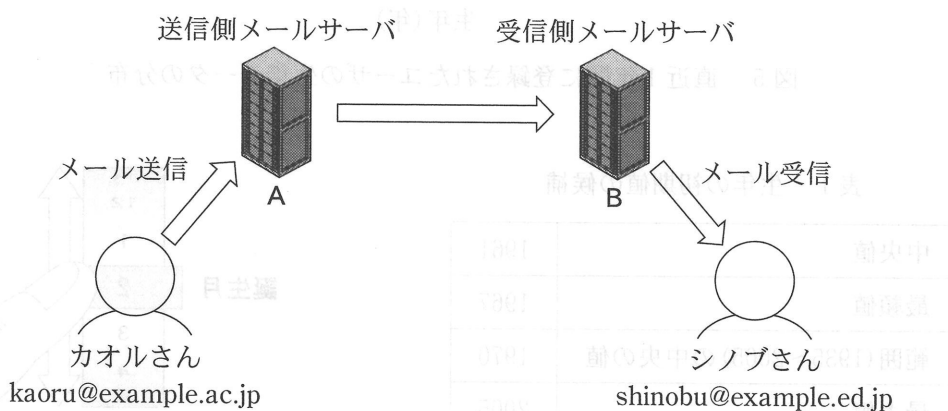


図7 カオルさんからシノブさんへメールを送るときの概念図

- a カオルさんが、次に示す誤りの例あ～うのようにシノブさんのメールアドレスを間違えて送信したとき、配信途中のどこかでそのメールをシノブさんに届けられないことが検出される。あ～うのそれぞれの場合で、そのメールが届けられないことを最初に検出したサーバとその理由の組合せとして正しいものを、後の①～⑦のうちから一つずつ選べ。ただし、同じものを繰り返し選んでもよい。なお、誤って指定した sinobu という文字列に対応するユーザ名やドメイン名は存在しないものとする。exmple.ed.jp という文字列についても、同様とする。

誤りの例(下線部の文字列には綴り<sup>つづ</sup>りの間違いがある。)

あ sinobu@example.ed.jp   

い example.ed.jp@sinobu   

う sinobu@exmple.ed.jp   

|   | サーバ | 理由                      |
|---|-----|-------------------------|
| ① | A   | 送信側のドメイン名が存在しない         |
| ② | B   | 送信側のドメイン名が存在しない         |
| ③ | A   | 受信側のドメイン名が存在しない         |
| ④ | B   | 受信側のドメイン名が存在しない         |
| ⑤ | A   | ユーザ名が存在しない              |
| ⑥ | B   | ユーザ名が存在しない              |
| ⑦ | A   | 受信側のドメイン名とユーザ名の両方が存在しない |
| ⑧ | B   | 受信側のドメイン名とユーザ名の両方が存在しない |

b 図7のメールを送る過程で用いられる通信先のIPアドレスを特定する仕組みとして最も適当なものを，次の①～④のうちから一つ選べ。

- ① DNS
- ② WWW
- ③ パケット通信
- ④ ルーティング

第2問 次の問い(A・B)に答えよ。(配点 30)

A 次の文章を読み、後の問い(問1～3)に答えよ。

A国では、役所が発行する**住民証明**に、住民の氏名、生年月日、住所(以下、この三つをまとめて**住民情報**と呼ぶ。)が記載されている。住民情報は就職先企業などから求められたときに自分自身で申告することができる。企業は、申告された住民情報が役所に登録された住民情報と一致しているかを確認するために、役所が発行する住民証明の提出を一緒に求めることがある。紙に印刷された役所発行の**住民証明(紙媒体)**を取得するには、請求者が役所に直接訪れるだけでなく、コンビニエンスストアなどに設置されている**多機能プリンタ**で印刷しても入手することができる。

図1は、A国の住民証明(紙媒体)発行のための情報システムを表したものである。ここで、①、②、…の数字は操作・要求やデータの移動が発生する時間的順序を表している。なお、以降の問題を含め、住民証明の請求に必要な個人認証は請求操作に含まれており、認証に関する不正行為への対策は別途行われているとする。

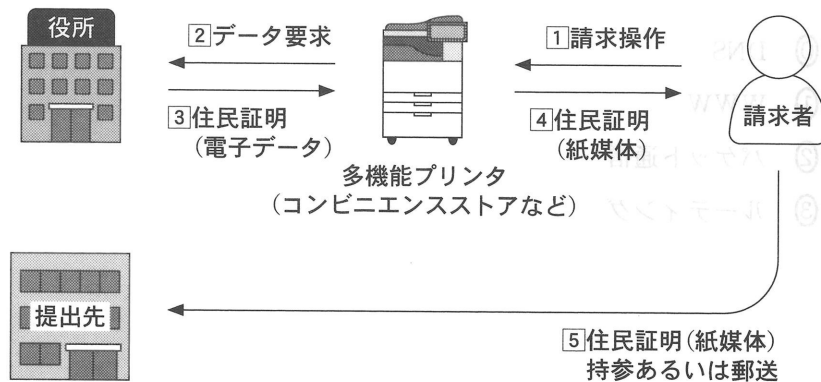


図1 多機能プリンタで住民証明(紙媒体)を入手できる情報システム

問 1 次の文章を読み、空欄 **ア** ・ **イ** に入れるのに最も適当なものを、後の解答群のうちから一つずつ選べ。ただし、解答の順序は問わない。

図 1 に示した情報システムでは、提出先に住民証明(紙媒体)を持参あるいは郵送しなければならない。そこで、住民証明(電子データ)を電子的に提出できるように、情報システムを改良する。

図 2 に示す情報システムでは、請求者が個人保有のパーソナルコンピュータ(PC)からインターネットを介して住民証明(電子データ)を請求して入手できる。図 1 の情報システムと異なり、図 2 の情報システムには「**ア**」や「**イ**」といった特徴がある。

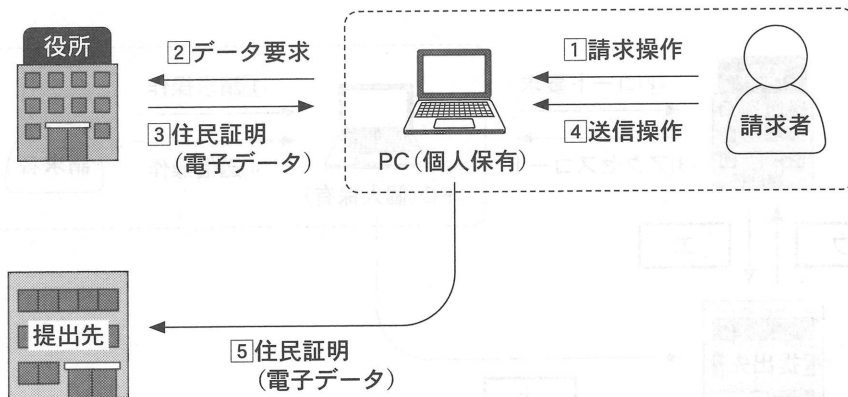


図 2 個人保有の PC で住民証明(電子データ)を入手できる情報システム

**ア** ・ **イ** の解答群

- ① デジタルデバインドが解消され、誰でも簡単に請求できる。
- ② 住民証明(電子データ)を入手するにはコンビニエンスストアを訪れる必要がある。
- ③ 自宅や出先から住民証明(電子データ)を請求できる。
- ④ 請求者が操作しなくても提出先は住民証明(電子データ)を入手できる。
- ⑤ 入手した住民証明(電子データ)をインターネット経由で送ることができる。

