

B5サイズで
作成しています

令和7年度大学入学共通テスト 試作問題『情報Ⅰ』

〔100点〕

- 試験時間 60分
- 出題範囲 「情報Ⅰ」の内容から出題
- 作成の趣旨及び留意点

本試作問題は、令和7年度大学入学共通テストから新たに出題科目として設定する『情報Ⅰ』について具体的なイメージの共有のために作成・公表するものです。

本試作問題は専門家により作成されたものですが、過去の大学入試センター試験や大学入学共通テストと同様の問題作成や点検のプロセスを経たものではありません。

なお、令和7年度大学入学共通テストの出題内容については、本試作問題の作成を踏まえつつ、引き続き検討することとしています。

※ 本試作問題に関する説明は、「試作問題「情報」の概要」を御覧ください。

○ 解 答

- 1 解答は、解答用紙の問題番号に対応した解答欄にマークしなさい。
第 2 問の **ア** と表示のある問いに対して **③** と解答する場合は、次の例のように問題番号 **2** の解答記号 **ア** の解答欄の **③** にマークしなさい。

例 1

2	解 答 欄										
	0	1	2	3	4	5	6	7	8	9
ア	①	②	③	④	⑤	⑥	⑦	⑧	⑨	

- 2 問題の文中の **イ** **ウ** などの に数字 (0~9) 又は文字 (a~d) を入れるよう指示された場合、次の方法で解答用紙の指定欄に解答しなさい。

例 2 **イ** **ウ** に 38 と答えたいとき

イ	①	②	③	④	⑤	⑥	⑦	⑧	⑨
ウ	①	②	③	④	⑤	⑥	⑦	⑧	⑨

なお、同一の問題文中に、**ア** , **イ** **ウ** などが 2 度以上現れる場合、原則として、2 度目以降は、 **ア** , **イ** **ウ** のように細字で表記します。

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 インターネットを使ったサービス利用に関する次の問い(a・b)に答えよ。

a SNS やメール, Web サイトを利用する際の注意や判断として, 適当なものを, 次の ①~⑤ のうちから二つ選べ。ただし, 解答の順序は問わない。

ア . イ

- ① 相手からのメッセージにはどんなときでも早く返信しなければいけない。
- ② 信頼関係のある相手と SNS やメールでやり取りする際も, 悪意を持った者になりすましている可能性を頭に入れておくべきである。
- ③ Web ページに匿名で投稿した場合は, 本人が特定されることはない。
- ④ SNS の非公開グループでは, どんなグループであっても, 個人情報を書き込んでも問題はない。
- ⑤ 一般によく知られているアニメのキャラクターの画像を SNS のプロフィール画像に許可なく掲載することは, 著作権の侵害にあたる。
- ⑥ 芸能人は多くの人に知られていることから肖像権の対象外となるため, 芸能人の写真を SNS に掲載してもよい。

b インターネット上の情報の信ぴょう性を確かめる方法として、最も適切なものを次の①～③のうちから一つ選べ。

- ① 検索エンジンの検索結果で、上位に表示されているかどうかで判断する。
- ② Q&A サイトの回答は、多くの人に支持されているベストアンサーに選ばれているかどうかで判断する。
- ③ SNS に投稿された情報は、共有や「いいね」の数が多いかどうかで判断する。
- ④ 特定の Web サイトだけでなく、書籍や複数の Web サイトなどを確認し、比較・検証してから判断する。

問2 次の文章の空欄 **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。

データの通信において、受信したデータに誤りがないか確認する方法の一つにパリティチェックがある。この方法では、データにパリティビットを追加してデータの誤りを検出する。ここでは、送信データの1の個数を数えて、1の個数が偶数ならパリティビット0を、1の個数が奇数ならパリティビット1を送信データに追加して通信することを考える。例えば、図1に示すように送信データが「01000110」の場合、パリティビットが1となるため、パリティビットを追加したデータ「010001101」を送信側より送信する。

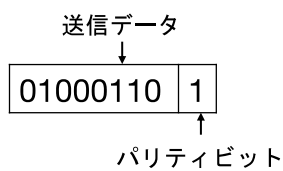


図1 送信データ「01000110」とパリティビット

受信側では、データの1の個数が偶数か奇数かにより、データの通信時に誤りがあったかどうかを判定できる。この考え方でいくと、**エ**。

例えば、16進法で表記した「7A」を2進法で8ビット表記したデータに、図1と同様にパリティビットを追加したデータは、「**オ**」となる。

エ の解答群

- ① パリティビットに誤りがあった場合は、データに誤りがあるかどうかを判定できない
- ② パリティビットを含め、一つのビットの誤りは判定できるが、どのビットに誤りがあるかは分からない
- ③ パリティビットを含め、一つのビットの誤りは判定でき、どのビットに誤りがあるかも分かる
- ④ パリティビットを含め、二つのビットの誤りは判定できるが、どのビットに誤りがあるかは分からない
- ⑤ パリティビットを含め、二つのビットの誤りは判定でき、どのビットに誤りがあるかも分かる



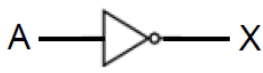
オ の解答群

- | | | |
|-------------|-------------|-------------|
| ① 011110100 | ② 011110101 | ③ 011110110 |
| ④ 011110111 | ⑤ 101001110 | ⑥ 101001111 |

問3 次の文章を読み、空欄 **カ** ～ **ク** 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。

基本的な論理回路には、論理積回路（AND 回路）、論理和回路（OR 回路）、否定回路（NOT 回路）の三つがあげられる。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、入力と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) S 航空会社が所有する旅客機の後方には、トイレが二つ ($A \cdot B$) ある。トイレ A とトイレ B の両方が同時に使用中になると乗客の座席前にあるパネルのランプが点灯し、乗客にトイレが満室であることを知らせる。入力 A は、トイレ A が使用中の場合には 1、空いている場合には 0 とする。B についても同様である。出力 X はランプが点灯する場合に 1、点灯しない場合に 0 となる。これを実現する論理回路は次の図 2 である。

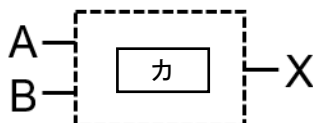


図 2 (1)の論理回路

(2) S航空会社では新しい旅客機を購入することにした。この旅客機では、トイレを三つ(A・B・C)に増やし、三つのうちどれか二つ以上が使用中になったら混雑を知らせるランプを点灯させる。入力や出力は(1)と同様とする。この場合の真理値表は **キ** で、これを実現する論理回路は図3である。

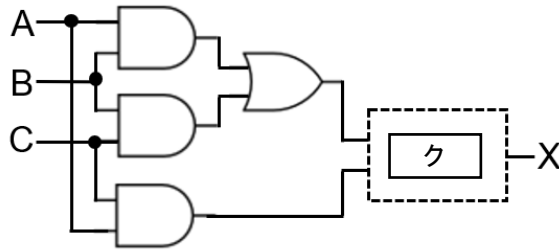
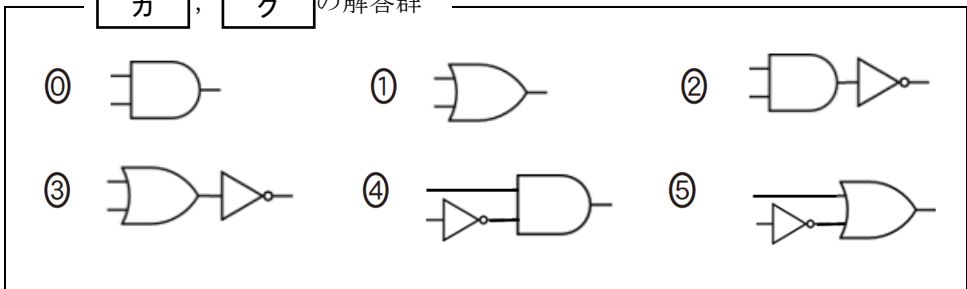


