

京都府私立中学高等学校情報科研究会

2025年度（令和7年度） 夏期研究会 次第

日時：2025年（令和7年）7月3日（木）

場所：京都産業大学14号館1階14113会議室

テーマ「プログラミング必修化と大学入試、そして大学教育のこれから」

●意見交換

「大学の個別入試における情報Ⅰについて」

「高校の情報Ⅰにおけるプログラミング必修化による高校生の状況」

「情報Ⅱ等の上位科目でのプログラミング」

「大学入学までに高校生に求めたい内容」

1. 開会あいさつ
2. 高校側からの話題提供→質疑応答
3. 大学側からの話題提供→質疑応答
4. まとめ
5. 閉会あいさつ

施設見学 16:00～

17:00閉会

<ご案内>

研究会サイト<https://www.kyoto-shigaku.info>ぜひご利用ください。

事務局：同志社中学校・高等学校
〒606-8558 京都市左京区岩倉大鷲町89
TEL:075-781-7121 FAX:075-781-7124
庶務 鈴木 潤 (jnsuzuki@js.doshisha.ac.jp)

情報入試のご案内

京都産業大学 情報理工学部および理学部では、「情報」の入試を実施します。
情報の入試の関連情報を以下にまとめてありますので、是非ともサイトをチェックしてください。

京都産業大学の情報入試 情報プラス型とは？

公募推薦入試および一般選抜入試「前期日程」スタンダード2科目型に、情報科目をプラスして合否判定する入試です。

2026年度入試 情報プラス型対象日程：

<公募推薦入試>2025年11月22日（土）

<一般選抜入試「前期日程」スタンダード2科目型>2026年1月29日（木）

合否判定：以下に示す①と②の2回合否判定し、どちらかで合格圏内に入れば合格となる。

①英語 100点, 数学 100点

②英語 100点, 数学 100点, 情報 200点（情報理工学部）/100点（理学部）

特に、情報理工学部は情報の配点割合が、英語と数学の2倍となっており、情報の知識を活かせる入試となっています。

また、ご質問やご相談等ございましたら、右下の「お問い合わせフォーム」からお問い合わせください。

京都産業大学の情報入試に関する情報を掲載した Web サイト

情報入試を受験される
皆さんへのメッセージ



入試情報サイト情報プラス型



情報プラス型の説明動画
(YouTube)



模擬問題



情報理工学部
AO 入試過去問題



情報理工学部への
お問い合わせフォーム



むすんで、うみだす。

京都産業大学
KYOTO SANGYO UNIVERSITY

OPEN CAMPUS 2025

7/20 SUN.
8/2 SAT. 3 SUN.

2026年4月

- ▶ 文化学部が進化！
 - ▶ アントレプレナーシップ学環始動！
- オープンキャンパスで、新しくなる学びをチェック！

事前申込制
【当日申込可】

10:00～15:30
【9:15～受付】

お申し込み・詳細情報はこちら！



注目イベント

人気予備校講師による解説

受験英語対策講座！

■ 経済学部
経済学科

■ 経営学部
マネジメント学科

■ 法学部
法律学科
法政策学科

■ 現代社会学部
現代社会学科
健康スポーツ社会学科

■ 国際関係学部
国際関係学科

■ 外国語学部
英語学科
ヨーロッパ言語学科
アジア言語学科

■ 文化学部※
文化構想学科 **NEW**
京都文化学科 **RENEWAL**
文化観光学科 **NEW**

■ 理学部
数理科学科
物理科学科
宇宙物理・気象学科

■ 情報理工学部
情報理工学科

■ 生命科学部
先端生命科学科
産業生命科学科

■ アントレプレナーシップ学環※ **NEW**

※学部・学科・学環名など、記載内容は変更になる場合があります。(2026年4月開設に向けて設置申請中)

新しい学びをチェック

2026年4月
新たな文化学部へ!

2026年4月始動
アントレプレナーシップ学環

文化・歴史・芸術 × AI・データサイエンス

文化構想学科・京都文化学科・文化観光学科の3学科に! 伝統と最先端が融合した新しい教育がスタート! 「文化で未来を変える学び」がここにあります。

新しい大学教育のカタチ

少人数制(1学年30人)の実践的カリキュラムと自己開拓力を育む学びが特長です。起業家精神・実践力・行動力を養い、未来のビジネスリーダーを育成する新学部です。

オープンキャンパスでの充実したプログラム

大学紹介・入試説明

京都産業大学ってどんな大学? 文系・理系が融合する日本最大級のワンキャンパスでは、学びや学生生活の魅力を紹介し、入試制度やその特徴についてもポイントを確認しよう。

キャンパスツアー

学生スタッフとウォーキング・バスで、ユニバーサル・スタジオ・ジャパンの1.2倍ある広大なキャンパスを満喫しよう。

学生スタッフ企画

学生スタッフが企画・運営する特別イベント。京都産業大学での学生生活はもちろん、「学生のまち」といわれる京都での一人暮らしやアルバイトのことなど、気になることは何でもご相談ください。

施設見学・研究室公開

国内私立大学最大の天体望遠鏡を備える神山天文台や異文化交流を自由に楽しめるグローバル commons など、本学の先進的な学習環境を見学できます。

学部・学環プログラム

教員や先輩学生による学部・学科・学環紹介、研究紹介など各学部・学環が工夫を凝らした企画で特徴を紹介し、

学部・学環まる分り講座(文系編)

大学選びは学部が大事! 文系学部はどうやって選べば良いの? を解決しよう! 京都産業大学の文系学部・学環について分りやすく説明します。

総合型選抜入試説明会(学部・学環ごとに開催)

総合型選抜入試志願者必見! 総合型選抜入試について、学部・学環ごとに選考方法やポイントなどを説明します。

保護者企画

保護者の方向けに現在の大学の状況や入試のトレンド、本学の就職支援などをお伝えします。

OPEN CAMPUS 2025

魅力あふれるプログラムをご用意してお待ちしております!

OPEN CAMPUS POINT

1・2年生 大学生生活を体感しよう 3年生 志望学部を決めよう

最も来場者が多い夏のオープンキャンパス! 充実した学部・学環イベントで志望学部に対する理解をしっかりと深めよう。高校1・2年生は大学選びのスタートとして大学生生活を体感しよう。

受験英語対策講座

7.8月

早めに受験対策をスタートしたい高校1・2年生大歓迎です。

本学の英語の出題傾向や重要ポイントを予備校の人気講師が解説! 効率の良い学習方法を身に付けるチャンスです。

※内容は変更する場合があります。イベント詳細および事前申込は京都産業大学の「入試情報サイト」をご確認ください。



お友達登録はこちらから

オープンキャンパスをはじめとした最新情報をいち早くゲット!
LINEで「京都産業大学 入学センター」を友達追加しよう。

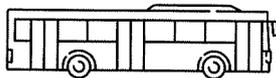


イベント詳細はこちらから

OPEN CAMPUS 2025サイト

京都産業大学 オープンキャンパス

検索



2025年、京阪出町柳駅から通学バスが倍増!
とても、通いやすくなりました!