問 2 次の文章を読み、空欄 **ウ** ～ **カ** に入れるのに最も適切なものを、後の解答群のうちから一つずつ選べ。

図 2 に示した情報システムでは、入手した住民証明(電子データ)を請求者が勝手に変更してしまうおそれがある。請求者が住民証明(電子データ)を変更できないようにするために、**アクセスコード**を利用する方式に情報システムを改良する。

図 3 に示す情報システムでは、役所は請求ごとに**アクセスコード**を発行する。**アクセスコード**を役所に送信することで、対応する住民証明(電子データ)を入手できる。

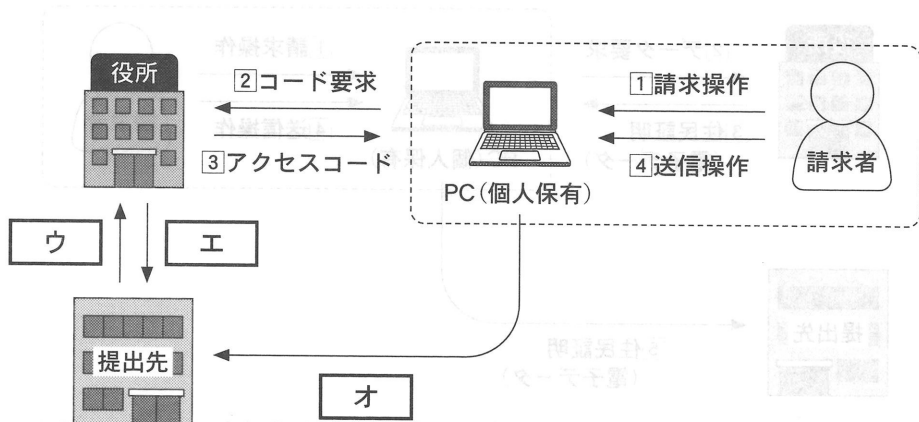


図 3 アクセスコードを利用する方式の情報システム

図 2 の情報システムと異なり、図 3 の情報システムでは **カ**。

ウ ～ オ の解答群

- ④ ⑤コード要求
- ① ⑤アクセスコード
- ② ⑤住民証明(電子データ)
- ③ ⑥コード要求
- ④ ⑥アクセスコード
- ⑤ ⑥住民証明(電子データ)
- ⑥ ⑦コード要求
- ⑦ ⑦アクセスコード
- ⑧ ⑦住民証明(電子データ)

カ の解答群

- ④ 請求者が住民証明(電子データ)の内容を見ることができる
- ① 提出先が住民証明(電子データ)の内容を見ることができる
- ② 請求者が住民証明(電子データ)を送る必要がない
- ③ 役所が住民証明(電子データ)を提供する必要がない

問 3 次の文章を読み、空欄 **キ** ～ **ケ** に入れるのに最も適当なものを、後の解答群のうちから一つずつ選べ。

図 3 に示した情報システムでは、アクセスコードが第三者に漏れると住民情報が流出するおそれがある。住民情報は請求者自身が提出先に伝えることができるので、そのことを利用して情報システムを図 4 のように改良する。図 4 に示す情報システムでは、**確認依頼コード**を利用する。なお、図 4 では、図の一部を「？」で隠してある。

図 1 ～ 3 に示した情報システムでは、役所の役割は **キ** することである。一方、図 4 に示す情報システムでは、役所の役割を **キ** することから **ク** することに変更している。図 3 の情報システムにおけるアクセスコードとは異なり、図 4 の情報システムにおける**確認依頼コード**は、**ケ**。

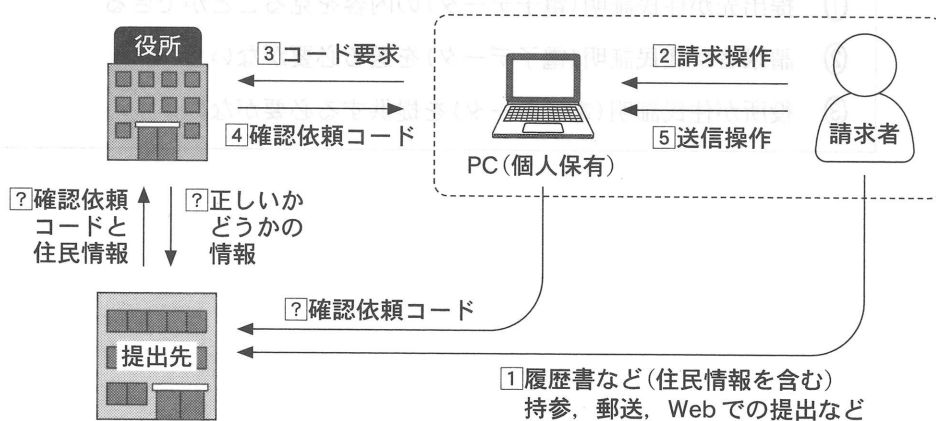


図 4 確認依頼コードを利用する方式の情報システム

キ

ク

の解答群

- |              |               |
|--------------|---------------|
| ① 住民情報を入手    | ① 住民情報を提供     |
| ② 住民情報を削除    | ③ 住民情報を電子化    |
| ④ 住民情報の誤りを訂正 | ⑤ 住民情報の正しさを証明 |

ケ

の解答群

- ① コードのコピーを誰も作れない
- ② コードだけを送信すれば住民情報を入手できる
- ③ コードだけを送信しても住民情報を入手できない
- ④ コードに住民情報が含まれる
- ⑤ コードに住民情報が含まれない

B 次の文章を読み、後の問い(問1～4)に答えよ。

ゲームの画面などは一つの画像ではなく、背景の画像やキャラクターの画像など複数の画像を組み合わせて作られている。例えば、図5(A)の画像の上に、コンピュータを使って描いた図5(B)のようなゲームのキャラクター画像をそのまま重ねると、図5(C)のように図5(B)の白い背景が表示されてしまう。

