

試行問題

情報

I 次の問い(1)～(5)に答えよ。(45点)

[解答番号 ～]

(1) 個人情報保護に関する説明をした記述として、最も当てはまらないものを次の①～④のうちから一つ選べ。

- ① 基本四情報とは、氏名、住所、生年月日、性別から構成される4つの重要な個人情報のことである。
- ② 情報が流出することにより偏見や差別などが生じる可能性があるものに個人の信条がある。
- ③ 肖像権とは、他人の顔や姿などを本人の許可がなくても撮影して利用することができるものである。
- ④ 個人情報を集める際には利用目的を示す必要があり、その目的以外に利用しないことが義務づけられている。

(2) 情報の暗号化方式として、共通鍵暗号方式と公開鍵暗号方式がある。公開鍵暗号方式にのみ対応する特徴として、最も適当なものを次の①～④のうちから一つ選べ。

- ① 暗号化する前のデータを、鍵を用いて暗号化する。
- ② 送信者は受信者ごとに違う鍵を用意する必要がある。
- ③ 暗号化されたデータは、暗号文のままネットワーク上でやり取りされる。
- ④ 受信者は自分だけが所有する秘密鍵を用いて暗号化されたデータを元の状態に戻す。

(3) 次の文章の空欄 に当てはまる数字をマークせよ。また、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選べ。

音は、PCM（パルス符号変調）方式の場合、4段階の手順でデジタル化される。1・2段階目は、音をマイクロホンでアナログの電気信号に変換したものを、一定の時間間隔で区切り、区切った箇所における波の高さを取り出すことで、標本化する。取り出した波の高さに打つ点を標本点という。3段階目は、電圧を一定間隔に分割し、標本点の値に最も近い値に変換することで、量子化する。4段階目は、量子化した数値を2進法で表現することで、符号化する。例えば、図1に示すようなアナログ波形の場合、時間を6分割すると 個の標本点が得られる。

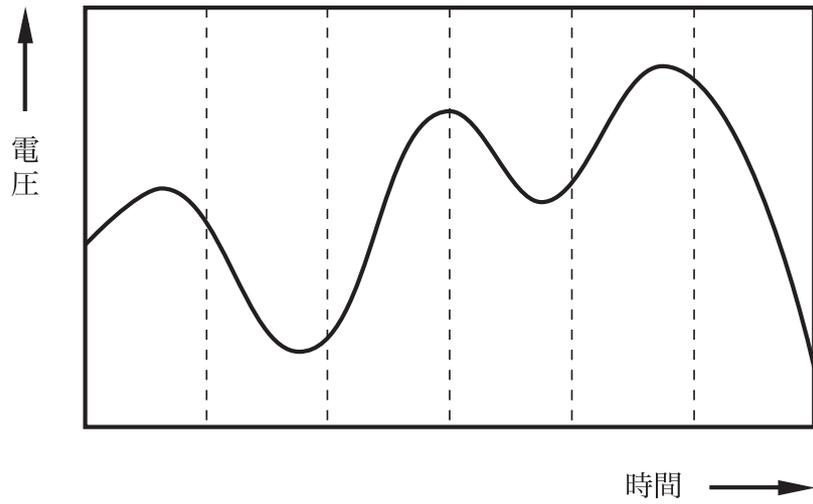
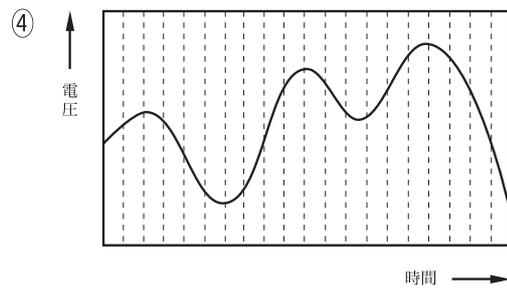
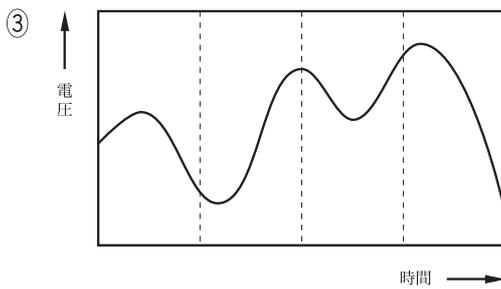
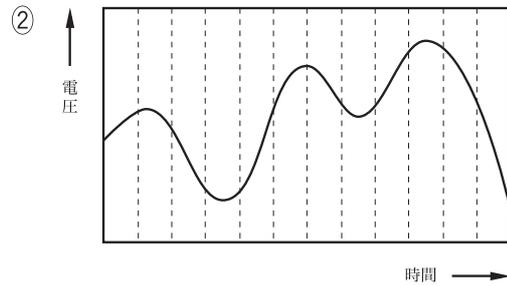
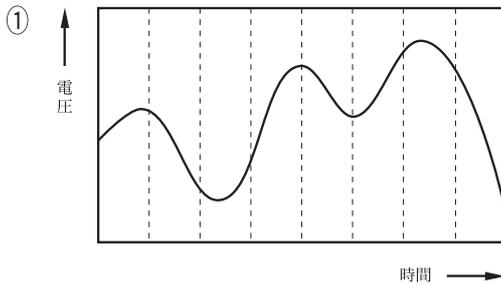


図1 アナログ波形

図1の波形を6分割すると、音の波形の再現性があまりよくない。波形の特徴をとらえ、かつデータ量が多くなり過ぎない程度に波形の再現性を高くした標本化周期は である。

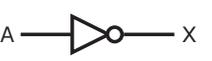
の解答群



(4) 次の文章を読み、空欄 **5** ・ **6** 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。

論理回路の基本は、論理積回路（AND 回路）、論理和回路（OR 回路）、否定回路（NOT 回路）の3種類であり、これらの図記号と真理値表は次の表1で示される。真理値表とは、論理回路について、すべての入力の場合とそれに対応する出力の関係を示した表である。