京都駅より
所要時間 約29分

京都駅

地下鉄 約20分

国際会館駅

地下鉄 約14分

北大路駅

大阪梅田駅より
所要時間 約66分

大阪梅田駅

阪急 約41分

烏丸駅

地下鉄四條駅へ乗り換え

淀屋橋駅より
所要時間 約84分

淀屋橋駅

京阪 約54分

出町柳駅

京都バス 約9分

市バス・京都バス 約15分

地下鉄 約16分

国際会館駅

京都バス 約9分

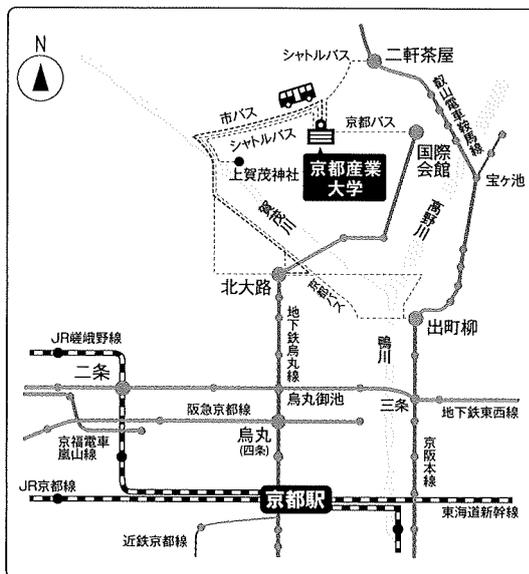
京都バス直行19分(普通30分)

※乗り換え時間は含みません

京都産業大学

京都産業大学 入学センター

〒603-8555 京都市北区上賀茂本山
TEL.075-705-1437



交通アクセスの詳細はこちらから



※QRコードはデバイスにより読み取れない場合があります

試験日		前期日程					
		1/29 (情報プラス型)					
[前期日程] 情報プラス型	大問	I	14点	Ⅲ	16点	V	12点
		Ⅱ	14点	Ⅳ	24点	Ⅵ	20点
	合計	100点					

■理学部・情報理工学部のみ出願可。解答方法は、記述式

出題内容

大問ごとの構成要素を示します。

- (I) コンピュータの構成・データ表現・データ量
- (II) データ表現 (バーコードを題材として)
- (III) ビット演算・フローチャート
- (IV) プログラミング (受講生をランダムにグループ分けする)
- (V) データ分析 (表とグラフからの情報読み取り)
- (VI) データ分析 (モデル化とシミュレーション)

出題のねらい

2025年度入試から共通テストにも「情報」が出題され、ようやく個別入試においても受験者の「情報」の資質を問えるようになりました。共通テストは「情報Ⅰ」の全領域から出題されますが、情報理工学部および理学部における今回の問題は、主に両学部に必要な情報の資質があることを確かめるためのもので、幾つかの領域に絞り込んで出題しています。

つまりコンピュータやデジタル処理の基本的な理解 (I・II・III)、プログラミング (IV)、データ分析 (V・VI) の三つです。ただしIIの後半は単純なデータ表現 (組み合わせ) の問いから少し離れて、思考力を問う性質を込めています。またIIIはフローチャートの形をしていますがビット演算命令を用いて手続的な処理を追いかけている点ではプログラミング的な要素もあります。そうすることで丸覚えや短絡的な推定ではなく、理解できていることや提示された情報から思考し・判断する力を評価しようとしています。

入試対策としては Web に公開されている模擬試験問題やそれ以前の情報理工学部総合型選抜入試 (「筆記型」を含む) の過去問を見ておくと良いでしょう。また他大学の個別入試や事前に公開されたサンプル問題なども参考になると思われます。

結果を振り返って

- (I) 教科書に良く出てくるタイプの問題です。期待通り設問 (A) はよく出ていましたが、逆に (B) は得点率がかなり低くなりました。バイト数の計算などは「情報」の教科書の練習問題に良く出ています。転送速度 (一秒あたりのビット数) やフレームレート (一秒あたりのフレーム数) などを組み合わせると難しくなる、ということのようですが、しかしこれは出来てほしいところでした。今後こうした問題が出ますので押さえておくことを勧めます。
- (II) バーコードの形をしていますが、実質的には桁数を n とした二値表現に関する話題で、設問 (A)(B) はその値の範囲を問うものです。数学の「組み合わせ」の問題として解けますね。設問 (C) はコンピュータ特有の「読み取り異常」が発生した場合について検討するものです。少し数学から離れて思考することになります。いずれの設問もよく出ていました。
- (III) ビット列に対する演算処理を扱う問題です。AND・OR は良く出

てくる概念でしょうが、ここではシフト処理について提示しています。そのうえでビット列を「逆順」にする手順を日本語 (箇条書き) とフローチャートで示し、それをトレースすることで解答できる問題です。仕組みを提示して、その理解に基づいて思考する、という「情報」の問題としてはよくあるタイプのもので、これも全体的によく出ていました。

(IV) ストレートにプログラミング能力を問う問題です。N人の受講生をランダムにグループ分けする筋書きですが、内部的には配列データの入れ替え操作です。その手順が提示されるので、それに合わせた適切な手続きを選ぶか記述する形で解答します。全体に得点率が低かったのですが、それでも設問 (B) はとてもよく出ていました。つまり提示されたシャッフルの手順自体はよく理解できていたのに、コードが入ると急に答えられなくなるように思えます。逆にこの部分で得点できると高順位に付けられます。

(V) 実際のデータを材料に、旅行目的ごとの旅行者数の変化・傾向を見る問題です。設問 (A) は表から具体的な数字をピックアップして計算する形、設問 (B) はグラフから傾向を読み取る形で、データを正しく解釈出来ているか確認します。この種の問題はどうしても説明記述が長くなりがちですが、そこからデータの意味を適切に解釈し、正しく反応できる能力を評価しています。皆さんそこにうまく対応出来ていたようで、全体によく出ていました。

(VI) この問題ではエアコンを入れた時の室内温度の変化 (実験値) をグラフの形で提示し、これを分析してモデル式を導出、その式を元に計測値とモデル計算値を比べて評価する作業が順に示されます。その過程でそれぞれ適切な値や選択肢を答える、という、まさにモデル化とシミュレーションの典型的な問題ですね。やはり説明記述が長く、Vの問題に比べると少し低めの正答率となりつつも、おおよそよく出ていました。

アドバイス

今回が初めての「情報」の出題となり、既に述べた三つの柱、コンピュータやデジタル処理の基本的な理解、プログラミング、データ分析を軸とした問題が出されました。そこでは共通テストと比べて教科書から少し踏みこんだ、より深い理解や思考力を問うものが含まれています。まだ初年度で、今後の展開がどうなるか不確定なところも多いですが、この方向性はおおよそ変わらないものと思います。

以下、幾つか情報について学ぶ、情報の入試を受ける際に心がけると良いことを示します。

1. 共通テストも含め、「情報」の問題は全体に説明が長くなる傾向にあります。VI問題も説明記述が長めですが、逆にこのくらいシンプルで典型的なモデルでも、問題にすると結構な説明量になることが分かるでしょう。類題的な問題の解き方、答え方などに慣れることも必要でしょうが、何より読解力・思考力をつけるつもりで過去問や参考問題に取り組むことを勧めます。
2. たとえばIIIの「ソフト命令」はCPUの機械語命令としてはごく普通で、そこで出てきた「特定のビットをマスクする」操作もビット