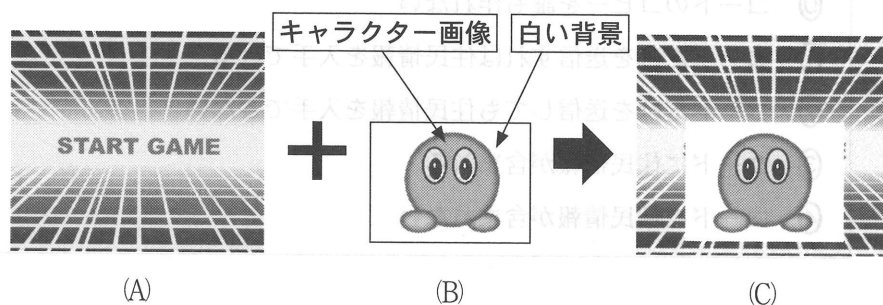


図5 背景の画像にキャラクターの画像を重ねる様子

図5(B)の白い背景を透過させ、その部分に図5(A)の画像を表示させたい。これは、画像を重ね合わせ、画素単位のビット演算を組み合わせて実現することができる。ここでは1画素(点)を4ビットで表現した16階調のグレースケールとし、黒色の点の値は0000<sub>(2)</sub>、白色の点の値は1111<sub>(2)</sub>とする。

最初に、図5(B)のキャラクターの部分すべてを白色、透過したい部分をすべて黒色にした図6の画像を作成しておく。

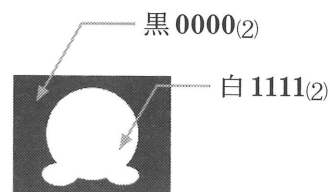


図6

次に、図5(A)の画像に図6の画像を図5(C)と同様に重ねて、(a)重なる各点の4ビットについてビットごとにOR演算をすると図7の画像になる。さらに、(b)図7の画像に図5(B)の画像を重ねて、重なる各点の4ビットについて、ビットごとに「ある演算」を行うと図8の画像になる。

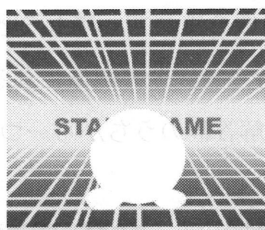


図 7

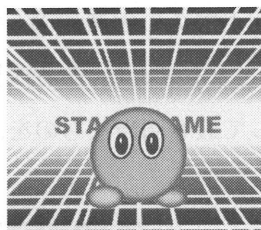


図 8

なお，以降の問題で使用する OR 演算(論理和)，AND 演算(論理積)，NOT 演算(否定)の真理値表は表 1 のとおりである。

表 1 論理演算子の真理値表

| OR 演算(論理和) |     |                   | AND 演算(論理積) |     |                    | NOT 演算(否定) |         |
|------------|-----|-------------------|-------------|-----|--------------------|------------|---------|
| $x$        | $y$ | $x \text{ OR } y$ | $x$         | $y$ | $x \text{ AND } y$ | $x$        | NOT $x$ |
| 0          | 0   | 0                 | 0           | 0   | 0                  | 0          | 1       |
| 0          | 1   | 1                 | 0           | 1   | 0                  | 1          | 0       |
| 1          | 0   | 1                 | 1           | 0   | 0                  |            |         |
| 1          | 1   | 1                 | 1           | 1   | 1                  |            |         |

問 1 下線部(a)の操作について，図 6 の白色の点  $1111_{(2)}$  と重なる図 5 (A)の点の値が  $1010_{(2)}$ ，図 6 の黒色の点  $0000_{(2)}$  と重なる図 5 (A)の点の値が  $0110_{(2)}$  であった場合のそれぞれについて，ビットごとに OR 演算した結果として最も適当なものを，後の解答群のうちから一つずつ選べ。

点  $1111_{(2)}$  と点  $1010_{(2)}$  とのビットごとの OR 演算結果 コ

点  $0000_{(2)}$  と点  $0110_{(2)}$  とのビットごとの OR 演算結果 サ

コ ・ サ の解答群

|                |                |                |                |
|----------------|----------------|----------------|----------------|
| ① $0000_{(2)}$ | ② $0011_{(2)}$ | ③ $0101_{(2)}$ | ④ $0110_{(2)}$ |
| ⑤ $1001_{(2)}$ | ⑥ $1010_{(2)}$ | ⑦ $1100_{(2)}$ | ⑧ $1111_{(2)}$ |

問 2 下線部(b)「図 7 の画像に図 5 (B)の画像を重ねて、重なる各点の 4 ビットについて、ビットごとに「ある演算」を行うと図 8 の画像になる」における「ある演算」として最も適当なものを、後の解答群のうちから一つ選べ。 シ

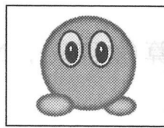


図 5 (B)(再掲)

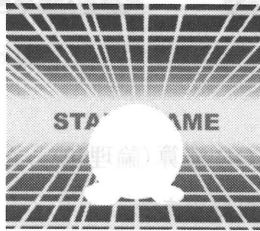


図 7 (再掲)

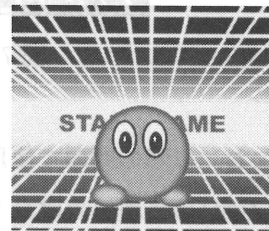


図 8 (再掲)

シ の解答群

① AND 演算

② OR 演算

③ AND 演算の後に NOT 演算

④ OR 演算の後に NOT 演算

問 3 図 5 (B)から図 6 を作成するには、図 5 (B)の背景色 (白一色) を指定して背景の領域を選択すれば容易に実現できる。しかし、図 9 のような背景が一色でない画像から、図 10 の画像を作成することは容易ではない。そこで、色の範囲を指定して、その色から成る領域を選択することができる画像編集ソフトウェアを使って背景を選択することを考える。図 11 は画像編集ソフトウェアで表した図 9 の各階調の画素数を表したヒストグラムである。なお、図 9 の画像は 1 画素を 8 ビットで表現した 256 階調のグレースケールである。図 9 の背景 (熊以外) だけを選択するには階調のどの範囲を指定すれば良いか。最も適当なものを、図 11 中の①～④のうちから一つ選べ。 ス

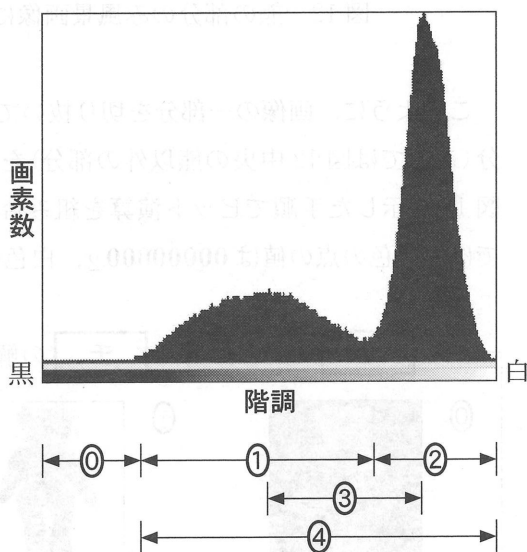
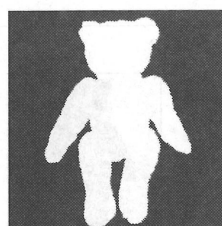