表1 図記号と真理値表

回路名	論理積回路	論理和回路	否定回路																																												
図記号																																															
真理値表	<table border="1"> <thead> <tr> <th colspan="2">入力</th> <th>出力</th> </tr> <tr> <th>A</th> <th>B</th> <th>X</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	入力		出力	A	B	X	0	0	0	0	1	0	1	0	0	1	1	1	<table border="1"> <thead> <tr> <th colspan="2">入力</th> <th>出力</th> </tr> <tr> <th>A</th> <th>B</th> <th>X</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	入力		出力	A	B	X	0	0	0	0	1	1	1	0	1	1	1	1	<table border="1"> <thead> <tr> <th>入力</th> <th>出力</th> </tr> <tr> <th>A</th> <th>X</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> </tr> </tbody> </table>	入力	出力	A	X	0	1	1	0
入力		出力																																													
A	B	X																																													
0	0	0																																													
0	1	0																																													
1	0	0																																													
1	1	1																																													
入力		出力																																													
A	B	X																																													
0	0	0																																													
0	1	1																																													
1	0	1																																													
1	1	1																																													
入力	出力																																														
A	X																																														
0	1																																														
1	0																																														

1) 特定の穴へのボールの入り方により勝敗が決まるゲームがある。AかBの穴にのみボールが入った場合には勝ちであるが、Cの穴に一つでもボールが入ってしまうと負けになる。入力A～Cはボールが入っていない場合には0、入っている場合には1とする。出力Xは勝ちである場合に1、負けである場合に0となる。これを実現する論理回路は次の図2である。

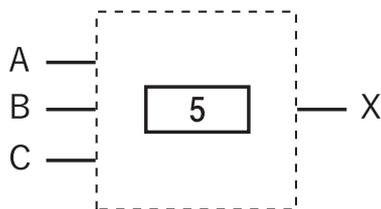


図2 1) の論理回路

2) 1) のゲームを少し複雑にした別のゲームがある。このゲームでも同様に、ある穴に一つでもボールが入ってしまうと負けになるとする。入力や出力の0と1は1)と同様とする。これを実現する論理回路は図3であり、図3において1)のCの穴と同様の役割をする穴は **6** である。

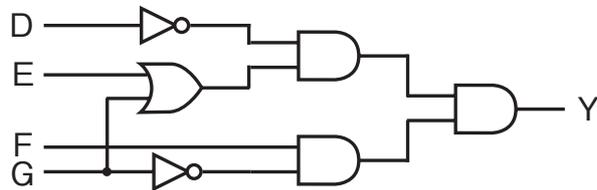
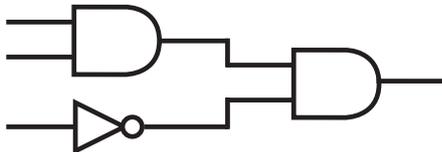


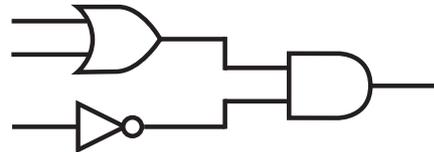
図3 2) の論理回路

5 の解答群

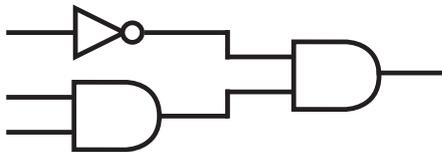
①



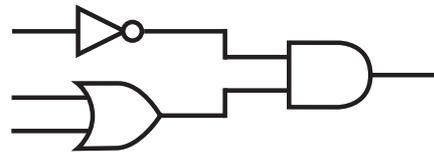
②



③



④



6 の解答群

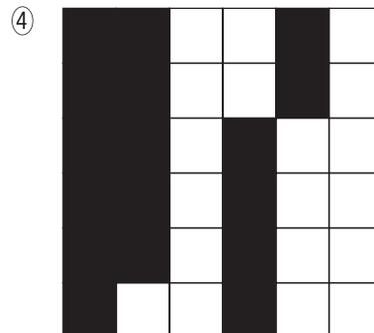
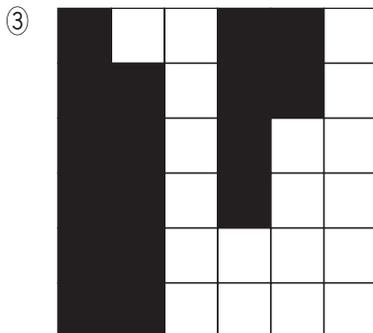
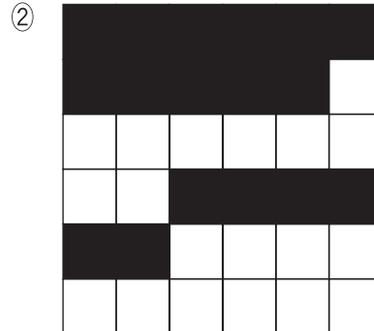
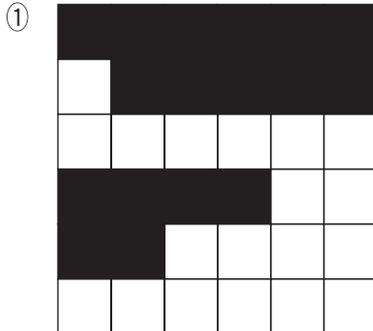
- | | | | |
|-------|-------|-------|-------|
| ① Dのみ | ② Eのみ | ③ Fのみ | ④ Gのみ |
| ⑤ DとE | ⑥ DとF | ⑦ DとG | ⑧ EとF |
| ⑨ EとG | ⑩ FとG | | |