処理ではごく一般的です。そうしたことを知っていた人は素早く解けたと思いますが、知らなくても説明をしっかりと読めば正しく解ける問題になっています。この「提示された情報を読み、理解して再構築する」ことそのものが「情報」の分野における重要な能力の一つと考えて、そうした問題を用意しています。参考問題などでは、見慣れない題材の問題にこそじっくり取り組むと訓練になって良いと思います。

3. プログラミング問題は苦手な人が多いようですが、逆にここが出来

ると高得点につながりますから、しっかり取り組む価値はあると思います。ただ、教科書ではかなり単純なプログラム例しか出ていませんし、今回の共通テストの問題も比較的シンプルなプログラムでした。それに較べると（本学に限らず他大学にしても）個別入試は、より複雑な処理、長めのコードが提示されると思われます。プログラミング慣れのために、ある程度以上複雑な処理のものを自分で書くことも重要ですが、一方で他人が書いたコードやその説明を読む経験を多くすると良いです。

プログラミング問題で用いる表記例

プログラミングに関する問題および解答において、独自の記法を用いてプログラミングを記述します。以下はその（疑似言語による）例示です。

プログラム表記の例示

2025年度入試版

本試験の間、および解答においてプログラムを記述するために用いる記法について説明する。

演算子

数値に対して算術演算を行うことができる。演算子は以下の通り。なお、*、/、% の計算が + や - よりも先に行われる。

+ 加算(足し算) - 減算(引き算)
* 乗算(掛け算) / 除算(割り算)
% 剰余算(割り算の余り)。(例: 7%3 は 1)

数値同士、文字列同士を比較できる。結果は真または偽である。

A == B AとBの値が等しい。
A != B AとBの値が等しくない。

数値の大小を比較できる。結果は真または偽である。

A <= B AがB以下 A < B AがBより小さい
A >= B AがB以上 A > B AがBより大きい

複数の条件を組み合わせたリ、ある条件を否定するために以下の論理演算子を用いることができる。

条件1 AND 条件2 条件1 OR 条件2 !条件

算術演算と比較演算では、算術演算が先に計算される。また、OR よりもAND、ANDよりも!が優先される。式の中で () を使い、計算の順序を示すことができる。

文

count = count + 1 変数に式の値を代入する。左の例は変数の値を1増やす。

for i = 0 to N-1
 table[i] = 0 変数の値を0から(N-1)まで1ずつ増やして繰り返す。
end

while a[n] == b[n] 条件が成り立つ間、繰り返しを実行する。
 n = n + 1
end

while a[n] == b[n] if文の条件が成り立てば、if a[n] == 0 while文の繰り返しを打ち切る
 break (同じように、for文の繰り返しも打ち切ることができる)。
 end
 n = n + 1
end

if kekka != 0 条件が成り立つときに実行する。
 print("当選")
end

```
if kion < 30    条件が成り立つかどうかで、実行
  print("実施")    することを変える。
else
  print("中止")
end
```

```
if tokuten >= 80    複数の条件を順番に調べて
  print("優秀")    実行することを変える。
elif tokuten >= 60
  print("合格")
else
  print("追試")
end
```

```
return gokei    関数の実行を終了して値を呼び出し側へ戻す。
```

```
return    関数の実行を終了する(戻り値の必要ない場合)。
```

```
print("合計:", w + i)    値や文字列を表示する。
```

変数と配列: 変数(または配列)は、関数の内部で宣言した場合、その関数でのみ利用可能な変数(ローカル変数)となり、関数の外部で宣言するところからでも利用可能な変数(グローバル変数)となる。

```
var i, j    変数を宣言する。宣言と同時に初期値を指定できるが、指定がない変数の初期値は不定(意味のない値)である。
var total = 0
```

```
var table[10]    配列を宣言する。上の例ではtable[0]からtable[9]まで10個の要素が用意される。
```

```
var s[] = {-1, 0, 1}    初期値を指定した配列を宣言する。上の例ではs[0]に-1, s[1]に0, s[2]に1が設定される。
```

関数

```
func add(a, b)    関数(サブルーチン)を定義する。戻り値のある関数は return文で値を指定する。左の関数は次の例のように呼び出せる。
var sum
sum = a + b
return sum
end    total = add(m, 500)
```

```
func show(t)    戻り値のない関数は return
  print("答:", t)    文を省略できる。左の関数は次の例のように呼び出せる。
end    show(n * 2)
```

コメント: プログラムの記述中に # が現れた場合、そこから行末までの文字列はコメント(注釈)とみなし、実行されない。ただし、文字列の中に現れた # はコメントとしては扱わない。

プログラム例1: 買い物合計額を計算する関数の定義

買い物項目数が n、配列 price, amount に買った品物の単価と個数が格納されているとする。合計金額が 2000円以上なら送料が無料になる。変数 delivery はグローバル変数である。

```
var delivery = 500 # 送料
func shopping(price, amount, n)
  var pay = 0
  var i
  for i = 0 to n - 1
    pay = pay + price[i] * amount[i]
  end
  if pay >= 2000 # 合計が2000円以上
    return pay
  end
  return pay + delivery
end
```

プログラム例2: 三角形の種類を調べる関数の定義

```
func triangle(x, y, z)
# 引数は3辺の長さ。ただし、x ≥ y ≥ z とする。
if x >= y + z
  print("三角形ではない")
elif x == y OR y == z
  if x == z
    print("正三角形")
  else
    print("二等辺三角形")
  end
else
  print("三角形")
end
end
```

本表記方法は既存のプログラミング言語とは異なる疑似言語によるものである。

[I] 以下の文章を読んで、設問(A)と設問(B)に答えなさい。

図 I-1 は、コンピュータを構成する基本的な装置とその関係を表したものである。

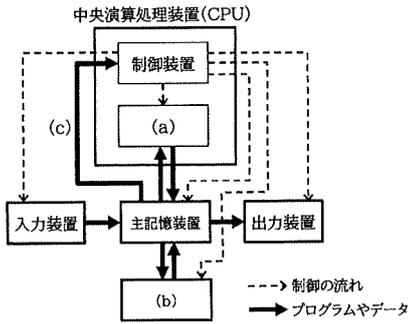


図 I-1. コンピュータの構成

設問(A)

空欄 [ア] ~ [オ] に入れるのに最も適切な語を、解答群のうちから1つずつ選び、その番号を解答欄に記入せよ。

- 図 I-1 の(a)は [ア] であり、(b)は [イ] である。
- 入力装置の例としては [ウ] があり、出力装置の例としては [エ] がある。
- 図 I-1 の(c)が示す、主記憶装置から制御装置に送られるものは主に [オ] である。

[ア] ~ [オ] の解答群:

- [(1) メインメモリ (2) 補助記憶装置 (3) キーボード (4) プリンタ
 (5) アルゴリズム (6) オブジェクト (7) プロトコル (8) プログラム
 (9) 演算装置 (10) ドライバ]

設問(B)

入力装置として、動画を撮影できるカメラを考える。撮影した動画データは図 I-1 の(b)に一旦蓄積される。いくつかの動画データを、コンピュータにつながるネットワークを通じて他のコンピュータに転送したところ、平均的な転送速度は 40Mbps であった。