図3 (2)の論理回路

カ, **ク** の解答群



キ の解答群

①	②	③	④																																																																																																																																																																
<table border="1" style="width: 100%; border-collapse: collapse; text-align: left;"> <thead> <tr> <th colspan="3">入力</th> <th>出力</th> </tr> <tr> <th>A</th> <th>B</th> <th>C</th> <th>X</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	入力			出力	A	B	C	X	0	0	0	0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0	0	1	0	1	0	1	1	0	0	1	1	1	1	<table border="1" style="width: 100%; border-collapse: collapse; text-align: left;"> <thead> <tr> <th colspan="3">入力</th> <th>出力</th> </tr> <tr> <th>A</th> <th>B</th> <th>C</th> <th>X</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	入力			出力	A	B	C	X	0	0	0	0	0	0	1	1	0	1	0	1	0	1	1	0	1	0	0	1	1	0	1	0	1	1	0	0	1	1	1	1	<table border="1" style="width: 100%; border-collapse: collapse; text-align: left;"> <thead> <tr> <th colspan="3">入力</th> <th>出力</th> </tr> <tr> <th>A</th> <th>B</th> <th>C</th> <th>X</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	入力			出力	A	B	C	X	0	0	0	0	0	0	1	0	0	1	0	0	0	1	1	1	1	0	0	0	1	0	1	1	1	1	0	1	1	1	1	1	<table border="1" style="width: 100%; border-collapse: collapse; text-align: left;"> <thead> <tr> <th colspan="3">入力</th> <th>出力</th> </tr> <tr> <th>A</th> <th>B</th> <th>C</th> <th>X</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	入力			出力	A	B	C	X	0	0	0	0	0	0	1	1	0	1	0	1	0	1	1	1	1	0	0	1	1	0	1	1	1	1	0	1	1	1	1	1
入力			出力																																																																																																																																																																
A	B	C	X																																																																																																																																																																
0	0	0	0																																																																																																																																																																
0	0	1	0																																																																																																																																																																
0	1	0	0																																																																																																																																																																
0	1	1	0																																																																																																																																																																
1	0	0	0																																																																																																																																																																
1	0	1	0																																																																																																																																																																
1	1	0	0																																																																																																																																																																
1	1	1	1																																																																																																																																																																
入力			出力																																																																																																																																																																
A	B	C	X																																																																																																																																																																
0	0	0	0																																																																																																																																																																
0	0	1	1																																																																																																																																																																
0	1	0	1																																																																																																																																																																
0	1	1	0																																																																																																																																																																
1	0	0	1																																																																																																																																																																
1	0	1	0																																																																																																																																																																
1	1	0	0																																																																																																																																																																
1	1	1	1																																																																																																																																																																
入力			出力																																																																																																																																																																
A	B	C	X																																																																																																																																																																
0	0	0	0																																																																																																																																																																
0	0	1	0																																																																																																																																																																
0	1	0	0																																																																																																																																																																
0	1	1	1																																																																																																																																																																
1	0	0	0																																																																																																																																																																
1	0	1	1																																																																																																																																																																
1	1	0	1																																																																																																																																																																
1	1	1	1																																																																																																																																																																
入力			出力																																																																																																																																																																
A	B	C	X																																																																																																																																																																
0	0	0	0																																																																																																																																																																
0	0	1	1																																																																																																																																																																
0	1	0	1																																																																																																																																																																
0	1	1	1																																																																																																																																																																
1	0	0	1																																																																																																																																																																
1	0	1	1																																																																																																																																																																
1	1	0	1																																																																																																																																																																
1	1	1	1																																																																																																																																																																

問4 次の文を読み、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。ただし、空欄 **コ**・**サ** は解答の順序は問わない。

情報を整理して表現する方法として、アメリカのリチャード・S・ワーマンが提唱する「究極の5つの帽子掛け」というものがある。これによれば、情報は無限に存在するが、次の5つの基準で情報の整理・分類が可能という。

- ・場所・・・物理的な位置を基準にする
例：都道府県の人口，大学のキャンパスマップ
- ・アルファベット・・・言語的な順番を基準にする（日本語なら五十音）
例：辞書，電話帳
- ・時間・・・時刻の前後関係を基準にする
例：歴史年表，スケジュール
- ・カテゴリー・・・物事の差異により区別された領域を基準にする
例：生物の分類，図書館の本棚
- ・階層（連続量）・・・大小や高低など数量的な変化を基準にする
例：重要度順のToDoリスト，ファイルサイズの大きい順

この基準によれば、図4の「鉄道の路線図」は **ケ** を基準にして整理されており、図5のある旅行会社のWebサイトで提供されている「温泉がある宿の満足度評価ランキング」は **コ** と **サ** を基準に整理・分類されていると考えられる。

ケ ～ **サ** の解答群

① 場所	② アルファベット	③ 時間
④ カテゴリー	⑤ 階層（連続量）	

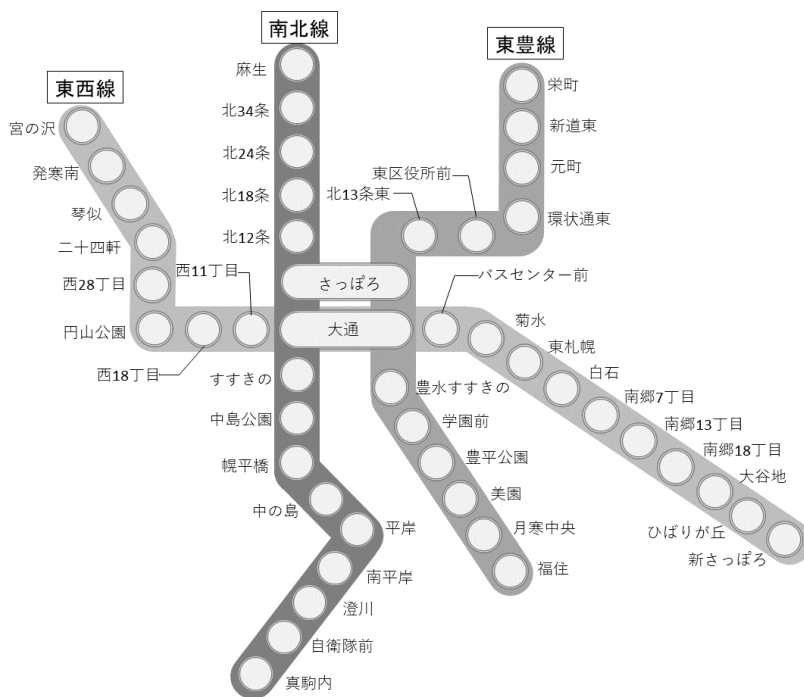


図4 鉄道の路線図

○△※旅行社

○△※旅行社 ホテル・旅館 満足度評価ランキング

リゾートホテル

シティホテル

温泉がある宿

食事が人気の宿

隠れ宿

ペットが泊まれる宿

温泉がある宿の満足度評価ランキング 1~10 位
前へ 次へ

順位	総合評価	ホテル・旅館(宿泊プラン)
1位	★★★★★	長野 △△温泉 ○○○館 1泊2日 ¥19,800
2位	★★★★★	神奈川 ◇◇温泉 ホテル△△△ 1泊2日 ¥25,000
3位	★★★★☆	群馬 ▽▽温泉 湯宿☆☆☆ 1泊2日 ¥19,500
4位	★★★★☆	大分 □□温泉 ◎◎◎館 1泊2日 ¥21,400
5位	★★★★☆	秋田 ○○温泉 ▽▽▽旅館 1泊2日 ¥18,800
6位	★★★★☆	愛媛 ☆☆☆温泉 □□□ホテル 1泊2日 ¥15,800