- (5) 次の文章を読み、空欄 **7** ~ **10** に当てはまる数字をマークせよ。ただし、**8** **9** **10** は整数値であり、必要であれば小数第一位を四捨五入せよ。もし答えが1桁や2桁の場合は10や100の位を0とマークせよ（例：10であれば010、1であれば001）。3桁とも正解しないと点を与えない。また、空欄 **11** 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選べ。

データの可逆圧縮の方法の一つとして、ランレングス法がある。この圧縮法は、連続する同じ記号の列を、列の長さを示す数字で置き換える圧縮方式である。例として、「AAAAABBBBBBBBAAAA」は、Aが5個、Bが7個、Aが4個続いているので、「A5B**7**A4」と表すことができる。

ある6×6ビットの図の白の部分を0、黒の部分を1としてランレングス法で圧縮すると、「1 11 0 9 1 6 0 10」となった。圧縮後のデータ量が28ビットであったとすると、このデータの圧縮率は、約 **8** **9** **10** %である。このデータを展開したものは **11** である。ただし、圧縮は左上から右方向に進め、右端に来たら1つ下の行に進み、次の行も左から右方向に進めるものとする。展開は圧縮方法と同様に、6×6ビットに左上からおこなったものとする。

11 の解答群



II 次の問い (A・B) に答えよ。(35点)

[解答番号 12 ~ 18]

A 次の太郎さんと先生の会話文を読み、問い (1) ~ (4) に答えよ。

太郎：昨日学校からメールが届いたんですが、文字化けしてしまって読めませんでした。

先生：それは、12 と文字化けが起こってしまったりすることがありますね。

太郎：じゃあ、文字コードの種類を同じにすればいいので、文字コードの設定を変えてみたら解決できますか？

先生：そうですね。学校からのメールはHTMLメール形式で送信されたため、おそらく「UTF-8」という文字コードを使用しているはずですが、しかし、太郎さんのメールソフトウェアでは「a シフト JIS コード」などが使用されているので、そこで食い違いが起こっていますね。

太郎：設定を変えたら読めるようになりました。

b UTF-8は、Unicodeのエンコーディング方式のうちの一つですね。Unicodeは世界で使用されているほとんどの文字だけではなく絵文字なども表現できるんですよ。

先生：Unicodeで表現できる符号の組合せの数は111万4112個あるので、膨大な量の文字を表現することができます。この符号によって構成される集合を「符号空間」と言います。

太郎：それはすごい量ですね。

先生：また、日本語だけでも文字コードが数種類あるので、いろいろ調べてみたら面白いと思いますよ。

(1) 空欄 12 に当てはまる文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 文字をエンコードする際に、コンピュータが2進数に変換し始める位置を間違える
- ② 学校で使用した文字のフォントが特殊で、デコードする際に変換がうまくいかない
- ③ コンピュータでは2進数を使って文字を表すため、エンコードとデコードで文字コードが異なる
- ④ コンピュータでエンコードする際に文字を2進数で表し、デコードする際に16進数で表す

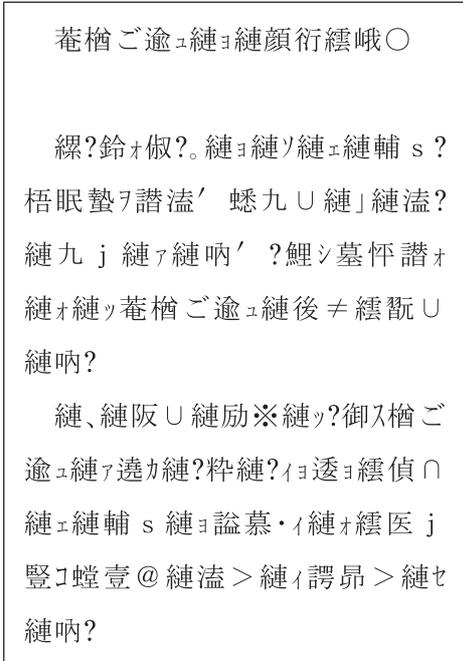


図1 文字化けしたメール

- (2) 下線部 **a** の文字コード表の一部を表 1 に示す。英文字「R」を 2 進数で表したものと
して最も適当なものを、後の①～⑥のうちから一つ選べ。 13

表 1 文字コード表の一部

		上の桁				
		2 進数	…	0100	0101	…
下の桁	2 進数	16 進数	…	4	5	…
	0000	0	…	@	P	…
	0001	1	…	A	Q	…
	0010	2	…	B	R	…
	0011	3	…	C	S	…
⋮	⋮	⋮	⋮	⋮	⋮	

13 の解答群

- ① 00100101 ② 25 ③ 01010010
④ 52 ⑤ 0010201015 ⑥ 0101500102

- (3) 下線部 **b** について、Unicode は文字をその符号空間に割り当てた符号化文字集合であり、
16 進数で符号空間を表している。これをそのままエンコードする方式は UTF-32 であるが
符号が全て 4 バイトである。一方、UTF-8 はアルファベットなら 1 バイト、ひらがななら
3 バイトで表すことができる。UTF-32 は Unicode のコードをそのまま反映しているが、
一般的には UTF-8 の方がよく使用される。その理由として、最も適当なものを次の①～
④のうちから一つ選べ。 14

- ① UTF-8 の方がバイト数が少なくすむので、データをやり取りする際にかかる時間が短
くなるから。
② UTF-32 はより正確に文字の形を表せるが、そこまで正確さは必要ないから。
③ UTF-8 は最初に開発されたので、より多くの人々がこちらの方式をコンピュータにダ
ウンロードしているから。
④ UTF-32 の方はコードの種類が Unicode の種類よりも多くなっているため、容量が大
きすぎるから。

- (4) UTF-16 は、UTF-8 のように文字の種類によってバイト数が増えるエンコード方式である。アルファベットなら 1 バイト、ひらがななら 3 バイトで表す UTF-8 と、アルファベットもひらがなも 2 バイトで表す UTF-16 でそれぞれエンコードするとき、合計バイト数を比較した結果として最も適当なものを後の①～④のうちから一つ選べ。ただし、カギかっこは文字列に含めない。