このとき、空欄 [カ] ・ [キ] に入る数値として最も適切なものを、それぞれの解答群のうちから1つずつ選び、その番号を解答欄に記入せよ。

なお、1MB(メガバイト)は 1,000,000B(100 万バイト)とする。

- 200MB の動画データを転送するのにかかる時間はおよそ [カ] 秒となる。

[カ] の解答群:

- [(1) 1 (2) 4 (3) 5 (4) 10 (5) 20
 (6) 40 (7) 50 (8) 100 (9) 200 (10) 400]

- カメラの解像度は 2000×1000 画素で、各画素は RGB 各 8bit のフルカラーで、毎秒 15 フレームの動画を撮影できるとする。このとき、1 フレームのデータ量は [キ] MB となる。

[キ] の解答群:

- [(1) 2 (2) 6 (3) 9 (4) 16 (5) 48
 (6) 60 (7) 72 (8) 90 (9) 160 (10) 240]

[II] 以下の文章を読んで、設問(A)~(C)に答えなさい。

バーコードの規格にはさまざまなものがあるが、この問題では単純化した以下のルールによる表現方式(以降「本方式」とする)を検討する。

- ルール 1 : 1 つの文字を n 本のバーで表現する
- ルール 2 : バーには幅が広いものと細いもの 2 種類ある
- ルール 3 : ルール 1 に示した n 本のバーのうち 2 本だけが広い

ルール中の n を 5 とした場合、A~D の文字を例えば図 II-1 のように表現できる。



図 II-1. n を 5 とした場合の本方式での A~D の文字の表現例

なお、本方式ではバーコードを左から右に読むため、例えば図 II-2 に示すように文字「A」を 180 度回転させたバーコードは別の文字として扱われる。



図 II-2. 文字 A(左)を 180 度回転させたバーコード(右) (矢印は読む方向を意味する)

設問(A)

ルール中の n を 5 とした場合の本方式で表現できる文字は最大何種類か解答欄に記入せよ。

設問(B)

ルール中の n をいくつに変更すれば、数字 10 種(0~9)と英大文字 26 種(A~Z)を表現できるようになるか検討し、その最小値を解答欄に記入せよ。

設問(C)

設問(A)に示した方式で英字からなる製品記号 3~5 文字をバーコードで表現し、ラベルに印刷した箱の側面に貼って在庫管理に使用したい。図 II-3 に製品記号「ABCD」に貼るラベルの例を示す。バーコードの読み取り機は矢印が示すようにバーに対して横方向にレーザーを当てて(スキヤンして)バーの幅の広さを読み取る。



図 II-3. ラベルの例

しかしラベルはシワや汚損によって、正しくバーコードから製品記号を読み取れずエラー(そのようなバーの並びの文字はない)と判断されるか、あるいは誤った(箱の中の製品と一致しない)製品記号として認識される可能性がある。以下のそれぞれの状況説明について正しいものには○、誤っているものには×を解答欄に記入せよ。

- バーの幅について、1 本だけ広い幅を細い、または細い幅を広いと読み間違えた場合、エラーとなる
- バーの幅について、2 本の広い幅を細いと読み間違えた場合、エラーではなく誤った(箱の中の製品と一致しない)製品記号と認識される
- 箱の上下(天地)が逆になって積まれ、読み取り機がバーの並び順を逆から読んだとしても、必ず正しい(箱の中の製品と一致する)製品記号を認識できる
- シワやラベルの欠損によってバーをいくつか認識できなかった場合、エラーになるだけで誤った製品記号として認識されることはない

〔 III 〕 以下の文章を読んで、設問に答えなさい。

8つのビットの並び(以下単にビット列と呼ぶ)の処理について考える。

空欄 **ア** ~ **オ** に入る適切な数値あるいはビット列を解答欄に記入せよ。ただし、**ウ** は対応する解答群から適切なものを選び、その番号を解答欄に記入せよ。

設問

- ビット列 A 01010011 を 10 進法で表したときの値は **ア** である。
- ビット列 A の下から 2 桁目が 1 であるかを判定したい。そのためにはまず、ビット列 A とビット列 **イ** との **ウ** 演算を行う。ただし、ビット列に対する演算は表 III-1 の通りである。次に **ウ** 演算の結果が **イ** と一致するかを調べ、一致していればビット列 A の下から 2 桁目が **エ** であり、一致しなければ **オ** であるとわかる。
ウ の解答群: 【 (1) AND (2) OR (3) LSHIFT (4) RSHIFT 】
- ビット列 A を逆順に並び替える処理を考える。結果となるビット列 R を 00000000 で初期化した後に、以下の(1)~(4)の手順を 8 回繰り返す。
 - A の最下位ビットを取得する。
 - 取得したビットが 1 であれば結果となる R の最下位ビットを 1 にする。そうでなければ何もしない。
 - R を左に 1 桁ずらす。
 - A を右に 1 桁ずらす。

(問題は右ページに続く)

表 III-1. ビット列に対する演算(右ページに続く)

演算	説明	例
AND	2つのビット列を比べ、同じ桁のビット同士を比較し、両方とも 1 の場合のみ 1 となる演算を行う。右例は 2 つのビット列 01010101 と 00111100 の AND 演算の例であり、結果は 00010100 であることを示している。	<pre> 0 1 0 1 0 1 0 1 0 0 1 1 1 1 0 0 ----- 0 0 0 1 0 1 0 0 </pre>
OR	2つのビット列を比べ、同じ桁のビット同士を比較し、少なくとも一方が 1 の場合に 1 となる演算を行う。右例は 2 つのビット列 01010101 と 00111100 の OR 演算の例であり、結果は 01111101 であることを示している。	<pre> 0 1 0 1 0 1 0 1 0 0 1 1 1 1 0 0 ----- 0 1 1 1 1 1 0 1 </pre>

図 III-1 は、このビット列 A を逆順に並び替える処理を実施するフローチャートである。

空欄 **カ** ~ **ケ** に、以下の解答群から適切なものを選び、その記号を解答欄に記入してフローチャートを完成させよ。なお、フローチャート中の A には開始時点で逆順にするビット列が設定されており、終了時点では R に結果となるビット列が格納されるものとする。

カ ~ **ケ** の解答群:

- | | |
|--------------------|-------------------|
| (a) A AND 00000001 | (b) A OR 00000001 |
| (c) A AND 10000000 | (d) A OR 10000000 |
| (e) R AND 00000001 | (f) R OR 00000001 |
| (g) R AND 10000000 | (h) R OR 10000000 |
| (i) LSHIFT A | (j) RSHIFT A |
| (k) LSHIFT R | (l) RSHIFT R |