図 11 図 9 の各階調のヒストグラム

問 4 次の文章を読み、空欄 **セ** ～ **チ** に入れるのに最も適当なものを、後の解答群のうちから一つずつ選べ。

ポスターやパンフレットなどで、風景画像(図 12 左)に別の画像の一部分(図 12 中央の熊の部分)を切り抜いて重ねられたような合成画像(図 12 右)を見ることがある。



図 12 熊の部分のみ風景画像に合成する様子

このように、画像の一部分を切り抜いて別の画像に重ねるには、背景の部分(ここでは図 12 中央の熊以外の部分)を透過させる必要がある。これは、図 13 で示した手順でビット演算を組み合わせれば実現できる。なお、ここでは、黒色の点の値は  $00000000_{(2)}$ 、白色の点の値は  $11111111_{(2)}$  とする。

**セ** ・ **ソ** , **チ** の解答群

|                                                                                                                  |                                                                                                          |                                                                                                                   |
|------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------|
| <p>⑥</p>  <p>背景：黒</p>         | <p>⑦</p>  <p>背景：白</p> | <p>⑧</p>  <p>熊：黒</p>          |
| <p>⑨</p>  <p>背景：白<br/>熊：黒</p> | <p>⑩</p>  <p>熊：白</p>  | <p>⑪</p>  <p>背景：黒<br/>熊：白</p> |

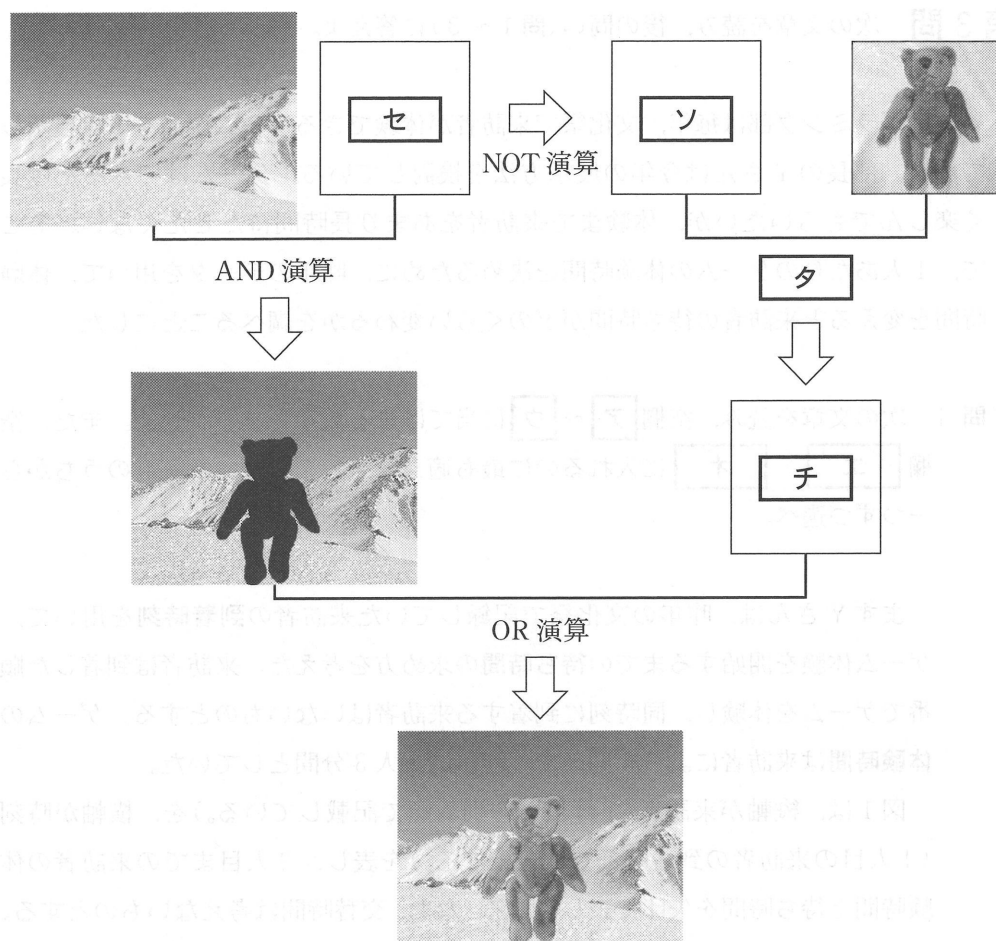


図 13 風景画像に別の画像の一部を合成する手順

- タ** の解答群
- |                    |                   |
|--------------------|-------------------|
| ① AND 演算           | ① OR 演算           |
| ② AND 演算の後に NOT 演算 | ③ OR 演算の後に NOT 演算 |

### 第3問 次の文章を読み、後の問い(問1～3)に答えよ。(配点 25)

プログラミング部は毎年、文化祭で来訪者が体験できるゲーム作品を1点展示しており、部長のYさんは今年の展示方法を検討している。ゲームはできるだけ長く楽しんでもらいたいが、体験まで来訪者をあまり長時間待たせたくない。そこで、1人あたりのゲームの体験時間を決めるために、昨年のデータを用いて、体験時間を変えると来訪者の待ち時間がどのくらい変わるかを調べることにした。

問1 次の文章を読み、空欄 **ア** ～ **ウ** に当てはまる数字をマークせよ。また、空欄 **エ** ・ **オ** に入れるのに最も適当なものを、後の解答群のうちから一つずつ選べ。

まずYさんは、昨年の文化祭で記録していた来訪者の到着時刻を用いて、ゲーム体験を開始するまでの待ち時間の求め方を考えた。来訪者は到着した順番でゲームを体験し、同時刻に到着する来訪者はいないものとする。ゲームの体験時間は来訪者によらず同一で、昨年は1人3分間としていた。

図1は、縦軸が来訪者(1行目から到着順で記載している。)を、横軸が時刻(1人目の来訪者の到着時刻を0分とする。)を表し、3人目までの来訪者の体験時間と待ち時間を矢印で表している。なお、交替時間は考えないものとする。