15

「mansikka はいちごのこと」

- ① UTF-8 も UTF-16 も同じ合計バイト数である。
- ② UTF-8 は UTF-16 より合計バイト数が 7 バイト多い。
- ③ UTF-8 は UTF-16 より合計バイト数が 1 バイト少ない。
- ④ UTF-8 は UTF-16 より合計バイト数が 8 バイト少ない。

B 次の文章を読み，後の問い（1）～（3）に答えよ。

Mさんはお風呂の自動保温機能に興味を持ったので，この現象がどのようなになっているか，図的モデルと数式モデルを用いて考えてみることにした。

（1） 次の文章中の空欄 **16** に当てはまる図的モデルとして最も適当なものを，後の①～④のうちから一つ選べ。

Mさんは，自分の家のお風呂の加温方式は強制循環方式であることを確認した。強制循環方式とは，一定量の水を吸い込み，それを温めて浴槽に戻すというものである。Mさんはこの方式が現在の水温と設定温度の温度差に比例して加温し，設定温度を保つと仮定した。また，ある時間間隔において，その時間間隔の始まりの時点での温度と終わりの時点での温度の変化を，現在の水温（その時間間隔の始まりの時点での温度）と設定温度との温度差に対する割合として表したものを「温度変化率」という言葉で定義した。

この現象を表す図的モデルを描いてみると **16** であった。ただし，図的モデル中の矢印は，影響を与える対象を示している。例えば，図1のaは影響を与える要素で，bは影響を受ける要素となる。

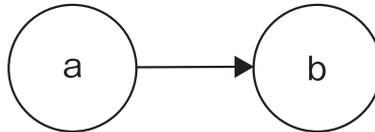
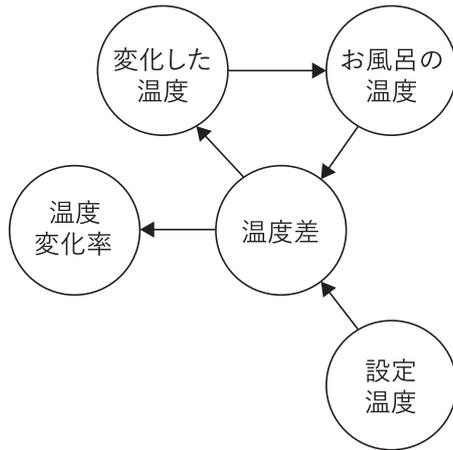
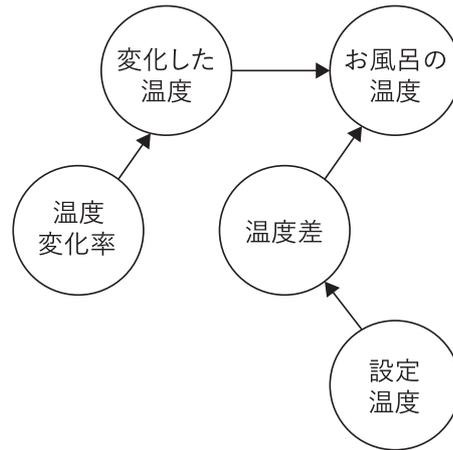


図1 図的モデルの例

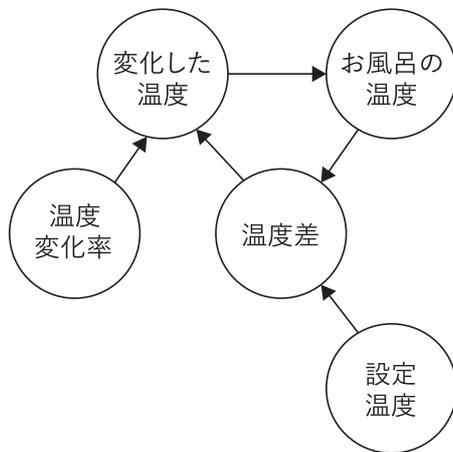
①



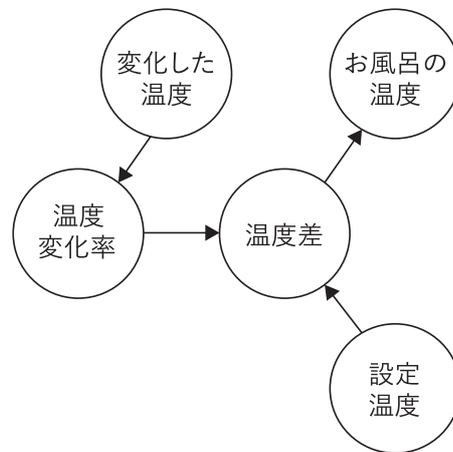
②



③



④



(2) 次の文章中の空欄 **17** に当てはまる数式として最も適当なものを、後の①～④のうちから一つ選べ。

一分間の温度変化率を 0.2 で変化しないものとする。初期のお風呂の水温を 30℃ とし、設定温度を 40℃ としたとき、ある一分間の区間の後のお風呂の水温を表す式は **17** であることがわかった。

- ① 変化後の水温 = $30 + (40 - 30) \times 0.2$
- ② 変化後の水温 = $30 + (\text{現在の水温} - 30) \times 0.2$
- ③ 変化後の水温 = $\text{現在の水温} + (40 - 30) \times 0.2$
- ④ 変化後の水温 = $\text{現在の水温} + (40 - \text{現在の水温}) \times 0.2$

- (3) 最初の水温が 30°C 、一分間の温度変化率が 0.2 (一定)、設定温度が 40°C であるときのグラフは図2のようになった。このグラフから読み取れないことを、後の①～④のうちから一つ選べ。 18