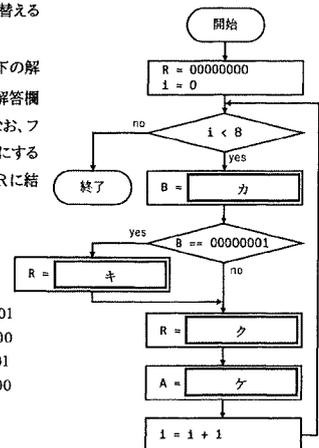


図 III-1. ビット列を逆順に並び替えるフローチャート

演算	説明	例
LSHIFT	与えられたビット列を左に 1 桁ずらす演算である。桁があふれたとき、あふれた桁は廃棄し、新たな桁には 0 が付与される。右例は 01010101 を左に 1 桁ずらした例であり、結果は 10101010 であることを示している。	<pre> 0 1 0 1 0 1 0 1 1 0 1 0 1 0 1 0 </pre>
RSHIFT	与えられたビット列を右に 1 桁ずらす演算である。桁があふれたとき、あふれた桁は廃棄し、新たな桁には 0 が付与される。右例は 01010101 を右に 1 桁ずらした例であり、結果は 00101010 であることを示している。	<pre> 0 1 0 1 0 1 0 1 0 0 1 0 1 0 1 0 </pre>

〔 IV 〕 以下の文章を読んで、設問(A)~(D)に答えなさい。

この問題は、別紙に示す「プログラム表記の例示」にしたがってプログラムを記述する。また、プログラム(コード)を空欄に記述する問題については解答欄あたり 1 行で記述せよ。

ある授業(受講生数 N 人、N は 2 以上)では毎週グループワークを実施する計画であり、受講生を毎回できるだけ異なるグループに分けたい。そこで、受講生をランダムにグループ分けするプログラムを作ろうと考えた。なお、受講生には出席番号 1~N 番が付与されている。

以降の設問における出席番号やグループ番号は 1 から始まっているが、プログラム中の配列の要素番号は 0 から始まっていることに注意せよ。

設問(A)

まず、受講生に順にグループ番号を割り当てるプログラムについて考える。具体的にはグループの数を M として、出席番号順に 1, 2, ..., M, 1, 2, ..., M, 1, 2, ... とグループ番号を割り当てる。そのように動作するプログラム IV-1 を作成した。この例は、6 人を 2 人ずつ 3 つのグループに分ける場合である。解答欄を埋めてプログラムを完成させよ。

```

var students = 6      # 受講生数(N=6)
var groups = 3       # グループの数(M=3)
var count = 1        # グループ番号の初期値
var gID[students]   # 受講生に割り当てるグループ番号
var i
for i = 0 to students - 1
  gID[i] = count
  count = count + 1

  if 
    count = 1
  end
end
                    
```

プログラム IV-1. 初期グループ決定プログラム

設問(B)

次に、プログラム IV-1 が順番に割り当てたグループ番号を、ランダムな順序になるように入れ替える(シャッフルする)方法を考える。シャッフル操作について、具体的な手順を以下に示す。

- 手順 1. 出席番号 1 の受講生に注目する。
- 手順 2. 注目している受講生の出席番号以上の出席番号の受講生からランダムに 1 人選ぶ。もし選ばれたのが注目している受講生であれば何もしない。注目している受講生と違う受講生であれば、その受講生と注目している受講生のグループ番号を交換する。
- 手順 3. もし注目している受講生の出席番号が N-1 であれば処理を終了する。そうでなければ 1 つ大きな出席番号の受講生に注目を移し、手順 2 から繰り返す。

図 IV-1 から図 IV-3 は上記手順に従った操作結果の例である。この例では 6 人を 2 人ずつ 3 つのグループに分けている。

図 IV-1 はプログラム IV-1 の終了後の状態を示す。

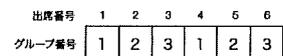


図 IV-1. プログラム IV-1 終了後の状態

図 IV-2 は 1 度目の手順 2 の実施において、出席番号 1 の受講生に注目し、1 以上の出席番号(つまり全員)を対象にしたランダム選択によって出席番号 3 の受講生が選ばれ、出席番号 1 と 3 の受講生のグループ番号が交換された状態を示す。

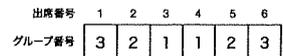


図 IV-2. 1 度目の手順 2 終了後の状態

図 IV-3 はその後、注目が出席番号 2 の受講生に移り、2 度目の手順 2 においてランダムに出席番号 4 の受講生が選択され、出席番号 2 と 4 の受講生のグループ番号が交換された状態を示す。

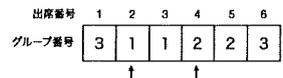


図 IV-3. 2 度目の手順 2 終了後の状態

図 IV-3 の状態では全員のシャッフルが終わっておらず、手順 2 が繰り返される。次に手順 2 を実行した際は出席番号 3 と 5 の受講生のグループ番号が交換された。その次の実行では 4 と 5、最後は 5 と 6 が交換され、プログラムは終了した。最終的に各受講生のグループ番号がどのようになっただか解答欄に記入せよ。

次に、プログラムIV-1の続きとしてシャッフル操作、つまり手順1~3を実現するプログラムを考
えた。手順1~3を以下に再掲する。

- 手順1. 出席番号1の受講生に注目する。
- 手順2. 注目している受講生の出席番号以上の出席番号の受講生からランダムに1人選ぶ。
もし選ばれたのが注目している受講生であれば何もしない。注目している受講生と
違う受講生であれば、その受講生と注目している受講生のグループ番号を交換す
る。
- 手順3. もし注目している受講生の出席番号がN-1であれば処理を終了する。そうでなけ
れば1つ大きな出席番号の受講生に注目を移し、手順2から繰り返す。

そのように動作するよう、プログラムIV-2を作成した。ただし、関数 random()は1から
65536までの整数値をランダムに返す関数とする。

```

var i, random_number
var select1 # 注目している出席番号に相当する配列の要素番号
var select2 # 交換対象となる出席番号に相当する配列の要素番号
for i = 0 to students - 2 # 出席番号 N-1 に相当する配列の要素番号は students-2

    select1 = A
    random_number = random()
    select2 = B

    if select1 != select2
        C
    end
end
    
```

プログラムIV-2. シャッフルプログラム

設問(C)

プログラムIV-2の空欄A~Cについて以下の小問(1)~(3)に答えよ。

- (1) 空欄Aは「注目している受講生の出席番号を設定する」処理である。適切な記述を解答欄
に記入せよ。
- (2) 空欄Bは「グループ番号を交換する対象となる受講生の出席番号を乱数に基づいて算出
する」処理である。適切な記述を以下から1つ選び、その記号を解答欄に記入せよ。

- (ア) i + random_number % (students - (i + 1))
- (イ) i + random_number % (students - i)
- (ウ) i + random_number % (students - (i - 1))
- (エ) i - 1 + random_number % (students - (i + 1))
- (オ) i - 1 + random_number % (students - i)
- (カ) i - 1 + random_number % (students - (i - 1))
- (キ) i + 1 + random_number % (students - (i + 1))
- (ク) i + 1 + random_number % (students - i)
- (ケ) i + 1 + random_number % (students - (i - 1))

- (3) 空欄Cは「gID[select1]の値と gID[select2]の値を入れ替える」処理である。適切な
記述を以下から1つ選び、その記号を解答欄に記入せよ。

- (ア) gID[select1] = gID[select2]
gID[select2] = gID[select1]
- (イ) var temp
gID[select1] = temp
gID[select2] = gID[select1]
gID[select1] = temp
- (ウ) var temp = gID[select1]
gID[select1] = gID[select2]
gID[select2] = temp
- (エ) var temp = gID[select1]
temp = gID[select2]
gID[select1] = temp
gID[select2] = temp

設問(D)