図5 温泉がある宿の満足度評価ランキング

第2問 次の問い（A・B）に答えよ。（配点 30）

A 次の太郎さんと先生の会話文を読み、問い（問1～4）に答えよ。

太郎：二次元コードって様々なところで使われていて、便利ですね。

先生：二次元コードといってもいろいろ種類があるけれど、日ごろよく目にするものは日本の企業が考えたんだよ。

太郎：すごい発明ですね。企業だから特許を取ったのでしょうか。

先生：もちろん。 ア 世の中で広く使われるようになったんだよ。



図1 二次元コードの例

太郎：どのくらいの情報を入れられるのでしょうか。

先生：大きさにもよるけど、図1 ぐらいの大きさであれば、数字なら187文字、英小文字なら78文字、記号や漢字なら48文字を入れられるよ。二次元コードの形状にはどんな特徴があるかな？

太郎：黒白の小さな正方形で構成されていて、3か所の隅に二重の少し大きな正方形がありますね。

先生：黒白の小さな正方形はセルと言って、1と0に符号化されるんだよ。図1の二次元コードは縦×横が33×33のセルで構成されているけど、文字種や文字数などによってセルの縦と横の数が変わり、それにつれて二次元コードの大きさも変わるね。A 3か所の隅にある二重の少し大きな正方形は、読み取り機にこの二次元コードがあることを教えている位置検出の目印なんだ。

太郎：この二次元コードって一部を隠しても正しく読み取れるんですね。

先生：B 誤り訂正機能だね。工場などでの製品管理でも使えるように、汚れや破損などで一部が読み取れなくても復元できるんだよ。読み取れない面積の割合によって復元できるレベルは4段階あるんだ。

太郎：すごい技術ですね。

先生：そうだね。自分でも二次元コードを作成できるから、いろいろ試してみたらどうかな。

問1 空欄 **ア** に当てはまる文として最も適当なものを、次の①～③のうちから一つ選べ。

- ① そこで、使用料を高くすることでこの二次元コードの価値が上がったから
- ② しかし、その後特許権を放棄して誰でも特許が取れるようにしたから
- ③ そして、特許権を行使して管理を厳密にしたから
- ④ でも、特許権を保有していても権利を行使しないとしていたから

問2 下線部Aの目印は、図2のように、例えば(a)～(c)のどの角度で読み取っても、黒白黒白黒の比が1:1:3:1:1となることで、二次元コードの目印として認識できるようになっている。これは、図3のように円形の目印でも同じと考えられるが、正方形の方が都合がよい。その理由として最も適当なものを、後の①～③のうちから一つ選べ。 **イ**

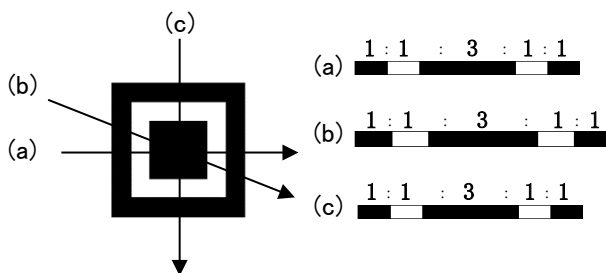


図2 位置検出の目印とその黒白の比

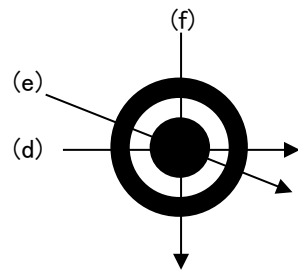


図3 円形の目印

- ① 円形では、(d)～(f)の角度によって黒白の比が異なってしまい、正しく読み取れなくなる可能性があるから。
- ② 円形だと上下左右がないので、二次元コードの向きが分からなくなるから。
- ③ プリンタやディスプレイの解像度によっては、正方形の目印に比べて正しく読み取れる小さな円形の目印を作ることが難しくなるから。
- ④ 円形では目印が斜めに傾いていても、それを認識することができないため正しく読み取ることができないから。

問3 太郎さんは、先生から二次元コードを作成することができる図4のようなWebアプリケーションを教えてもらった。この二次元コード画像作成ツールは、二次元コード化する文字列とセルのサイズ(大きさ)、誤り訂正のレベル(復元能力)、画像ファイル形式を指定すると二次元コードの画像が作成できるものであった。

二次元コード画像作成ツール

1 コード化する文字列

abcdefghijklmno

2 セルサイズ(大きさ)

小 中 大 特大

3 誤り訂正のレベル(復元能力)

レベルL (7%) レベルM (15%) レベルQ (25%) レベルH (30%)

4 画像ファイル形式

PNG形式(推奨) JPG形式 GIF形式









二次元コード画像の保存

図4 二次元コード画像作成ツールの画面

下線部Bについて、興味を持った太郎さんは、この作成ツールを使い、二次元コード化する文字列の長さや誤り訂正のレベルによってどのようにセルの縦と横の数が増えるか調べることにした。そこで、試しに英小文字(a~z)で構成する文字列の文字数をいろいろ変えて二次元コードを作成したところ、表1のようになった。表中のn×nはそれぞれセルの縦と横の数を表している。

なお、この作成ツールではセルの縦と横の数は自動的に最適な数に調整される。また、復元能力の値(%)が大きいほど誤りを訂正する能力が高いことを表し、例えば、復元能力30%は、二次元コードの面積の最大30%が読み取れなくてもデータを復元できることを意味する。

表1 英小文字のみで構成された文字列の文字数と復元能力を変えて作成した二次元コード

	15 文字	20 文字	30 文字	40 文字
復元能力 7 %	 21×21	 25×25	 25×25	 29×29
復元能力 30 %	 29×29	 29×29	 33×33	 37×37

この表1の結果から考えられることとして適当なものを、次の①～⑤のうちから二つ選べ。ただし、解答の順序は問わない。 ウ・エ



- ① 同じ復元能力であれば、文字数に比例してセルの数が多くなり、同じセルの大きさであれば二次元コードも大きくなる。
- ② 復元能力ごとに、文字数の一定の範囲でセルの縦と横の数が決まり、文字数が多くなるほど段階的にセルの縦と横の数は多くなる。
- ③ 文字数とセルの数には関係が見られない。
- ④ ある文字列を復元能力 30%で作成した二次元コードは、同じ文字列を復元能力 7%で作成したものに比べ約 4 倍のセルの数がある。
- ⑤ 復元能力 30%にするためには、復元能力 7%と比べより多くの情報が必要となる。
- ⑥ 同じ文字数であれば復元能力を変えてもセルの数は変わらない。