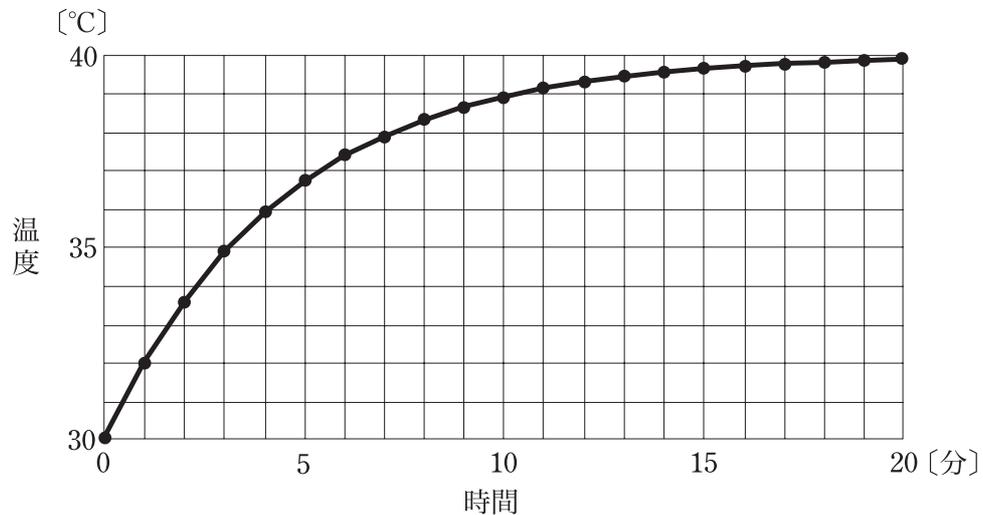


図2 自動保温機能による水温の変化

- ① 自動保温機能では、常に一定の熱量で加熱し続けているためエネルギー効率がよいと考えられる。
- ② 温度変化率が 0.2 より大きい場合、設定温度に到達するのに要する時間は短くなると考えられる。
- ③ 加温開始直後と設定温度近くでは加温開始直後の方がガスや電気などのエネルギーを多く使用していると考えられる。
- ④ 6分から7分の間に加温に使用された熱量は、加温開始から1分の間に加温に使用された熱量よりも少ない。

Ⅲ 次の問い(1)～(3)に答えよ。(40点)

[解答番号 19 ～ 26]

(1) 次の生徒(S)と先生(T)の会話文を読み、空欄 19 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選べ。

S：自分の家はイチゴ農家なので、収穫したイチゴを仕分けして出荷するのですが、手作業で仕分けなければいけないため、すごく時間がかかります。そのため、自動で仕分けをしてくれるような機械のプログラムを作りたいと思いました。

T：いいですね。それでは、どういう基準でイチゴを仕分けるのかを教えてください。

S：まず、サイズはS、M、Lの3種類あります。それから、傷があるかないかを見ます。傷があるものは売り物にできないからです。あと、色も見ます。全体が全て赤いのとムラのあるもので売値を変えるためです。

T：3種類の特徴を元にした仕分け方法があるわけですね。

S：では、これらの仕分けをプログラムに順番に書いていけばいいのですね。

T：ちょっと待ってください。プログラムで大事なことは効率性が高いアルゴリズムを用いることです。今回は外部の機械などを使用すると想定して、サイズは、重さをはかる機能を用い、傷と色はカメラ機能で判断するとします。そうすると、19 だと一番効率が悪くなってしまいますのわかりますか。

S：はい。そのアルゴリズムはイチゴを調べる手順が多いのと、機能を入れ替える回数も多いです。

T：そうですね。

S：一番効率がいい順番は、傷があるものを除外してから、サイズや色を仕分けしていく方法ですね。傷のチェックはカメラ機能を使うので、そのままカメラ機能を使う色の仕分けを2段階目にして、最後にサイズを仕分けるのがいいですね。

19 の解答群

- | | |
|-----------|-----------|
| ① サイズ→傷→色 | ② サイズ→色→傷 |
| ③ 色→傷→サイズ | ④ 色→サイズ→傷 |

(2) 次の文章や図中の空欄 **20** ～ **23** 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。

S：最も効率の良い仕分け方法を用いてプログラムを書いてみようと思うのですが、普通に書くと、仕分けの分岐も多いのですごく長くなってしまいます。

T：例えば、数式を用いてみるのはどうでしょう？算出された値の範囲で仕分けカテゴリーを決めるのです。

S：数式ですか？

T：難しく考えなくても大丈夫です。単純に全ての要素をある数値に置き換えてかけ算して出た値を用いてもよいのです。

S：なるほど。では、傷と色とサイズで何か数値を決めればいいのですね。サイズは単純に重さでいいと思います。

T：そうですね。傷があるものを10,000、ないものを1とすると、傷のあるものを全て取り除くことができますよ。

S：それでは、色は全体が赤のもの（良）を1、ムラがあるもの（可）を100とすれば、分けることができますね。

Sさんが先生（T）との会話からヒントを得て、変数 `bunrui` の数式をたてたところ **20** になった。また、イチゴにそれぞれ番号を振り、その番号を配列 `Ichigo` に格納することにした。このとき、調べるイチゴの総数を `n` とした。