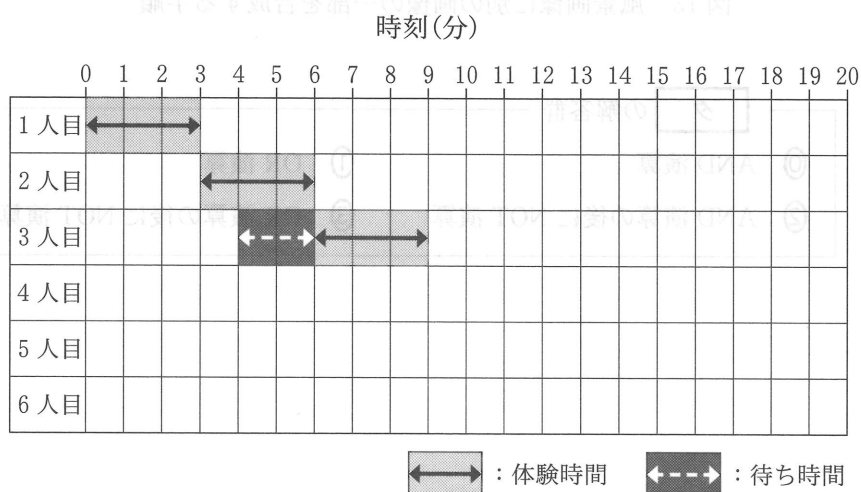


図1 昨年のゲームの体験時間と待ち時間(3人目の来訪者まで)

1 人目の来訪者の待ち時間はなく、ゲーム体験の開始時刻は 0 分、終了時刻は 3 分である。2 人目は到着時刻が 3 分だったので、待ち時間がない。一方、3 人目は 4 分に到着したが、2 人目の終了時刻が 6 分のため、2 分間の待ち時間が発生している。

次に Yさんは、図 1 の残りの部分を完成させ、6 人目までの来訪者の到着時刻、開始時刻、終了時刻、待ち時間を表 1 のとおりに整理した。4 人目は到着時刻が 10 分で、3 人目の終了時刻は  分のため、4 人目の待ち時間はない。5 人目の待ち時間は  分間、6 人目の待ち時間は  分間である。

表 1 昨年の来訪者の待ち時間を整理した結果

|      | 到着時刻 | 開始時刻 | 終了時刻                           | 待ち時間                           |
|------|------|------|--------------------------------|--------------------------------|
| 1 人目 | 0    | 0    | 3                              | 0                              |
| 2 人目 | 3    | 3    | 6                              | 0                              |
| 3 人目 | 4    | 6    | <input type="text" value="ア"/> | 2                              |
| 4 人目 | 10   | 10   | 13                             | 0                              |
| 5 人目 | 11   | ?    | ?                              | <input type="text" value="イ"/> |
| 6 人目 | 12   | ?    | ?                              | <input type="text" value="ウ"/> |

(表の一部を“?”で隠してある)

これらの結果から Yさんは、2 人目以降の来訪者について、開始時刻が「より遅く到着した場合には 、それ以外の場合は 」になること、終了時刻が「来訪者本人の開始時刻と体験時間の和」、待ち時間が「来訪者本人の開始時刻と  の差」で求められることを確認した。

・の解答群

- |              |               |
|--------------|---------------|
| ① 来訪者本人の到着時刻 | ① 直前の来訪者の到着時刻 |
| ② 来訪者本人の終了時刻 | ③ 直前の来訪者の終了時刻 |

問 2 次の文章を読み、空欄 **力** ～ **コ** に入れるのに最も適当なものを、後の解答群のうちから一つずつ選べ。ただし、空欄 **力** ・ **キ** , **コ** は同じものを繰り返し選んでもよい。また、空欄 **力** ・ **キ** の解答の順序は問わない。

次に、Yさんは問1で整理した考え方に基づき、昨年の6人目までの来訪者の到着時刻を用いて2人目以降の来訪者の待ち時間を求めるプログラムを作成した(図2)。

(01)行目の変数 **taiken** は体験時間を表す変数で、昨年の体験時間(3分間)を表す数値3を代入している。配列 **Touchaku**, **Kaishi**, **Shuryou** はそれぞれ、各来訪者の到着時刻、開始時刻、終了時刻を格納する。すべての配列の添字は1から始まり、来訪者の到着順を表している。例えば、各配列の添字1の要素には、1人目の来訪者に関する値を格納する。配列 **Touchaku** の各要素には来訪者の到着時刻を設定し、配列 **Kaishi**, **Shuryou** の各要素は0で初期化する。(03)行目では、関数「要素数」を用いて変数 **kyakusu** に来訪者数を代入する。

#### 【関数の説明】

**要素数 (配列)** …引数として配列が与えられ、その配列の要素数を返す。  
例えば、配列 **x** が **[6, 9, 1]** であるとき、**要素数(x)** は3となる。

(06)行目では、1人目の終了時刻を設定する。(07)～(10)行目の繰り返しでは、**i** 番目の来訪者の開始時刻、終了時刻、待ち時間を順に求める。(08)行目で関数「最大値」を用いて開始時刻を、(09)行目で終了時刻を求める。(10)行目で待ち時間を表示する。

#### 【関数の説明】

**最大値 (x, y)** …引数として与えられた二つの値の最大値を返す。  
例えば、**最大値(1, 5)** は5となり、**最大値(2, 2)** は2となる。

```

(01) taiken = 3
(02) Touchaku = [0, 3, 4, 10, 11, 12]
(03) kyakusu = 要素数(Touchaku)
(04) Kaishi = [0, 0, 0, 0, 0, 0]
(05) Shuryou = [0, 0, 0, 0, 0, 0]
(06) Shuryou[1] = taiken
(07) i を 2 から kyakusu まで 1 ずつ増やしながら繰り返す:
(08) | Kaishi[i] = 最大値(  ,  )
(09) | Shuryou[i] = 
(10) | 表示する(i, "人目の待ち時間:",
      |  -  , "分間")

```

図2 2人目以降の来訪者の待ち時間を求めるプログラム

・  ,  の解答群

- |                   |                  |
|-------------------|------------------|
| ① Touchaku[i - 1] | ① Shuryou[i - 1] |
| ② Touchaku[i]     | ③ Shuryou[i]     |
| ④ Touchaku[i + 1] | ⑤ Shuryou[i + 1] |

・  の解答群

- |                 |                           |
|-----------------|---------------------------|
| ① Kaishi[i - 1] | ① Kaishi[i] + Touchaku[i] |
| ② Kaishi[i]     | ③ Kaishi[i] + Shuryou[i]  |
| ④ Kaishi[i + 1] | ⑤ Kaishi[i] + taiken      |

問 3 次の文章を読み、空欄  ・  に入れるのに最も適当なものを、後の解答群のうちから一つずつ選べ。また、空欄  ・  に入れるのに最も適当なものを、図 3 の①～③のうちから一つずつ選べ。空欄  に当てはまる数字をマークせよ。