問4 次に、太郎さんは、図4のWebアプリケーションを使って試しに表2のⅠ～Ⅲの三つの文字列について二次元コードを作成してみた。復元能力は7%と30%の両方を作成し、セルサイズもいろいろ変えてみたところ、表3に示す二次元コードが作成された。その結果、Ⅰ～Ⅲの文字列はアルファベットや記号、漢字などが含まれているので、表1の英小文字のみで構成された文字列の文字数とセルの縦と横の数の関係には必ずしもなっていないことが分かった。表3の空欄 ～ に当てはまる適当な二次元コードを、後の解答群のうちから一つずつ選べ。

表2 二次元コードを作成した文字列

Ⅰ	https://www.example.ne.jp/
Ⅱ	DNC高等学校 https://www.example.ne.jp/
Ⅲ	DNC高等学校 東京都目黒区駒場*~*~*~* https://www.example.ne.jp/

表3 Ⅰ～Ⅲの文字列から作成された二次元コード

Ⅰの二次元コード 復元能力7% <input type="text" value="オ"/>	Ⅱの二次元コード 復元能力7% 29×29 	Ⅲの二次元コード 復元能力7% <input type="text" value="カ"/>
Ⅰの二次元コード 復元能力30% 33×33 	Ⅱの二次元コード 復元能力30% <input type="text" value="キ"/>	Ⅲの二次元コード 復元能力30% <input type="text" value="ク"/>

オ ~ ク の解答群

① 33×33



② 49×49



③ 25×25



④ 37×37



B 次の文章を読み、後の問い（問1～3）に答えよ。

Mさんのクラスでは、文化祭の期間中2日間の日程でクレープを販売することにした。1日目は、慣れないこともあり、客を待たせることが多かった。そこで、1日目が終わったところで、調理の手順を見直すなど改善した場合に、どのように待ち状況が変化するかシミュレーションすることにした。なお、このお店では同時に一人の客しか対応できないとし、客が注文できるクレープは一枚のみと考える。また、注文は前の客に商品を渡してから次の注文を聞くとして考える。

問1 次の文章および表中の空欄 **ケ** ～ **シ** に当てはまる数字をマークせよ。

まず、Mさんは、1日目の記録を分析したところ、注文から商品を渡すまでの一人の客への対応時間に約4分を要していることが分かった。

次に、クラスの記録係が1日目の来客時刻を記録していたので、最初の50人の客の到着間隔を調べたところ、表1の人数のようになった。この人数から相対度数を求め、その累積相対度数を確率とみなして考えてみた。また、到着間隔は一定の範囲をもとに集計しているため、各範囲に対して階級値で考えることにした。

表1 到着間隔と人数

到着間隔（秒）	人数	階級値	相対度数	累積相対度数
0 以上～ 30 未満	6	0 分	0.12	0.12
30 以上～ 90 未満	7	1 分	0.14	0.26
90 以上～150 未満	8	2 分	0.16	0.42
150 以上～210 未満	11	3 分	0.22	0.64
210 以上～270 未満	9	4 分	0.18	0.82
270 以上～330 未満	4	5 分	0.08	0.90
330 以上～390 未満	2	6 分	0.04	0.94
390 以上～450 未満	0	7 分	0.00	0.94
450 以上～510 未満	1	8 分	0.02	0.96
510 以上～570 未満	2	9 分	0.04	1.00
570 以上	0	-	-	-

そして、表計算ソフトウェアで生成させた乱数（0以上1未満の数値が同じ確率で出現する一様乱数）を用いて試しに最初の10人の到着間隔を、この表1をもとに導き出したところ、次の表2のようになった。ここでの到着間隔は表1の階級値をもとにしている。なお、1人目は到着間隔0分とした。

表2 乱数から導き出した到着間隔

	生成させた乱数	到着間隔
1人目	—	0分
2人目	0.31	2分
3人目	0.66	4分
4人目	0.41	2分
5人目	0.11	0分
6人目	0.63	3分
7人目	0.43	3分
8人目	0.28	2分
9人目	0.55	3分
10人目	0.95	ケ分

表2の結果から10人の客の待ち状況が分かるように、次の図1のように表してみることにした（図1は6人目まで記入）。ここで、待ち時間とは、並び始めてから直前の人の対応時間が終わるまでの時間であり、対応時間中の客は待っている人数に入れないとする。このとき、最も待ち人数が多いときはコ人であり（これを最大待ち人数という）、客の中で最も待ち時間が長いのはサシ分であった。

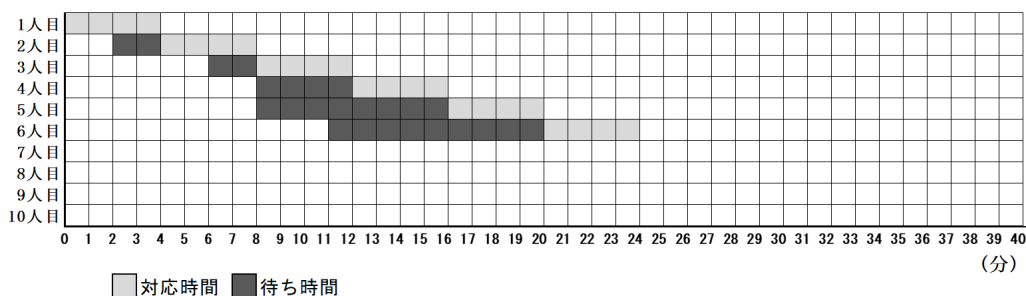


図1 シミュレーション結果（作成途中）

問2 図1の結果は、客が10人のときであったので、Mさんは、もっと多くの客が来た場合の待ち状況がどのようになるか知りたいと考えた。そこでMさんは、客が10人、20人、30人、40人来客した場合のシミュレーションをそれぞれ100回ずつ行ってみた。次の図2は、それぞれ100回のシミュレーションでの最大待ち人数の頻度を表したものである。