プログラムIV-1, IV-2によってグループのメンバーをランダムに決められるようになった。しかし
偶然とはいえ「新しいグループの受講生全員が、前回のグループ分けでも同じグループだった」と
いう状況は避けたい。そこで、そのような状況のグループがいくつ生じたかを数える処理の手順を
検討した。ここでは8人の受講生を3グループに分ける状況を想定し、そこで必要となる4つの変
数の初期状態の例を図IV-4に示す。

出席番号	1	2	3	4	5	6	7	8
新グループ	2	2	2	1	3	1	1	3
出席番号	1	2	3	4	5	6	7	8
旧グループ	3	3	1	1	2	1	2	2
グループ番号	1	2	3	グループ番号	1	2	3	
メモ	0	0	0	フラグ	0	0	0	

図IV-4. 4つの変数とその初期状態

それらの値を見れば、出席番号5と8
の受講生は新グループが3、旧グループ
が2で同じであり、「新しいグループの受講生全員が、前回のグループ分けでも同じグループだ
った」状況にあることがわかる。それ以外にそのような状況はない。

配列変数「メモ」および「フラグ」の扱い方は以下の手順の中で説明する。なお、メモとフラグの全
要素の初期値は0である。

検討した処理の手順をおおまかに述べる。

- 手順1. 出席番号1の受講生に注目する。
- 手順2. 注目している受講生の新グループ番号をng、旧グループ番号をogに取り出す。
- 手順3. メモのng番要素の値に応じて以下の手順3.1~3.3のいずれかを実施する。
- 手順3.1 もしメモのng番要素の値が初期値だった場合はメモのng番要素にogを記録する。
(これで少なくともそのグループに旧グループがogのメンバーが1人は居たことがわかる)
- 手順3.2 もしメモのng番要素の値がogに一致すれば何もしない。
- 手順3.3 メモのng番要素の値がogに一致しなければフラグのng番要素に1を記録する。
(これで少なくともそのグループに旧グループが異なるメンバーが居たことがわかる)
- 手順4. もし注目している受講生の出席番号がN-1であれば手順5に進む。
そうでなければ1つ大きな出席番号の受講生に注目を移し、手順2から繰り返す。
- 手順5. フラグを参照して、「新しいグループの受講生全員が、前回のグループ分けでも同じグ
ループだった」数を求め、処理を終了する。

プログラムIV-3に、前のページで説明した処理を行うプログラムを示す。解答欄を埋めてプロ
グラムを完成させよ。プログラムに登場する変数 students, groups の意味は設問(A)~(C)の
それらと同じである。配列変数 gID と oldGID が、これまでの説明の「新グループ」と「旧グループ」
に対応する。配列変数 memo, flag は同じく「メモ」「フラグ」に対応する。変数 same は手順5で
求めた数に対応する。

```

# 配列変数 gID, oldGID はあらかじめ定義され初期値が与えられている
# 配列変数 memo, flag はあらかじめ定義され初期値として0が与えられている
var i, ng, og
# 受講生を順に調べていく
for i = 0 to students - 1
    ng = gID[ i ] # その受講生の新グループ番号を得る

    og = 
    if # メモが初期値のままであれば
        else
            if
            end
        end
    end
end
# 全員が同じ旧グループだと判定された新グループの数を表示する
var same = 0
for i = 0 to groups - 1
    if
    end
end
print("same member case : ", same)
    
```

プログラムIV-3. 新グループの全員が同じ旧グループだったことを調べる処理

[V] 以下の文章を読んで、設問(A)と設問(B)に答えなさい。

学生 X は、近年の日本人の旅行・観光の実態を知るために「日本人の旅行・観光における消費の動向」について調べることとした。そこで学生 X は、観光庁が実施した「旅行・観光消費動向調査」から、2019 年から 2023 年まで(以後、5年間と表記)の日本人の国内旅行の延べ旅行者数、平均旅行回数、平均泊数、旅行単価に関する統計情報を収集した。ただし、100 万人以下は四捨五入した。学生 X は、国内旅行に関して、旅行種類の2区分(宿泊、日帰り)と旅行目的の3区分(観光・娯楽、帰省・知人訪問等、出張・業務)に関する延べ旅行者数を表 V-1 として整理し、どのような目的で旅行・観光を行っているのか分析した。

表 V-1. 国内旅行における旅行種類と旅行目的ごとの延べ旅行者数 [単位: 百万人]

年	国内旅行(宿泊)				国内旅行(日帰り)			
	観光・娯楽	帰省・知人訪問等	出張・業務	合計	観光・娯楽	帰省・知人訪問等	出張・業務	合計
2023	169	69	43	281	159	32	26	217
2022	145	58	30	233	139	27	19	185
2021	80	43	19	142	84	20	22	126
2020	92	46	23	161	89	23	20	132
2019	172	86	54	312	196	41	38	275

設問(A)

- 表 V-1 において、2023 年に観光・娯楽目的で国内旅行(宿泊)に行った延べ旅行者数は何人が解答欄に記入せよ。
- 表 V-1 において、旅行目的ごとに 2019 年の延べ旅行者数をそれぞれ基準としたとき、旅行目的ごとの基準に対する延べ旅行者数の割合が最も低い旅行目的とその年を知りたい。そこで、基準に対する延べ旅行者数の割合を旅行目的ごとに調べ、その割合が最も低い旅行目的は、どの年のどちらの旅行種類のどの旅行目的であるか解答欄に記入せよ。また、その割合を小数点以下第 2 位まで求めて解答欄に記入せよ。
- 表 V-1 から、年・旅行種類(宿泊、日帰り)を問わず、旅行目的としての割合が最も高いのは常に観光・娯楽である。そこで、5 年間のうち、どの年のどちらの旅行種類の観光・娯楽が、旅行目的として最も割合が高かったかを調べ、その年と旅行種類を解答欄に記入せよ。また、その割合を小数点以下第 2 位まで求めて解答欄に記入せよ。

情報

次に学生 X は、5年間の日本人の旅行回数と各旅行時の費用について調べるために、国内旅行の平均旅行回数、平均泊数、旅行単価に関して、統計情報を整理することにした。表 V-2 は、それぞれの5年間の平均値とその標準偏差と変動係数、最大値と最小値、最大値から最小値を引いた値(最大-最小と表記)を示している。ただし、変動係数とは、標準偏差を平均で割った値のことであり、相対的なばらつきを大きさの指標である。変動係数の値が大きいほど、平均値に対して値がばらついていることを示す。表 V-3 は平均旅行回数(宿泊)、旅行単価(宿泊)、平均旅行回数(日帰り)、旅行単価(日帰り)、平均泊数(宿泊)の5項目の間の5年間のデータの変化に関して相関係数を計算した結果である。図 V-1 は、表 V-3 に示した相関係数の導出に用いた5項目の間の5年間のデータの変化を示した図である。なお、表 V-3 と図 V-1 で用いられる単位「人回」とは、同じ人が 1 回の旅行で複数の目的地を訪れてもそれぞれ回ずつとして数えることを指す。つまり、その目的地の延べ訪問回数である。したがって、円/人回と泊/人回はそれぞれ、ある目的地を訪れた延べ回数を分母としたときの、その地での旅行単価と宿泊日数を表す。