Yさんは、昨年の様子から来訪者を10分間以上は待たせられないと考え、最も長く待たされる来訪者の待ち時間(以下、**最長待ち時間**と呼ぶ。)が10分間未満となる体験時間を調べるプログラムを作成した(図3)。

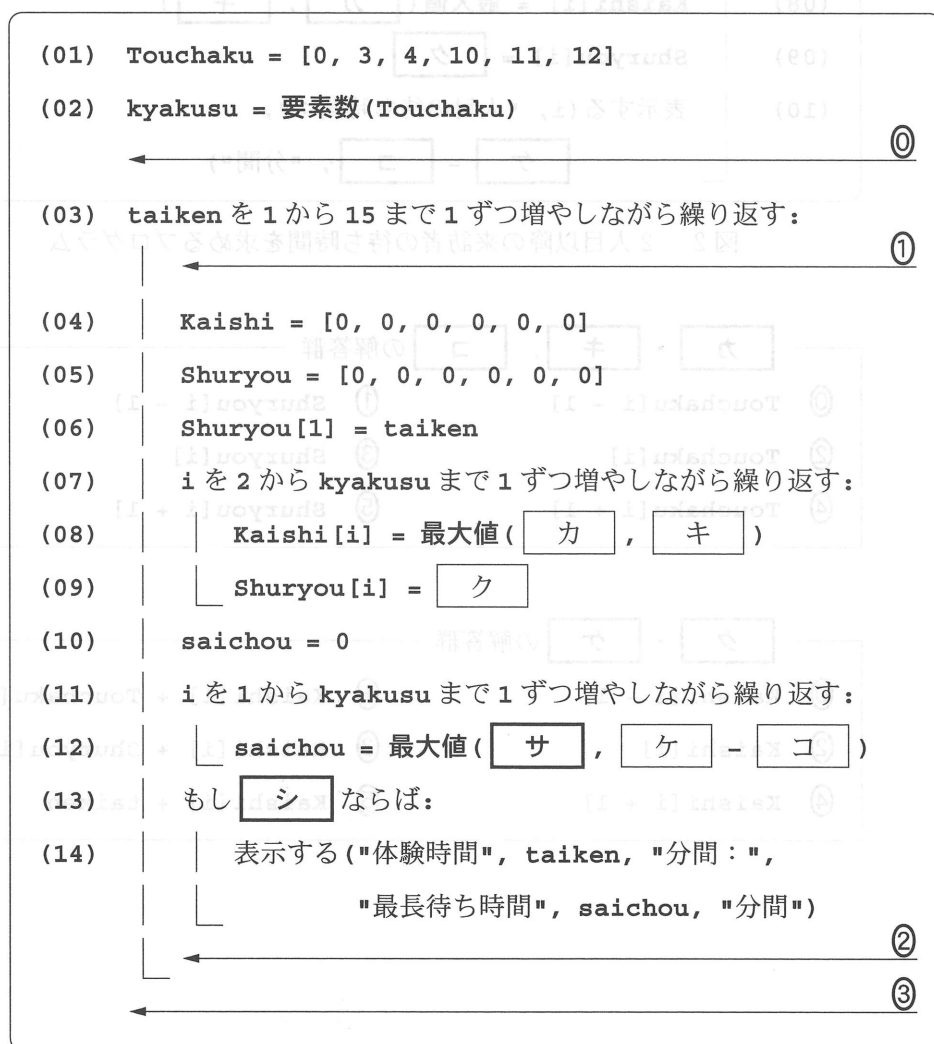


図 3 最長待ち時間が10分間未満となる体験時間を調べるプログラム

(03)行目からの繰り返しで各体験時間に対する最長待ち時間を求める。体験時間の上限は15分間とした。(04)～(09)行目の処理は図2のプログラムの(04)～(09)行目の処理と同じである。(10)～(14)行目で体験時間が **taiken** 分間の場合の最長待ち時間を求め、10分間未満ならば体験時間と最長待ち時間を表示する。図3のプログラムの実行結果は、図4のようになった。

体験時間 1 分間：最長待ち時間 0 分間  
 体験時間 2 分間：最長待ち時間 2 分間  
 体験時間 3 分間：最長待ち時間 4 分間  
 体験時間 4 分間：最長待ち時間 8 分間

図4 図3のプログラムの実行結果

ある体験時間に対して最長待ち時間が10分間以上となった時点で、(04)行目以降の繰り返しをやめても実行結果は変わらない。そこでYさんは、最長待ち時間が10分間以上となった時点で処理を止めるために、(03)行目を「(taiken <= 15) and (  )の間繰り返す:」に変更した。「and」は「かつ」を意味する論理演算子であり、左右の式がともに真のときにだけ真となる。さらに、「taiken = 1」と「saichou = 0」を  の位置に挿入した。また、「taiken = taiken + 1」を  の位置に挿入した。

配列 **Kaishi** の初期化処理の実行回数は、修正前のプログラムでは15回だったが、修正後のプログラムでは  回となった。

の解答群

- |               |             |
|---------------|-------------|
| ① Touchaku[i] | ① Kaishi[i] |
| ② Shuryou[i]  | ③ saichou   |

の解答群

- |                |               |
|----------------|---------------|
| ① saichou > 10 | ① taiken > 10 |
| ② saichou < 10 | ③ taiken < 10 |

**第4問** 次の文章を読み、後の問い(問1～4)に答えよ。(配点 25)

桜の花が好きなTさんは桜の開花日に興味をもち、気象庁のデータを用いて、桜の開花日を分析することにした。

問1 次の文章を読み、後の問い(a・b)に答えよ。

Tさんは気象庁のデータから、桜の開花日を観測している場所(観測点)ごとに、2014年から2023年までの開花日を表1にまとめた。例えば、稚内の2023年の開花日は5月3日である。なお、この気象庁のデータはオープンデータである。

表1 桜の開花日

| 観測点  | 2014年 | 2015年 | 2021年 | 2022年 | 2023年 |
|------|-------|-------|-------|-------|-------|
| 稚内   | 5月11日 | 5月3日  | 5月8日  | 5月6日  | 5月3日  |
| 旭川   | 5月2日  | 4月27日 | 4月29日 | 4月26日 | 4月25日 |
| 網走   | 5月7日  | 4月30日 | 5月6日  | 4月27日 | 4月28日 |
| 那覇   | 1月15日 | 1月15日 | 1月4日  | 1月11日 | 1月7日  |
| 南大東島 | 1月7日  | 1月15日 | 0     | 1月28日 | 2月7日  |

a 次の二つの文A、Bは、オープンデータについての説明である。その正誤の組合せとして最も適当なものを、後の①～③のうちから一つ選べ。

ア

A オープンデータは、誰でも入手可能である。

B オープンデータを加工・編集する場合には、利用申請が必須である。

|   | ① | ② | ③ |
|---|---|---|---|
| A | 正 | 正 | 誤 |
| B | 正 | 誤 | 正 |

- b 次の文章中の空欄  ・  に入れるのに最も適当なものを、後の解答群のうちから一つずつ選べ。

Tさんは、表1において、2021年の南大東島の開花日に0が入力されていることを見つけ、南大東島はその年に開花日の記録がない観測点であることがわかった。つまり、表1の0は  を意味するので、Tさんは表1から  が含まれる行を除外したデータを用いることにした。