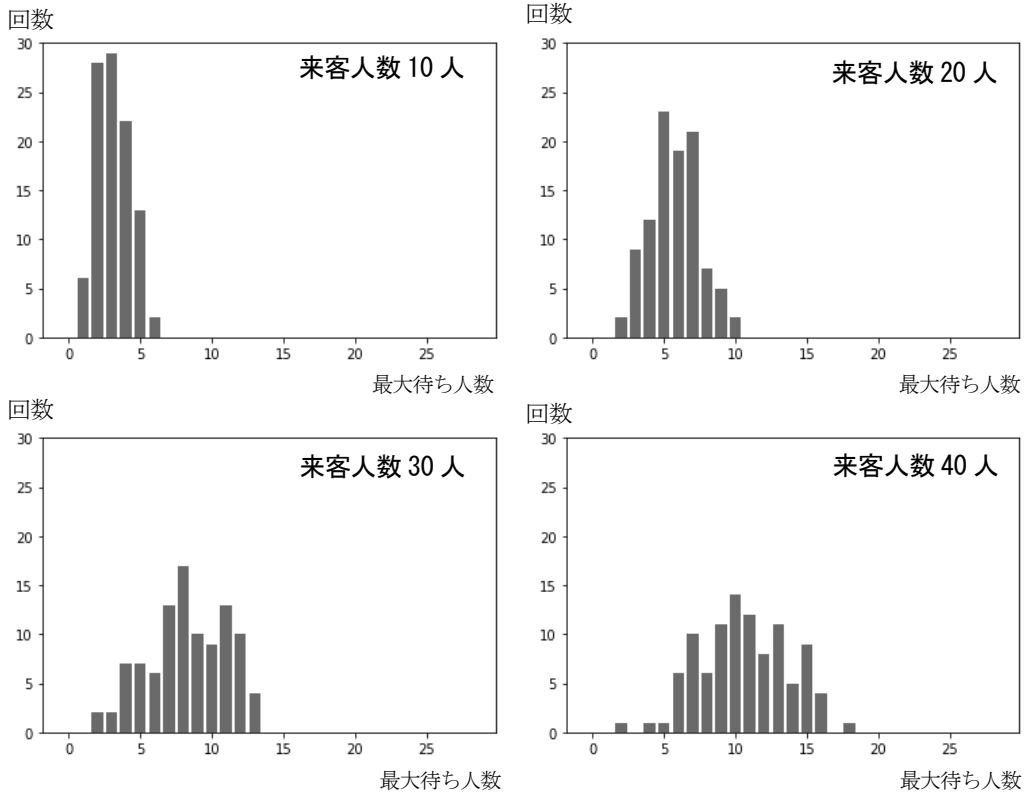


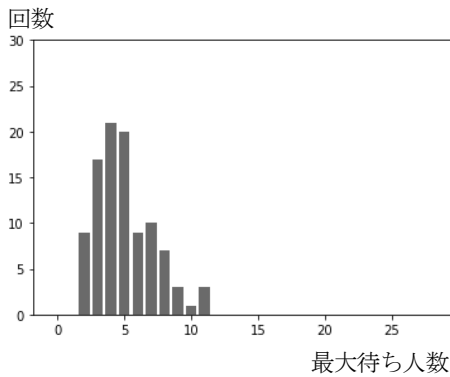
図2 シミュレーション結果

この例の場合において、シミュレーション結果から読み取れないことを次の①～③のうちから一つ選べ。 ス

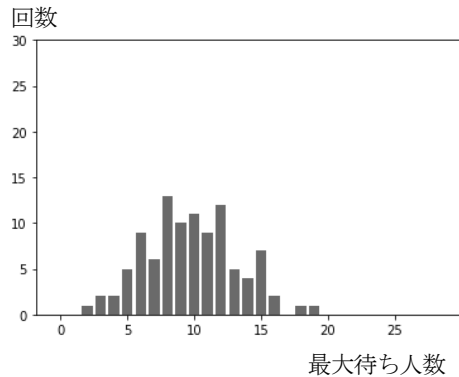
- ① 来客人数が多くなるほど、最大待ち人数が多くなる傾向がある。
- ② 最大待ち人数の分布は、来客人数の半数以下に収まっている。
- ③ 最大待ち人数は、来客人数の1/4前後の人数の頻度が高くなっている。
- ④ 来客人数が多くなるほど、最大待ち人数の散らばりが大きくなっている。

問3 1日目の午前中の来客人数は39人で、記録によれば一番長く列ができたときに10人の待ちがあったことから、Mさんは、図2の「来客人数40人」の結果が1日目の午前中の状況をおおよそ再現していると考えた。そこで、調理の手順を見直すことで一人の客への対応時間を4分から3分に短縮できたら、図2の「来客人数40人」の結果がどのように変化するか同じ乱数列を用いて試してみた。その結果を表すグラフとして最も適当なものを、次の①～③のうちから一つ選べ。 セ

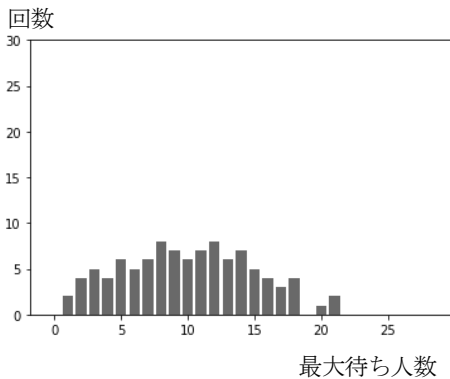
①



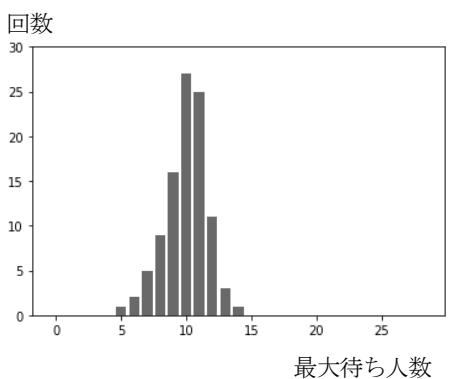
②



③



④



第3問 次の問い（問1～3）に答えよ。（配点 25）

問1 次の生徒（S）と先生（T）の会話文を読み、空欄 **ア** に当てはまる数字をマークせよ。また、空欄 **イ** ～ **エ** 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 **ウ** ・ **エ** は解答の順序は問わない。

S：この前、お客さんが460円の商品を買うのに、510円を払って、釣り銭を50円受け取っていたのを見て、授業で勉強したプログラミングで、そんな「上手な払い方」を計算するプログラムを作ってみたくて思いました。

T：いいですね。まず、「上手な払い方」とは何かを考える必要がありますね。

S：普通は手持ちの硬貨の枚数を少なくするような払い方でしょうか。

T：そうですね。ただ、ここでは、客が支払う枚数と釣り銭を受け取る枚数の合計を最小にする払い方を考えてみませんか？客も店も十分な枚数の硬貨を持っていると仮定しましょう。また、計算を簡単にするために、100円以下の買い物とし、使う硬貨は1円玉、5円玉、10円玉、50円玉、100円玉のみで500円玉は使わない場合を考えてみましょう。例えば、46円をちょうど支払う場合、支払う枚数はどうなりますか？

S：46円を支払うには、10円玉4枚、5円玉1枚、1円玉1枚という6枚で払い方が最小の枚数になります。

T：そうですね。一方、同じ46円を支払うのに、51円を支払って釣り銭5円を受け取る払い方では、支払いに2枚、釣り銭に1枚で、合計3枚の硬貨のやり取りになります。こうすると交換する硬貨の枚数の合計が最小になりますね。

S：これが上手な払い方ですね。

T：そうです。このように、客と店が交換する硬貨の合計が最小となる枚数、すなわち「最小交換硬貨枚数」の計算を考えましょう。

S：どうやって考えればいいかなあ。

T：ここでは、次の関数のプログラムを作り、それを使う方法を考えてみまし

よう。目標の金額を釣り銭無くちょうど支払うために必要な最小の硬貨枚数を求める関数です。

【関数の説明と例】

枚数(金額)… 引数として「金額」が与えられ、ちょうどその金額となる硬貨の組合せの中で、枚数が最小となる硬貨枚数が戻り値となる関数。
例：8円は「5円玉が1枚と1円玉が3枚」の組合せで最小の硬貨枚数になるので、枚数(8)の値は4となる。