表 V-2. 5 項目の 5 年間の平均値とその標準偏差と変動係数、最大値と最小値、最大値から最小値を引いた値

項目	平均値	標準偏差	変動係数	最大値	最小値	最大-最小
平旅行(宿)	1.80	0.59	0.33	2.47	1.13	1.34
旅単(宿)	54996	5681	0.10	63253	48361	14892
平旅行(帰)	1.49	0.49	0.33	2.18	1.01	1.17
旅単(帰)	17767	884	0.05	19027	16596	2431
平泊	2.30	0.10	0.04	2.43	2.11	0.32

表 V-3. 5年間のデータに関する5項目間の相関係数

項目	平旅行(宿)	旅単(宿)	平旅行(帰)	旅単(帰)	平泊
平旅行(宿)	1.00	0.76	0.98	0.53	-0.82
旅単(宿)		1.00	0.62	0.94	-0.99
平旅行(帰)			1.00	0.37	-0.69
旅単(帰)				1.00	-0.90
平泊					1.00

(注)表 V-2、表 V-3 内の項目において、平旅行(宿)は平均旅行回数(宿泊)(回/人)、旅単(宿)は旅行単価(宿泊)(円/人回)、平旅行(帰)は平均旅行回数(日帰り)(回/人)、旅単(帰)は旅行単価(日帰り)(円/人回)、平泊は平均泊数(宿泊)(泊/人回)のことをそれぞれ意味する。

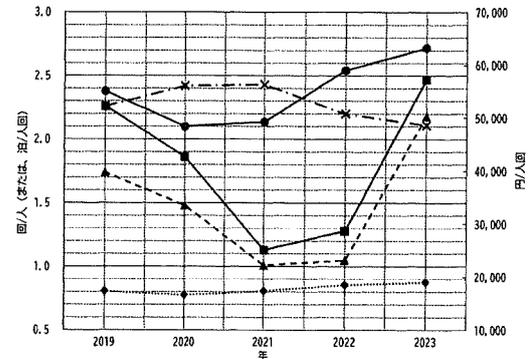


図 V-1. 平均旅行回数(宿泊)、旅行単価(宿泊)、平均旅行回数(日帰り)、旅行単価(日帰り)、平均泊数(宿泊)の5項目の5年間のデータの変化

設問(B)

表 V-2、表 V-3、図 V-1 から読み取れることを、次の解答群からすべて選び、それらの番号を解答欄に記入せよ。

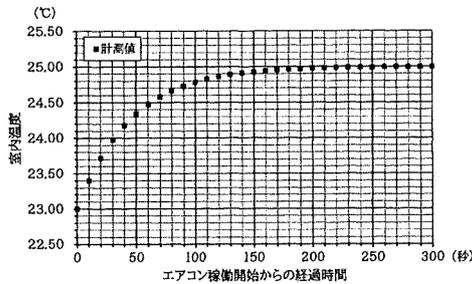
設問(B)の解答群

- 5項目の中で、旅行単価(宿泊)の標準偏差が最も大きいので、旅行単価(宿泊)が平均に対する値のばらつき方が最も大きい。
- 平均旅行回数(宿泊)の最大値-最小値の値は、平均旅行回数(日帰り)の最大値-最小値の値より大きいので、平均旅行回数(宿泊)は、平均旅行回数(日帰り)と比較して、平均値に対して値がばらついている。
- 5項目の中で、平均泊数(宿泊)のデータは変動係数が最も小さいので、平均値に対して値のばらつき方が最も小さい。
- 平均旅行回数(宿泊)と平均旅行回数(日帰り)の間には、高い相関がある。
- 旅行単価(宿泊)と平均泊数(宿泊)の間には、相関はない。

[VI] 以下の文章を読んで、設問(A)と設問(B)に答えなさい。

ある工場の倉庫では、倉庫の室内温度を設定温度 25.00℃に保つように冷暖房を行うエアコンを稼働させている。また、室内には温度計が設置されており、室内温度を 10 秒間隔で計測した値を知ることができる。

室内温度が 23.00℃で均一の状態からエアコンを稼働開始し、10 秒ごとの室内温度を計測したところ、下の図VI-1に示す計測結果のグラフが得られた。



図VI-1. 倉庫の室内温度の時間変化

この計測結果から、エアコン稼働開始後すぐは 10 秒間での室内温度の変化は大きく、室内温度がエアコン設定温度(25.00℃)に近づくとともに、10 秒間の室内温度の変化が小さくなるのがわかる。そこで、エアコン稼働開始時を 0 秒として、10 秒ごとの室内温度をシミュレートするモデルを作成するために、各計測時点における室内温度とエアコン設定温度との差、10 秒間の室内温度の変化との関係を調べることにした。例えば、エアコン稼働開始時(図VI-1の0秒)における室内温度(23.00℃)とエアコン設定温度(25.00℃)との差は、2.00℃であり、0 秒と 10 秒の室内温度の差は 0.40℃程度と読み取れる。

以降では、エアコン稼働開始時を計測番号 0 番として、計測番号 n 番 ($n = 0, 1, 2, \dots$) における室内温度を T_n で表すモデルを考える。図VI-2より、各計測時点から 10 秒後にかけての室内温度の上昇が、室内温度とエアコン設定温度との差と強く相関していることが読み取れる。そこで、各計測時点と 10 秒後との間の室内温度の差が、その計測時点における室内温度とエアコン設定温度(25.00℃)との差に比例するモデルを考えることにした。この比例定数を、1 秒あたりの温度変化率 α で表し、 T_{n+1} を求めるモデル式を、下のモデル 1 のように考えた。

モデル 1 内の [オ] に入る適切なものを、下の解答群 (1) ~ (5) の中から 1 つ選び、その番号を解答欄に記入せよ。

$$T_{n+1} = T_n + (25.00 - \text{オ}) \times \alpha \times 10 \text{ (℃)} \quad \dots \text{モデル 1}$$

[オ] の解答群:

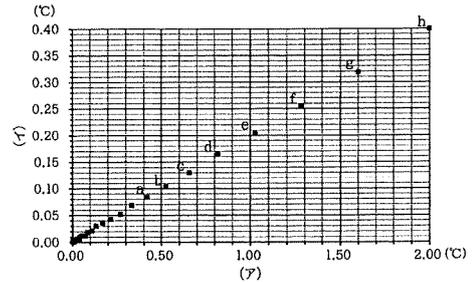
- [(1) T_0 (2) T_1 (3) T_{n-1} (4) T_n (5) T_{n+1}]

下の表VI-1は、モデル 1 のシミュレーション結果である。この表から、モデル 1 の温度変化率 α の値は [カ] であることがわかる。[カ] の解答欄に α の値を記入せよ。

表VI-1. モデル 1 のシミュレーション結果

計測番号 n	エアコン稼働開始からの経過時間(秒)	室内温度 T_n (℃)
0	0	23.00
1	10	23.40
2	20	23.72
⋮	⋮	⋮

このような関係を図VI-1の各計測データに対して調べて描画したグラフが、下の図VI-2である。



図VI-2. 温度差の関係性を表すグラフ

設問(A)

図VI-2のグラフの横軸(ア)と縦軸(イ)として適切なものを、下の解答群(1)~(4)の中からそれぞれ1つずつ選び、その番号をそれぞれの解答欄に記入せよ。

(ア)・(イ)の解答群:

- (1) 各計測時点の室内温度 - 設定温度(25.00℃)
- (2) 設定温度(25.00℃) - 各計測時点の室内温度
- (3) 各計測時点の室内温度 - 各計測時点から 10 秒後の室内温度
- (4) 各計測時点から 10 秒後の室内温度 - 各計測時点の室内温度

ここで、図VI-2のグラフ内に示した点 a ~ h の中で、図VI-1におけるエアコン稼働開始から 10 秒後の計測結果に対応する点は [ウ] であり、点 e は図VI-1におけるエアコン稼働開始から [エ] 秒後の計測結果に対応する。[ウ] の解答欄には a ~ h の中から適切なアルファベットを記入し、[エ] の解答欄には数値を記入せよ。

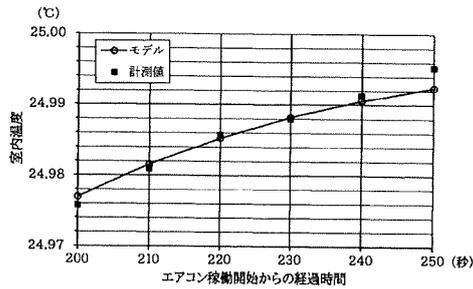
次に、室内温度が 25.00℃で一定となった後に倉庫のドアを開けて物資の出し入れを行い、作業後にドアを閉めたところ、室外からの暖かい空気の流入により、室内温度が 26.00℃に上昇した。このとき、エアコンは室内温度を設定温度 25.00℃ に近づけるように冷房運転に切り替わる。この冷房運転時の室内温度をモデル 1 によってシミュレートするとき、温度変化率 α がとり得る値として正しいものを下の解答群 (1) ~ (3) の中から 1 つ選び、その番号を [キ] の解答欄に記入せよ。

[キ] の解答群:

- [(1) プラスの値 (2) マイナスの値 (3) ゼロ]

また、 α の値が [カ] と同じ絶対値をもつとき、冷房運転時のモデル 1 の室内温度の算出結果が初めて 25.50℃を下回るのは、冷房運転開始時から [ク] 秒後である。[ク] の解答欄に秒数を数値で記入せよ。ただし、温度算出は 10 秒間隔で行うものとする。

設問(A)で作成したモデル1が室内温度の計測値をどの程度よく表しているかを検討するため、各計測時点での室内温度計測値と、モデルにより計算した室内温度の値の差(残差)を調べた。下の図VI-3は、図VI-1の200~250秒の範囲を拡大し、計測値とモデル計算値を重ねて描画したグラフである。

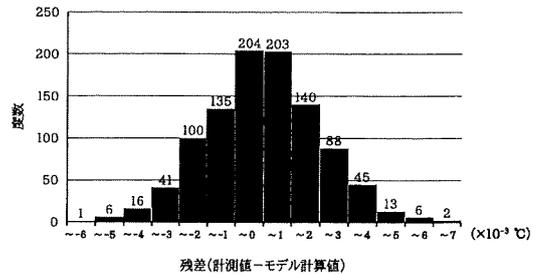


図VI-3. 室内温度の計測値とモデル計算値を重ねて示したグラフ

設問(B)

図VI-3のグラフにおいて計算される残差の中で絶対値が最も大きいのは、エアコン稼働開始からの経過時間が 秒の計測時点における残差である。 の解答欄に秒数を数値で記入せよ。

次に、合計1000個の計測時点における室内温度計測値を得て、モデル1の計算値との残差を求めてヒストグラムに表したところ、図VI-4が得られた。ただし、計測値からモデル計算値を引いた値を残差とした。また、「 $\sim a$ 」の表記は「 $a-1$ より大きく a 以下」であることを表し、ヒストグラムの各棒の上に記した数値は、ヒストグラムの各階級の範囲に含まれる度数を表している。



図VI-4. 合計1000個の室内温度計測値に対する残差のヒストグラム
ただし、 $\sim a$ は、 $a-1$ より大きく a 以下であることを表す

図VI-4のヒストグラムから、残差は 0.000°C を中心としてほぼ左右対称に度数が分布しており、 0.000°C に近い値ほど度数が多く存在していることがわかる。したがって、モデル1は室内温度の計測値に近い値を高い確率で算出できるモデルとなっており、計測値より高い値と低い値をほぼ同じ割合で算出する適切なモデルであると考えられる。特に、図VI-4では、計測値がモデル計算値より高く、(モデル計算値 +) $^{\circ}\text{C}$ 以下の範囲に 個のデータが存在し、計測値が(モデル計算値 -) $^{\circ}\text{C}$ より高く、モデル計算値以下の範囲に 個のデータが存在していることがわかる。また、今回のシミュレーションにおけるモデル計算値全体の98.5%が、(計測値 -) $^{\circ}\text{C}$ 以上で(計測値 +) $^{\circ}\text{C}$ より低い温度の範囲に入っていることがわかる。

と の解答欄にそれぞれのデータの個数を数値で記入し、 の解答欄に温度の値を記入せよ。

情報 (1/29実施分)

(I)	設問(A)	ア	イ	ウ	エ	オ	設問(B)	カ	キ
		9	2	3	4	8		6	2
(II)	設問(A)	10	設問(B)	9					
	設問(C)	1	2	3	4				
(III)	設問	1.	ア	イ			ウ	エ	オ
			83	000000010			1	1	0
		3.	カ	キ	ク	ケ			
		a	f	k	j				
(IV)	設問(A)	<pre> var students = 6 # 受講生数(N = 6) var groups = 3 # グループの数(M = 3) var count = 1 # グループ番号の初期値 var gID[students] # 受講生に割り当てるグループ番号 var i for i = 0 to students - 1 gID[i] = count count = count + 1 if count > groups count = 1 end end </pre>							
	設問(B)	出席番号	1	2	3	4	5	6	
		グループ番号	3	1	2	1	3	2	
	設問(C)	(1) select1 =	i			(2) イ	(3) ウ		
設問(D)	<pre> # 配列変数 gID, oldGID はあらかじめ定義され初期値が与えられている # 配列変数 memo, flag はあらかじめ定義され初期値として 0 が与えられている var i, ng, og # 受講生を順に調べていく for i = 0 to students - 1 ng = gID[i] # その受講生の新グループ番号を得る og = oldGID[i] </pre>								

(IV)	設問(D)	<pre> if memo[ng-1] == 0 # メモが初期値のままであれば memo[ng-1] = og else if memo[ng-1] != og flag[ng-1] = 1 end end end end # 全1が同じ旧グループだと判定された新グループの数を表示する var same = 0 for i = 0 to groups - 1 if flag[i] == 0 same = same + 1 end end print ("same member case : ", same) </pre>											
	設問(A)	1.			1	6	9	0	0	0	0	0	0
(V)	設問(A)	2.	2021	(国内旅行) 宿泊	出張・業務	割合		0.35					
		3.	2022	(国内旅行) 日帰り	割合	0.75							
		設問(B)	(3), (4)										
(VI)	設問(A)	(2)	(4)	g	30	(4)	0.02	(1)	40				
	設問(B)	ケ	コ	サ	シ								
		250	343	339	0.005 (または5×10 ⁻³)								