イチゴを重さで仕分ける際の数値は表1のようにし、重さの範囲は左の数値を含み、右の数値は含まないこととした。また、規格外のサイズのイチゴはないと仮定した。

Sさんは図1のようなプログラムを作成した。

表1 イチゴのサイズと重さの関係

サイズ	S	M	L
重さ [g]	1 ~ 12	12 ~ 19	19 ~ 31

- (1) Ichigo = [1, ..., n]
- (2) i を **21** 繰り返す :
- (3) | kizu = 【外部 (AI カメラ) からの入力】
- (4) | iro = 【外部 (AI カメラ) からの入力】
- (5) | omosa = 【外部 (はかり) からの入力】
- (6) | bunrui = **20**
- (7) | もし bunrui >= 1 and bunrui < 12 ならば :
- (8) | | 表示する ("イチゴ", Ichigo[i], "は S 良")
- (9) | そうでなくもし bunrui < 19 ならば :
- (10) | | 表示する ("イチゴ", Ichigo[i], "は M 良")
- (11) | そうでなくもし **22** ならば :
- (12) | | 表示する ("イチゴ", Ichigo[i], "は L 良")
- (13) | そうでなくもし bunrui >= 100 and bunrui < 1200 ならば :
- (14) | | 表示する ("イチゴ", Ichigo[i], "は S 可")
- (15) | そうでなくもし bunrui < 1900 ならば :
- (16) | | 表示する ("イチゴ", Ichigo[i], "は M 可")
- (17) | そうでなくもし **23** ならば :
- (18) | | 表示する ("イチゴ", Ichigo[i], "は L 可")
- (19) | そうでなければ :
- (20) | | 表示する ("イチゴ", Ichigo[i], "は出荷不可")

図1 イチゴの仕分けのプログラム

20 の解答群

- ① kizu ② iro ③ omosa
- ④ kizu * iro ⑤ kizu * omosa ⑥ iro * omosa
- ⑦ kizu * iro * omosa

21 の解答群

- ① 0 から n まで 1 ずつ増やしなが
- ② 0 から n まで 1 ずつ減らしなが
- ③ 0 から n + 1 まで 1 ずつ増やしなが
- ④ 0 から n + 1 まで 1 ずつ減らしなが
- ⑤ 0 から n - 1 まで 1 ずつ増やしなが
- ⑥ 0 から n - 1 まで 1 ずつ減らしなが

22 ・ 23 の解答群

- ① `bunrui > 19 and bunrui < 31`
- ② `bunrui < 19 or bunrui > 31`
- ③ `bunrui < 31`
- ④ `bunrui > 1900 and bunrui < 3100`
- ⑤ `bunrui < 1900 or bunrui > 3100`
- ⑥ `bunrui < 3100`

(3) 次の文章や図中の空欄 24 ・ 26 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 25 にあてはまる数字をマークせよ。

T：プログラム（図1）ができたようですね。

S：もしこれに箱詰め機能がついたら仕分け作業はもっと楽になると思います。箱詰めプログラムを作りたいのですが、箱とみなした別の配列にイチゴの番号を振り分けていくことができるのでしょうか？

T：関数を利用してみるのはどうでしょう？指定する配列の最後尾に格納したいデータなどを追加する関数「指定する配列名.追加する()」がありますよ。

【関数の説明】

指定する配列名.追加する()…格納したいデータを()内に入力すると、指定する配列の最後尾に追加される関数。例えば、配列 `Gusu = [2, 4, 6]` に 8 を追加したい場合、
`Gusu.追加する(8)`
を実行すると、
`Gusu = [2, 4, 6, 8]`
となる。

Sさんは、配列 `HakoS`, `HakoM`, `HakoL` にイチゴの番号を格納するようにした。また、イチゴ1箱は9個入りであるとし、配列 `Kazu` にそれぞれの分類の箱の空き個数を格納し、配列 `Kazu` の添字に対応する変数を `j` とした。

Sさんは簡易的なプログラムを最初に作成して確かめることにしたため、今回のプログラムは傷があるイチゴと色ムラがあるイチゴがないものとして作成した（図2）。また、配列 `Ichigo` には、サイズ `S` のイチゴの番号が9個、サイズ `M` のイチゴの番号が9個、サイズ `L` のイチゴの番号が9個ちょうど格納されているとする。なお、`n` には27が当てはまるが、問題の都合上 `n` のままにしてある。

```

(1) Ichigo = [1, ..., n]
(2) HakoS = [], HakoM = [], HakoL = []
(3) Kazu = [9, 9, 9]
(4) i を  繰り返す :
(5) | omosa = 【外部 (はかり) からの入力】
(6) | bunrui = 
(7) | もし bunrui >= 1 and bunrui < 12 ならば :
(8) | | j = 0
(9) | | もし Kazu[j] >  ならば :
(10) | | | Kazu[j] = 
(11) | | | HakoS.追加する (Ichigo[i])
(12) | | そうでなければ :
(13) | | | HakoS.追加する (Ichigo[i])
(14) | | | 表示する ("箱 S 詰め終わり")
(15) | そうでなくもし bunrui < 19ならば :
(16) | | j = 1
(17) | | もし Kazu[j] >  ならば :
(18) | | | Kazu[j] = 
(19) | | | HakoM.追加する (Ichigo[i])
(20) | | そうでなければ :
(21) | | | HakoM.追加する (Ichigo[i])
(22) | | | 表示する ("箱 M 詰め終わり")
(23) | そうでなくもし  ならば :
(24) | | j = 2
(25) | | もし Kazu[j] >  ならば :
(26) | | | Kazu[j] = 
(27) | | | HakoL.追加する (Ichigo[i])
(28) | | そうでなければ :
(29) | | | HakoL.追加する (Ichigo[i])
(30) | | | 表示する ("箱 L 詰め終わり")

```

図2 箱詰めのプログラム

24 の解答群

- ① kizu
- ② iro
- ③ omosa
- ④ kizu * iro
- ⑤ kizu * omosa
- ⑥ iro * omosa
- ⑦ kizu * iro * omosa

26 の解答群

- ① Kazu[j] - 1
- ② Kazu[j]
- ③ Kazu[j] + 1
- ④ j - 1
- ⑤ j
- ⑥ j + 1

IV 次の問い(1)～(5)に答えよ。(30点)

[解答番号 27 ～ 31]