T：これは、例えば、枚数(46) = と計算してくれるような関数です。これを使って最小交換硬貨枚数の計算を考えてみましょう。例えば、46円支払うのに、51円払って5円の釣り銭を受け取る払い方をした場合、客と店の間で交換される硬貨枚数の合計は、この関数を使うと、どのように計算できますか？

S： で求められますね。

T：一般に、商品の価格 x 円に対して釣り銭 y 円を $0, 1, 2, \dots$ と変化させて、それぞれの場合に必要な硬貨の枚数の合計を

$$\text{枚数}(\text{ウ}) + \text{枚数}(\text{エ})$$

と計算し、一番小さな値を最小交換硬貨枚数とすればよいのです。

S：なるほど。それで、釣り銭 y はいくらまで調べればよいのでしょうか？

T：面白い数学パズルですね。まあ、詳しくは今度考えるとして、今回は100円以下の商品なので y は99まで調べれば十分でしょう。

の解答群

- | | |
|------------------|------------------|
| ① 枚数(51) + 枚数(5) | ② 枚数(46) + 枚数(5) |
| ③ 枚数(51) - 枚数(5) | ④ 枚数(46) - 枚数(5) |

・ の解答群

- | | | | |
|-------|-------|-----------|-----------|
| ① x | ② y | ③ $x + y$ | ④ $x - y$ |
|-------|-------|-----------|-----------|

問2 次の文章の空欄 **オ** ～ **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。

S：まずは、関数「枚数(金額)」のプログラムを作るために、与えられた金額ちょうどになる最小の硬貨枚数を計算するプログラムを考えてみます。もう少しヒントが欲しいなあ。

T：金額に対して、高額 of 硬貨から使うように考えて枚数と残金を計算していくとよいでしょう。また、金額に対して、ある額の硬貨が何枚まで使えて、残金がいくらになるかを計算するには、整数値の商を求める演算『÷』とその余りを求める演算『%』が使えるでしょう。例えば、46円に対して10円玉が何枚まで使えるかは **オ** で、その際にいくら残るかは **カ** で求めることができますね。

S：なるほど！あとは自分でできそうです。

Sさんは、先生(T)との会話からヒントを得て、変数 **kingaku** に与えられた目標の金額(100円以下)に対し、その金額ちょうどになる最小の硬貨枚数を計算するプログラムを考えてみた(図1)。ここでは例として目標の金額を46円としている。

配列 **Kouka** に硬貨の額を低い順に設定している。なお、配列の添字は0から始まるものとする。最低額の硬貨が1円玉なので **Kouka[0]** の値は1となる。

先生(T)のヒントに従い、高額 of 硬貨から何枚まで使えるかを計算する方針で、(4)～(6)行目のような繰り返し文にした。この繰り返しで、変数 **maisu** に支払いに使う硬貨の枚数の合計が計算され、変数 **nokori** に残りいくら支払えばよいか、という残金が計算される。

実行してみると **ア** が表示されたので、正しく計算できていることが分かる。いろいろな例で試してみたが、すべて正しく計算できていることを確認できた。

- (1) `Kouka = [1,5,10,50,100]`
 (2) `kingaku = 46`
 (3) `maisuu = 0, nokori = kingaku`
 (4) `i` を `キ` ながら繰り返す:
 (5) `maisuu = ク + ケ`
 (6) `nokori = コ`
 (7) 表示する(`maisuu`)

図1 目標の金額ちょうどになる最小の硬貨枚数を計算するプログラム

`オ`・`カ` の解答群

- ① `46 ÷ 10 + 1` ② `46 % 10 - 1`
 ③ `46 ÷ 10` ④ `46 % 10`

`キ` の解答群

- ① 5 から 1 まで 1 ずつ減らし ② 4 から 0 まで 1 ずつ減らし
 ③ 0 から 4 まで 1 ずつ増やし ④ 1 から 5 まで 1 ずつ増やし

`ク` の解答群

- ① 1 ② `maisuu` ③ `i` ④ `nokori`

`ケ`・`コ` の解答群

- ① `nokori ÷ Kouka[i]` ② `nokori % Kouka[i]`
 ③ `maisuu ÷ Kouka[i]` ④ `maisuu % Kouka[i]`

問3 次の文章を参考に、図2のプログラムの空欄 **サ** ～ **タ** 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 **ス** ・ **セ** は解答の順序は問わない。

T：プログラム（図1）ができたようですね。それを使えば、関数「枚数(金額)」のプログラムができます。関数の引数として与えられる金額の値をプログラム（図1）の変数 **kingaku** に設定し、(7)行目の代わりに変数 **maisu** の値を関数の戻り値とすれば、関数「枚数(金額)」のプログラムとなります。では、その関数を使って最小交換硬貨枚数を計算するプログラムを作ってみましょう。ここでも、100円以下の買い物として考えてみます。

【関数の説明】(再掲)

枚数(金額)… 引数として「金額」が与えられ、ちょうどその金額となる硬貨の組合せの中で、枚数が最小となる硬貨枚数が戻り値となる関数。

Sさんは、図2のようなプログラムを作成した。変数 **kakaku** に与えられる商品の価格に対して、釣り銭を表す変数 **tsuri** を用意し、妥当な **tsuri** のすべての値に対して交換する硬貨の枚数を調べ、その最小値を求めるプログラムである。なお、ここでは例として商品の価格を46円としている。

このプログラムでは、先生(T)のアドバイスに従い、釣り銭無しの場合も含め、99円までのすべての釣り銭に対し、その釣り銭になるように支払う場合に交換される硬貨の枚数を求め、その最小値を最小交換硬貨枚数として計算している。

最小値の計算では、これまでの払い方での最小枚数を変数 **min_maisu** に記憶しておき、それより少ない枚数の払い方が出るたびに更新している。**min_maisu** の初期値には、十分に大きな値として100を用いている。100円以下の買い物では、使う硬貨の枚数は100枚を超えないからである。

```

(1) kakaku = 46
(2) min_maisu = 100
(3) サ を シ から 99 まで 1 ずつ増やしながら繰り返す：
(4)   shiharai = kakaku + tsuri
(5)   maisu = ス + セ
(6)   もし ソ < min_maisu ならば：
(7)   |   |   タ = ソ
(8) 表示する (min_maisu)

```

図 2 最小交換硬貨枚数を求めるプログラム

このプログラムを実行してみたところ 3 が表示された。46 円を支払うときの最小交換硬貨枚数は、支払いで 50 円玉が 1 枚、1 円玉が 1 枚、釣り銭で 5 円玉が 1 枚の計 3 枚なので、正しく計算できていることが分かる。同様に、**kakaku** の値をいろいろと変えて実行してみたところ、すべて正しく計算できていることを確認できた。

サ, ソ・タ の解答群

① maisu ② min_maisu ③ shiharai ④ tsuri

シ の解答群

① 0 ② 1 ③ 99 ④ 100

ス・セ の解答群

① 枚数 (shiharai) ② 枚数 (kakaku) ③ 枚数 (tsuri)
 ④ shiharai ⑤ kakaku ⑥ tsuri

第4問 次の文章を読み、後の問い（問1～5）に答えよ。（配点 25）

次の表1は、国が実施した生活時間の実態に関する統計調査をもとに、15歳以上19歳以下の若年層について、都道府県別に平日1日の中で各生活行動に費やした時間（分）の平均値を、スマートフォン・パソコンなどの使用時間をもとにグループに分けてまとめたものの一部である。ここでは、1日のスマートフォン・パソコンなどの使用時間が1時間未満の人を表1-A、3時間以上6時間未満の人を表1-Bとしている。