Tさんはまず、4月1日より早く開花した観測点の数が、年ごとにどのように変化しているかを見るために、  とした折れ線グラフにより可視化し、変化の傾向を把握した。

の解答群

- ② 補数      ① 乱数      ② 代表値      ③ 欠損値

の解答群

- ② 横軸を月、縦軸を2023年の各月に開花した観測点の数  
① 横軸を月、縦軸を10年間分の各月に開花した観測点の総数  
② 横軸を年、縦軸を各年の3月末までに開花した観測点の数  
③ 横軸を年、縦軸を各年の4月に開花した観測点の数

問 2 次の文章を読み、空欄 **エ** ～ **カ** に入れるのに最も適当なものを、  
後の解答群のうちから一つずつ選べ。

Tさんは、各年の開花日に関する経験則として、次の二つの仮説があること  
を見つけた。なお、気温の単位はセ氏(℃)である。

**仮説 1 (400 度の法則)**

2月1日からの各日の平均気温の和が 400 を超えると開花する。

**仮説 2 (600 度の法則)**

2月1日からの各日の最高気温の和が 600 を超えると開花する。

そこでTさんは、ある年で2月1日以降に開花した観測点を対象に、この  
二つの仮説を検証することにした。まず、この年の2月1日以降の平均気温と  
最高気温のデータを入手した。さらに、各観測点で2月1日からの平均気温の  
和が初めて 400 を超えた日を **400 度開花推定日**、2月1日からの最高気温の和  
が初めて 600 を超えた日を **600 度開花推定日**とし、同じ年の開花日(**実際の開  
花日**)と合わせて、表 2 にまとめた。

次に、Tさんは各観測点で開花推定日が実際の開花日からずれている日数を  
確認した。実際の開花日を基準にして、開花推定日が早ければ負の値、遅けれ  
ば正の値、一致していれば 0 として、400 度開花推定日と実際の開花日との差  
(**400 度開花差**)、ならびに 600 度開花推定日と実際の開花日との差(**600 度開花  
差**)を計算し、表 3 にまとめた。

例えば、名古屋の 400 度開花差は **エ**，600 度開花差は **オ** と計算  
できた。また、新潟と奈良のそれぞれで、どちらの開花推定日が実際の開花日  
に近いが、表 3 だけを用いて比較すると、**カ** ことがわかった。なお、開  
花差の絶対値が小さいことを「実際の開花日に近い」とする。

表2 開花推定日

| 観測点 | 400 度開花<br>推定日 | 600 度開花<br>推定日 | 実際の<br>開花日 |
|-----|----------------|----------------|------------|
| 稚内  | 5 月 31 日       | 5 月 21 日       | 5 月 8 日    |
| 旭川  | 5 月 26 日       | 5 月 7 日        | 4 月 29 日   |
| 新潟  | ?              | ?              | ?          |
| 名古屋 | 3 月 19 日       | 3 月 16 日       | 3 月 17 日   |
| 甲府  | 3 月 20 日       | 3 月 14 日       | 3 月 18 日   |
| 奈良  | ?              | ?              | ?          |

表3 開花差

| 観測点 | 400 度<br>開花差 | 600 度<br>開花差 |
|-----|--------------|--------------|
| 稚内  | 23           | 13           |
| 旭川  | 27           | 8            |
| 新潟  | 4            | -1           |
| 名古屋 | エ            | オ            |
| 甲府  | 2            | -4           |
| 奈良  | -1           | -5           |

(表の一部を“?”で隠してある)

エ・オの解答群

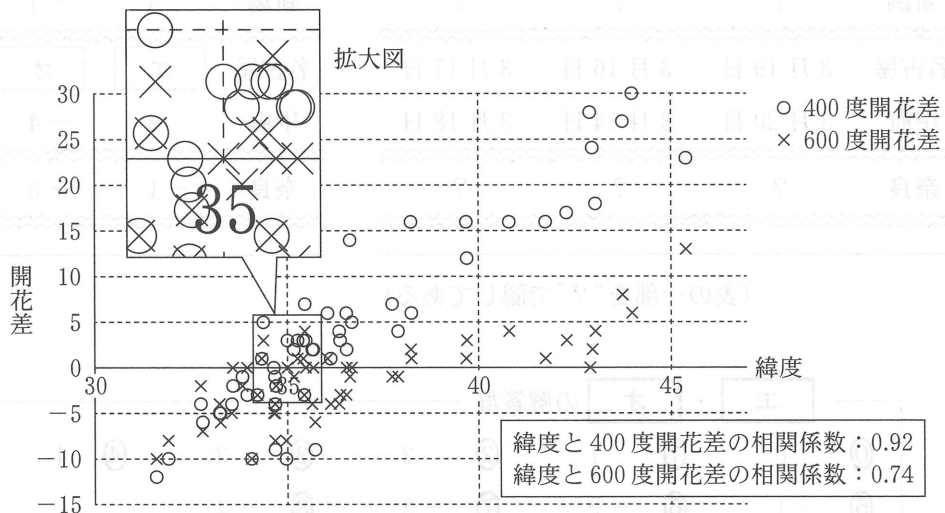
- ① 0      ② 1      ③ 2      ④ 3      ⑤ 4  
 ⑥ -1      ⑦ -2      ⑧ -3      ⑨ -4

カの解答群

- ① 新潟と奈良の両方とも、400 度開花推定日の方が近い  
 ② 新潟と奈良の両方とも、600 度開花推定日の方が近い  
 ③ 新潟は 400 度開花推定日の方が近く、奈良は 600 度開花推定日の方が近い  
 ④ 新潟は 600 度開花推定日の方が近く、奈良は 400 度開花推定日の方が近い  
 ⑤ 新潟と奈良の両方とも、表 3 だけではどちらの開花推定日が近いかわからない

問 3 次の文章を読み、後の問い(a・b)に答えよ。

Tさんは観測点の緯度も開花差に影響するのではないかと考えた。そこで、ある年のデータを用いて、横軸に緯度、縦軸に開花差をとった散布図(図1)を作成し、相関係数を求めた。なお、各観測点の緯度はすべて異なっている。