次の表1は、国が実施した統計調査をもとに、全都道府県別の外国人居住者とその地域の住民の外国語受講者割合をまとめたものの一部である。

表1 都道府県別の外国人居住者と外国語受講者割合

都道府県	外国人居住者〔人〕	外国語受講者〔%〕
北海道	36,316	11.1
青森	5,693	7.4
岩手	7,203	7.7
宮城	21,089	12.4
沖縄	18,535	14.4

(法務省と総務省統計局の2021年のデータをもとに作成)

Nさんは、近年増加傾向にある外国人居住者の数と、日本人の外国語受講者の割合に相関があるのかどうか調べてみることにした。

(1) Nさんはこれらのデータから次のような仮説を考えた。表1のデータだけを用いて分析できない仮説を、次の①～④のうちから一つ選べ。 27

- ① 外国人居住者が多い順と、外国語受講者割合が大きい順に並べると似たような順番になるのではないか。
- ② 首都圏にある都県と九州地方にある県を比較すると、首都圏にある都県の方が外国語受講者割合が大きいのではないか。
- ③ インドネシア人居住者が多い都道府県では、インドネシア語受講者の割合も多いのではないか。
- ④ 外国人居住者の最大値を保有する都道府県と、外国語受講者の最大値を保有する都道府県は一致するのではないか。

Nさんは外国語受講者の割合を箱ひげ図（外れ値は○で表記，平均値は×で表記）で表してみることにした。

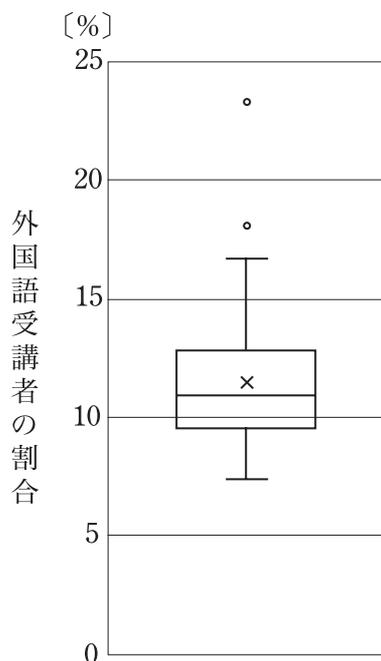


図1 都道府県別の外国語受講者の割合の箱ひげ図

(2) 図1から読み取れることとして，最も適当なものを次の①～④のうちから一つ選べ。

28

- ① 外国語受講者の割合が10%から15%までの間の範囲には，45の都道府県が分布している。
- ② 外国語受講者の割合が多い上位半分の都道府県は，下位半分と比べてデータの散らばりが小さい。
- ③ 外国語受講者の割合の平均値と中央値を比べると，平均値は中央値よりも小さくなっている。
- ④ 第3四分位数と第1四分位数の差は，最小値と第1四分位数の差よりも大きくなっている。

Nさんは、外国人居住者と外国語受講者の相関を見るには、外国人居住者の人数ではなく、人口に対する割合にして外国語受講者の割合と比較しなければいけないことに気づいた。そこで、各都道府県の総人口を調べ外国人居住者の割合を出し、外国語受講者の割合とともに散布図に示した。なお、この散布図には、完全に重なっている点はない。

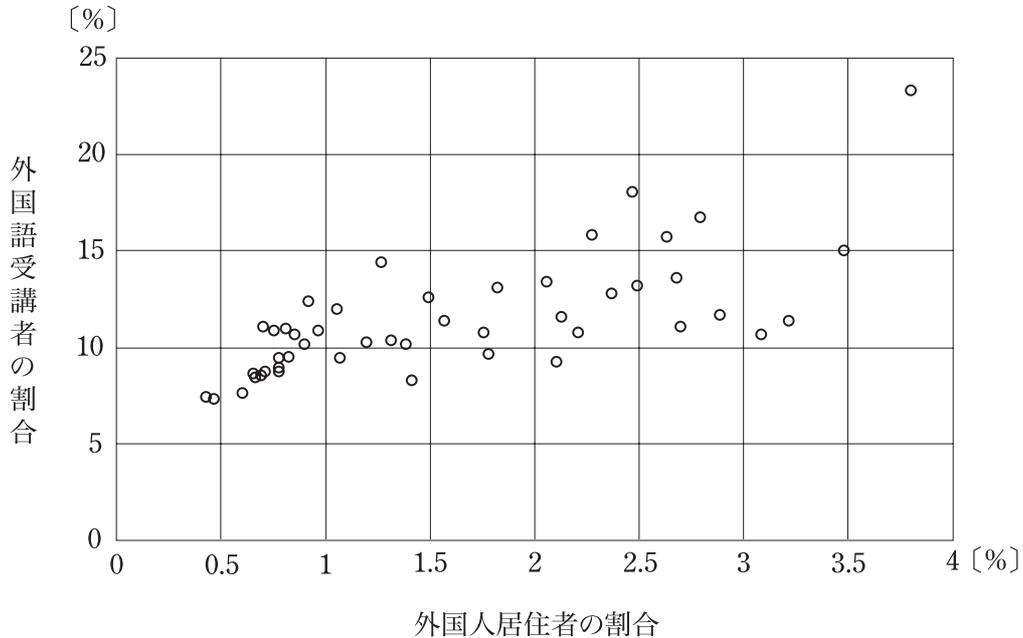


図2 外国人居住者の割合と外国語受講者の割合の散布図

相関係数 0.71

(3) 図2から読み取れることとして、最も適当なものを次の①～④のうちから一つ選べ。

29

- ① 外国人居住者の割合が増えるにつれて、外国語受講者の割合が減る傾向にある。
- ② 外国人居住者の割合が増えるにつれて、外国語受講者の割合も増える傾向にある。
- ③ 外国人居住者の割合が1%未満のデータと1%以上のデータの散らばりの度合いは1%以上のデータの方が小さい。
- ④ 外国語受講者の割合が10%未満のデータと10%以上のデータの散らばりの度合いは10%未満のデータの方が大きい。