表1-A：スマートフォン・パソコンなどの使用時間が

1時間未満の人の生活行動時間に関する都道府県別平均値

都道府県	睡眠 (分)	身の回りの 用事 (分)	食事 (分)	通学 (分)	学業 (分)	趣味・娯楽 (分)
北海道	439	74	79	60	465	8
青森県	411	74	73	98	480	13
茨城県	407	61	80	79	552	11
栃木県	433	76	113	50	445	57

表1-B：スマートフォン・パソコンなどの使用時間が

3時間以上6時間未満の人の生活行動時間に関する都道府県別平均値

都道府県	睡眠 (分)	身の回りの 用事 (分)	食事 (分)	通学 (分)	学業 (分)	趣味・娯楽 (分)
北海道	436	74	88	63	411	64
青森県	461	57	83	55	269	44
茨城県	443	80	81	82	423	63
栃木県	386	120	79	77	504	33

(出典：総務省統計局の平成28年社会生活基本調査により作成)

花子さんたちは、表1-Aをスマートフォン・パソコンなどの使用時間が短いグループ、表1-Bをスマートフォン・パソコンなどの使用時間が長いグループと設定し、これらのデータから、スマートフォン・パソコンなどの使用時間と生活行動に費やす時間の関係について分析してみることにした。

ただし、表1-A、表1-Bにおいて一か所でも項目のデータに欠損値がある場合は、それらの都道府県を除外したものを全体として考える。なお、以下において、データの範囲については、外れ値も含めて考えるものとする。

問1 花子さんたちは、これらのデータから次のような仮説を考えた。表1-A、表1-Bのデータだけでは分析できない仮説を、次の①～③のうちから一つ選べ。

- ① 若年層でスマートフォン・パソコンなどの使用時間が長いグループは、使用時間が短いグループよりも食事の時間が短くなる傾向があるのではないか。
- ② 若年層でスマートフォン・パソコンなどの使用時間が長いグループに注目すると、スマートフォン・パソコンなどを朝よりも夜に長く使っている傾向があるのではないか。
- ③ 若年層でスマートフォン・パソコンなどの使用時間と通学の時間の長さは関係ないのではないか。

問2 花子さんたちは表1-A、表1-Bのデータから睡眠の時間と学業の時間に注目し、それぞれを図1と図2の箱ひげ図（外れ値は○で表記）にまとめた。これらから読み取ることができる最も適当なものを、後の①～③のうちから一つ選べ。

イ

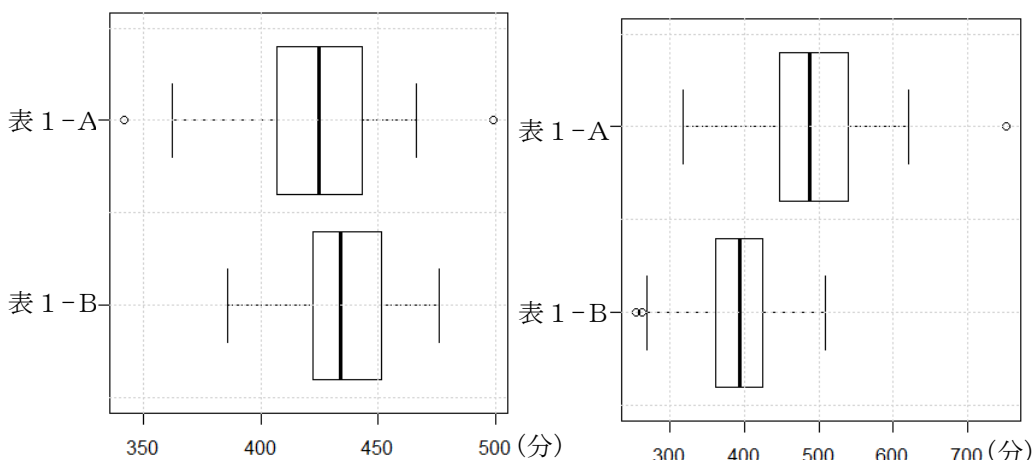


図1 睡眠の時間の分布

図2 学業の時間の分布

- ① 睡眠の時間が420分以上である都道府県の数を見たとき、表1-Aの方が表1-Bよりも多い。
- ② 学業の時間が550分以上の都道府県は、表1-Aにおいては全体の半数以上あり、表1-Bにおいては一つもない。
- ③ 学業の時間が450分未満の都道府県は、表1-Bにおいては全体の75%以上であり、表1-Aにおいては50%未満である。
- ④ 都道府県別の睡眠の時間と学業の時間を比較したとき、表1-Aと表1-Bの中央値の差の絶対値が大きいのは睡眠の時間の方である。

(下書き用紙)

情報 I の試験問題は次に続く。

問3 花子さんたちは、スマートフォン・パソコンなどの使用時間の長さの違いが、睡眠の時間と学業の時間のどちらに大きく影響しているかについて調べることにした。そのために、都道府県ごとに睡眠の時間と学業の時間のそれぞれにおいて、表1-Aの値から表1-Bの値を引いた差について考え、その結果を次の図3の箱ひげ図（外れ値は○で表記）で表した。図3について述べたこととしてA～Eの中から正しいものはどれか。当てはまるものの組合せとして最も適当なものを、後の①～⑤のうちから一つ選べ。 ウ

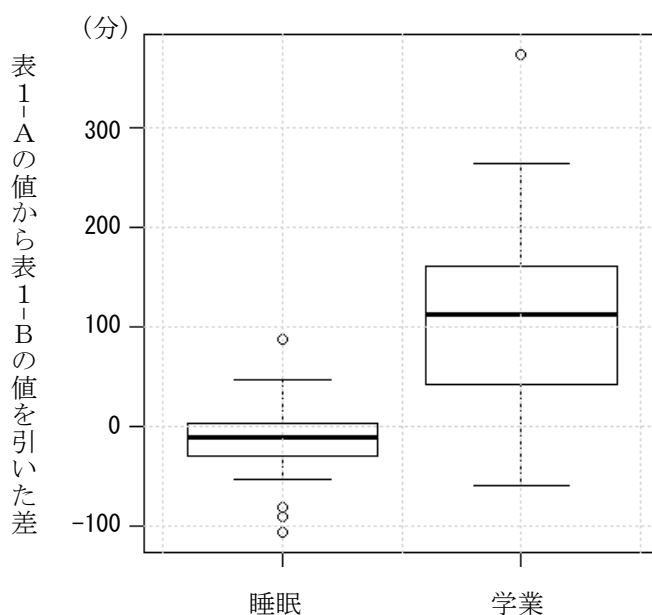


図3 生活行動時間の差

- A 学業の時間の差が正の値になっている都道府県の若年層は、スマートフォン・パソコンなどの使用時間が短いグループの方が、学業の時間が長い傾向にある。
- B 睡眠の時間の差が正の値になっている都道府県の若年層は、スマートフォン・パソコンなどの使用時間が短いグループの方が、睡眠の時間が短い傾向にある。
- C スマートフォン・パソコンなどの使用時間による生活行動時間の差は、睡眠の時間よりも学業の時間の方に顕著に表れている。
- D スマートフォン・パソコンなどの使用時間による生活行動時間の差は、学業の時間よりも睡眠の時間の方に顕著に表れている。
- E スマートフォン・パソコンなどの使用時間による生活行動時間の差は、学業の時間と睡眠の時間の両方に同程度に表れている。