(注) 観測点の緯度は、観測点の都庁舎・市庁舎がある場所の緯度を用いた。

図1 緯度と開花差の散布図

a 図1と相関係数から読み取れることとして正しいものを、次の①～④のうちから二つ選べ。ただし、解答の順序は問わない。 キ・ク

- ① 400度開花差よりも600度開花差の方が、散布図の点の集まりが直線に近い。
- ② 開花推定日と実際の開花日が一致する観測点の数は、400度開花推定日よりも600度開花推定日の方が多い。
- ③ 緯度が最も高い観測点で、それぞれの開花差の値が最大となっている。
- ④ 任意の二つの観測点を比較して、緯度が高い観測点の方が、400度開花差の値か600度開花差の値の少なくとも一方が必ず大きい。
- ⑤ 緯度の値が40以上のすべての観測点で、400度開花推定日は600度開花推定日から一週間以上遅い。

- b 次の文章を読み、空欄 **ケ** ～ **サ** に入れるのに最も適当なものを、後の解答群のうちから一つずつ選べ。

Tさんは、緯度が高い観測点で2月1日以降に平均気温が氷点下(0℃以下)となる日があることを見つけ、開花差に影響するのではないかと考えた。

図1の作成に用いたデータから、2月1日以降に平均気温が氷点下であった日が存在する観測点(氷点下観測点)のみを集めたグループ( $G_1$ )とそれ以外の観測点を集めたグループ( $G_2$ )に分け、400度開花差に関する箱ひげ図(図2)を作成した。なお、どちらの箱ひげ図にも外れ値はなかった。

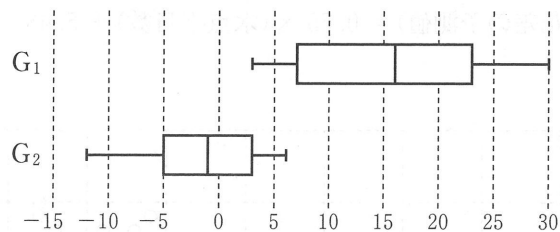


図2 400度開花差に関する箱ひげ図

図2から、開花差の絶対値が **ケ** 以上となる観測点は、 $G_1$  では半数以上あるが、 $G_2$  では1か所もないことから、グループ間の違いを確認した。また、400度開花推定日と実際の開花日が一致している観測点(一致観測点)について、 **コ** 一致観測点の数がわかり、 **サ**  $G_2$  に存在することがわかった。

**ケ** の解答群

- ① 5                      ② 10                      ③ 15                      ④ 20

**コ** ・ **サ** の解答群

- ① 図1だけで  
 ② 図2だけで  
 ③ 図1もしくは図2のいずれからでも  
 ④ 図1と図2の両方を組み合わせることで

問 4 次の文章を読み、空欄 **シ** ・ **ス** に入れるのに最も適当なものを、  
後の解答群のうちから一つずつ選べ。

Tさんは、氷点下観測点の開花推定日を補正できないかと考えた。

1年間のデータだけでは足りないと感じたTさんは、ある3年間の気象データを用いて、まず、各年の気象データから氷点下観測点を抽出し、2月1日以降に平均気温が氷点下であった日数(氷点下日数)を数えた。次に、抽出された3年分の氷点下観測点について、氷点下日数と400度開花差の散布図と回帰直線(図3)を作成した。なお、回帰直線の方程式は、次のとおりである。

$$(\text{開花差の予測値}) = 0.55 \times (\text{氷点下日数}) + 5.48$$

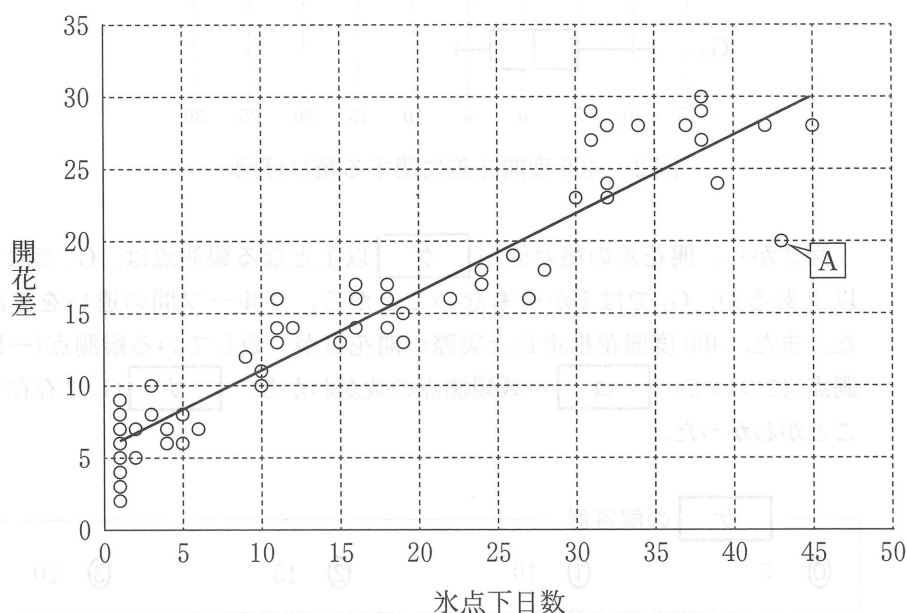


図3 氷点下日数と400度開花差の散布図と回帰直線

Tさんは、開花差の予測値の小数第1位を四捨五入した値(補正日数)を用いた次の補正方法により、実際の開花日に近い補正日(400度補正日)が得られるのではないかと考えた。ただし、補正日数が負の値の場合は、補正しないことにした。

### 補正方法

400 度開花推定日から補正日数分さかのぼった日を 400 度補正日とする。

400 度開花推定日より 400 度補正日の方が実際の開花日に近くなることを「うまく補正できる」とする。Tさんは図3から、**シ**ことがわかった。

さらに、Tさんは新たに別の年のデータを取得し、この補正方法を試した。観測点Xにおいて、400 度開花推定日が4月23日、氷点下日数は18日であった。これより、観測点Xの400 度補正日は**ス**と計算できた。実際の開花日が4月12日だったことから、この年の観測点Xではうまく補正できた。

#### **シ**の解答群

- ① すべての氷点下観測点でうまく補正できる
- ② 散布図上にある点Aで示した氷点下観測点を除いたすべての氷点下観測点でうまく補正できる
- ③ 氷点下日数が1日である氷点下観測点の一部を除いたすべての氷点下観測点でうまく補正できる
- ④ 回帰直線上または回帰直線よりも上側に点がある氷点下観測点だけうまく補正できる
- ⑤ 回帰直線上または回帰直線よりも下側に点がある氷点下観測点だけうまく補正できる

#### **ス**の解答群

- ① 4月5日      ② 4月8日      ③ 4月12日      ④ 4月15日