ある2つの変数の間に相関関係があるとき、これらの変数を用いて変数間の関係を式で表すことができる。この関係式を直線で表したものを回帰直線という。また、回帰直線の式の実際のデータに対する精度を表したものを決定係数という。この係数は1に近いほど精度が高く、一般には0.5以上であると精度が高い部類に入る。

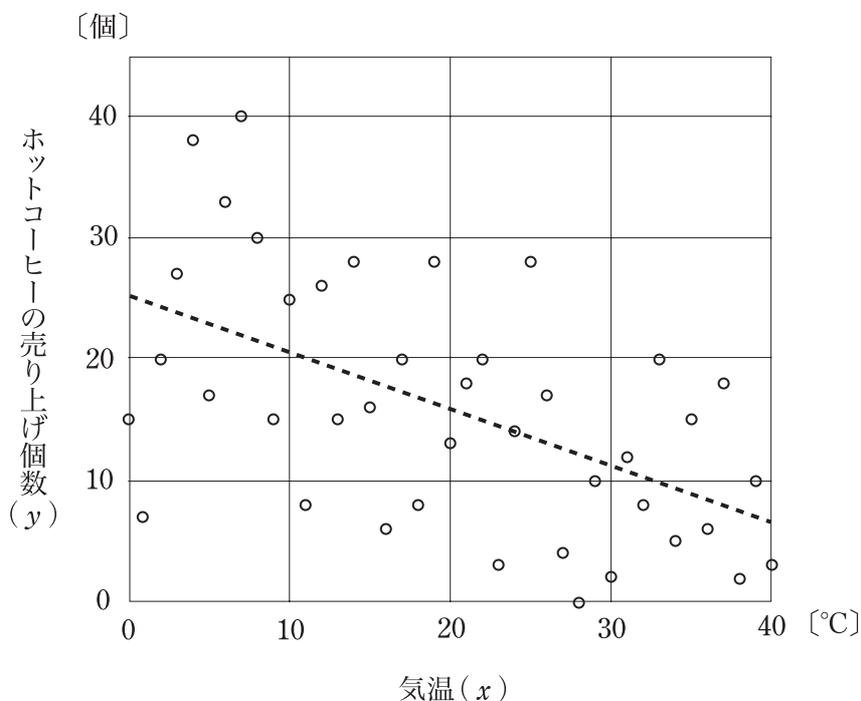


図3 回帰直線の例

回帰直線の式 $y = -0.47x + 25$

決定係数 0.30

(4) 図3の回帰直線から読み取れることとして、最も適当なものを次の①～④のうちから一つ選べ。なお、この散布図には、完全に重なっている点はない。 30

- ① 直線の傾きが右下がりであるので、強い負の相関がある。
- ② この回帰直線の式を用いても、気温からの売り上げ個数の推定値の精度は高くない。
- ③ データと回帰直線との距離が小さいものは、外れ値である。
- ④ 回帰直線の上に乗っている点は、売り上げ個数と気温の中央値である。

Nさんは、図2の散布図の回帰直線を求め、図4に表した。

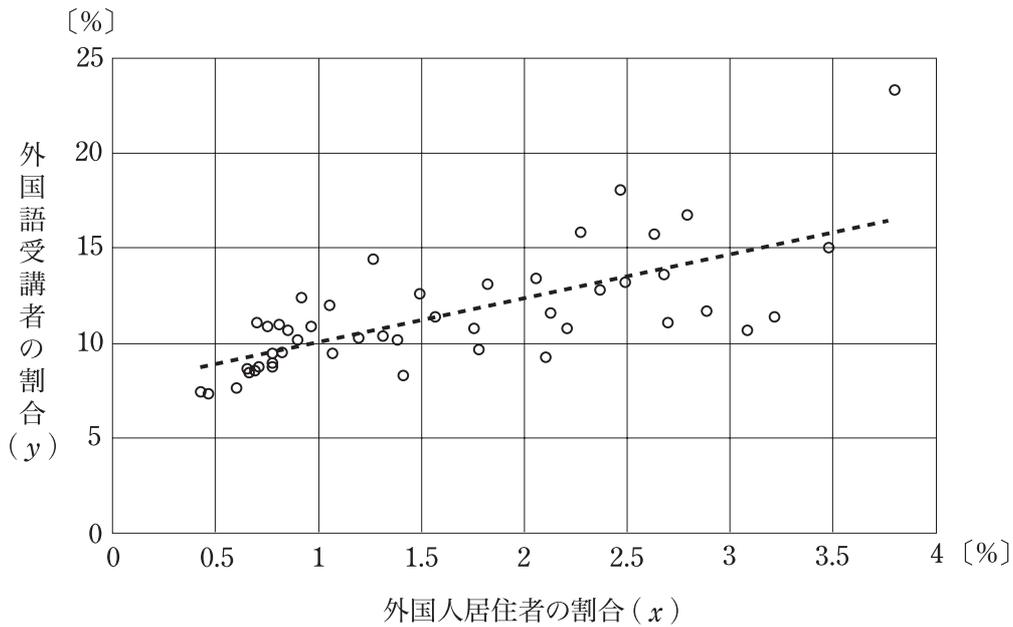


図4 回帰直線をかき加えた外国人居住者の割合と外国語受講者の割合の散布図

回帰直線の式 $y = 2.3x + 7.7$

決定係数 0.50

(5) 図4から読み取れることとして、最も適当なものを次の①～④のうちから一つ選べ。

31

- ① この散布図から算出される回帰直線の式の精度は高くない。
- ② 回帰直線の式の y は各都道府県の外国人居住者の割合を表す変数である。
- ③ 外国人居住者の割合が3%を超える都道府県では、回帰直線よりも上側に分布することはない。
- ④ 外国人居住者と外国語受講者の割合が両方とも小さい地域は、大きい地域と比べ散らばりが小さい。