② AとC

① AとD

② AとE

③ BとC

④ BとD

⑤ BとE

問4 花子さんたちは、表1-Aについて、睡眠の時間と学業の時間の関連を調べる
 こととした。次の図4は、表1-Aについて学業の時間と睡眠の時間を散布図で
 表したものである。ただし、2個の点が重なって区別できない場合は □ で示し
 ている。

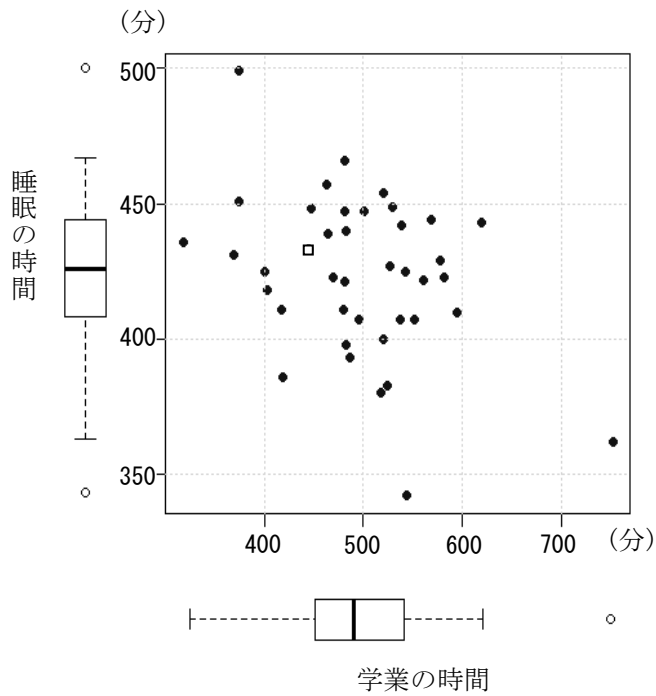


図4 表1-Aの学業の時間と睡眠の時間の散布図

都道府県単位でみたとき、学業の時間と睡眠の間には、全体的には弱い負の相関があることが分かった。この場合の負の相関の解釈として最も適当なものを、次の①～③のうちから一つ選べ。なお、ここでは、データの範囲を散らばりの度合いとして考えることとする。

エ

- ① 睡眠の時間の方が、学業の時間より散らばりの度合いが大きいと考えられる。
- ② 睡眠の時間の方が、学業の時間より散らばりの度合いが小さいと考えられる。
- ③ 学業の時間が長い都道府県ほど睡眠の時間が短くなる傾向がみられる。
- ④ 学業の時間が長い都道府県ほど睡眠の時間が長くなる傾向がみられる。

問5 次の文章を読み、空欄 **オ** に当てはまる数字をマークせよ。また、空欄 **カ** に入れるのに最も適当なものを、図6中の①～③のうちから一つ選べ。空欄 **キ** 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。

花子さんたちは都道府県別にみたときの睡眠の時間を学業の時間で説明する回帰直線を求め、図4の散布図にかき加えた(図5)。すると回帰直線から大きく離れている県が多いことが分かったため、自分たちの住むP県がどの程度外れているのかを調べようと考え、実際の睡眠の時間から回帰直線により推定される睡眠の時間を引いた差(残差)の程度を考えることとした。そのために、残差を比較しやすいように、回帰直線の式をもとに学業の時間から推定される睡眠の時間(推定値)を横軸に、残差を平均値0、標準偏差1に変換した値(変換値)を縦軸にしてグラフ図6を作成した。参考にQ県がそれぞれの図でどこに配置されているかを示している。また、図5の□で示した点については、問題の都合上黒丸で示している。

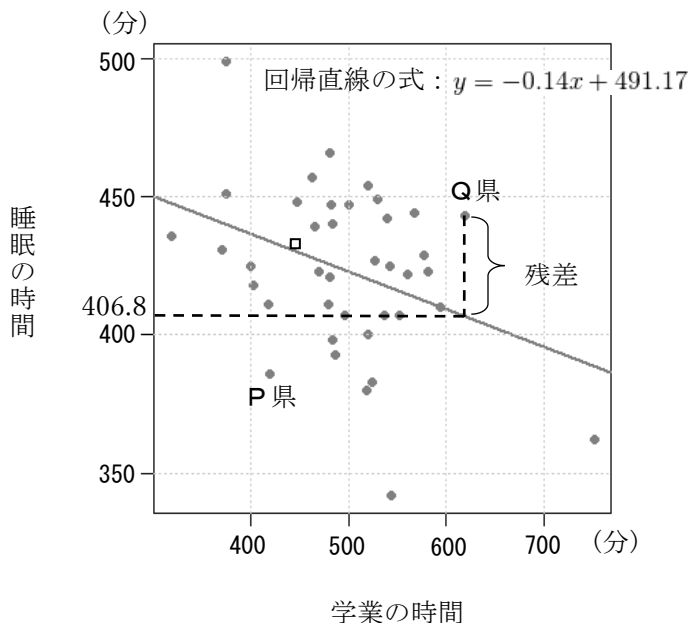


図5 回帰直線をかき加えた散布図

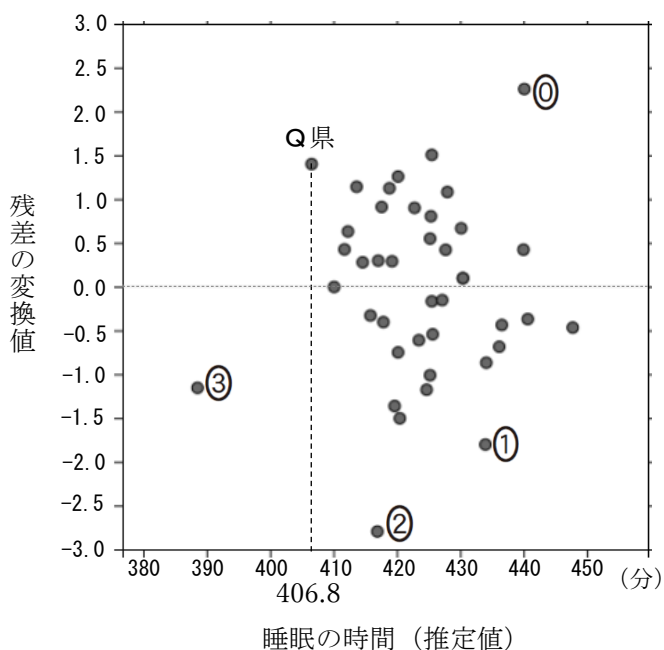


図6 睡眠の時間（推定値）と残差の変換値との関係

図5と図6から読み取ることができることとして、平均値から標準偏差の2倍以上離れた値を外れ値とする基準で考えれば、外れ値となる都道府県の数はお 個である。図5中のP県については、図6中の①～④のうち カ に対応しており、花子さんたちはこの基準に従いP県は キ と判断した。花子さんたちは学業の時間以外の他の要因の影響についても考え、さらに都道府県の特徴について分析することとした。

キ の解答群

- ① 外れ値となっている ② 外れ値となっていない
- ③ 外れ値かそうでないかどちらともいえない