

# 2020年度 A O入試 1次選考問題

## 情報理工学部

# 情報科目

### 注意事項

1. 試験開始の合図があるまで、問題用紙を開いてはいけません。
2. 解答はすべて、所定の解答用紙に記入してください。
3. 解答用紙に受験番号と氏名（フリガナ）を記入してください。
4. 解答時間は60分です。問題は12ページあります。
5. 問題用紙・解答用紙および計算用紙はすべて回収します。一切持ち帰ってはいけません。

〔 I 〕 次の (1) ~ (5) の説明文に該当する用語を、下記枠内に記したのから選択し、ア~コの記号で答えなさい。

- (1) 近頃の CPU (中央演算装置) は、1 個のパーツの中に複数個の CPU に相当する回路が組み込まれている。
- (2) 汎用的な処理ができる CPU に対して、ベクトルや行列の演算に特化した性能を発揮するプロセッサが、CG 処理や AI 処理などに活用されている。
- (3) コンピュータの画面表示やプリンタでの印刷において、文字の表示・印刷サイズを拡大縮小しても、文字の形が滑らかで綺麗に拡大縮小できる字体データの形式がある。
- (4) コンピュータどうしの通信には、通信し合うための取り決めがあり、それに従って通信することが多い。
- (5) プログラミング言語の処理系には、事前に機械語に翻訳するコンパイラ以外に、命令を逐次解釈しながら動作するものがある。

|                                   |   |          |
|-----------------------------------|---|----------|
| ア. GPU (Graphics Processing Unit) | イ. マルチレイヤ                               | ウ. アセンブラ |
| エ. アウトラインフォント (ベクターフォント)          | オ. マルチコアプロセッサ                           |          |
| カ. インタプリタ                         | キ. FPGA (Field-Programmable Gate Array) | ク. プロトコル |
| ケ. ビットマップフォント (ラスターフォント)          | コ. マージン                                 |          |

- 〔Ⅱ〕 36の正の約数の集合を  $D = \{1, 2, 3, 4, 6, 9, 12, 18, 36\}$  とし、二項関係  $m \leq n$  は  $m$  が  $n$  の約数であると定義する。この関係  $\leq$  を図示したものが図 II-1 です。図 II-1 における関係  $\leq$  について、設問 (A) ~ 設問 (C) に答えなさい。

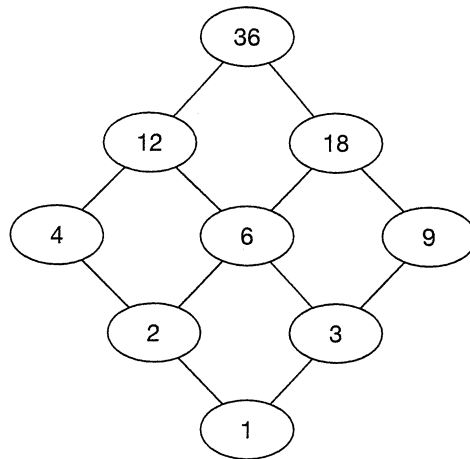


図 II-1 集合  $D$  における関係  $\leq$  の構造

設問 (A)

関係  $\leq$  において  $x \leq 9$  または  $9 \leq x$  が成立しない  $D$  の要素  $x$  を全て答えなさい。

設問 (B)

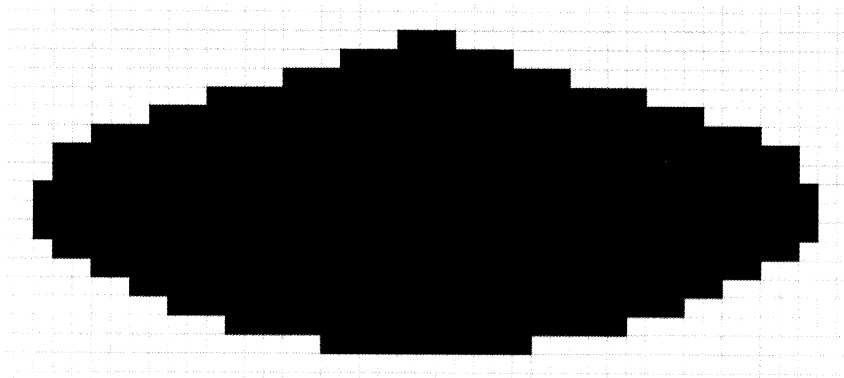
集合  $M_1 = \{2, 3, 6\}$  としたとき、関係  $\leq$  において  $M_1$  の任意の要素  $x$  に対して  $x \leq y$  となるような  $D$  の要素  $y$  を全て答えなさい。

設問 (C)

集合  $M_2 = \{4, 18\}$  としたとき、関係  $\leq$  において  $M_2$  の任意の要素  $x$  に対して  $z \leq x$  となるような  $D$  の要素  $z$  を全て答えなさい。

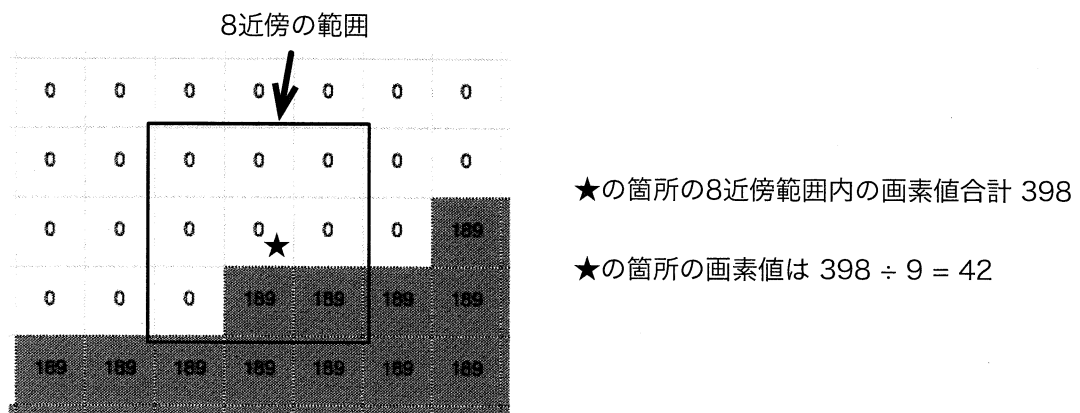
〔Ⅲ〕 次の文章を読んで、設問（A）と設問（B）に答えなさい。

図Ⅲ-1のような四角い画素で構成されたグラフィックス画面において、図形の境界線が曲線や斜めの直線の場合にジャギー（ギザギザ部分）が見えることがある。このジャギーをなるべく目立たなくするように工夫して画面表示させたい。



図Ⅲ-1 ジャギーのあるグラフィックス図形の例

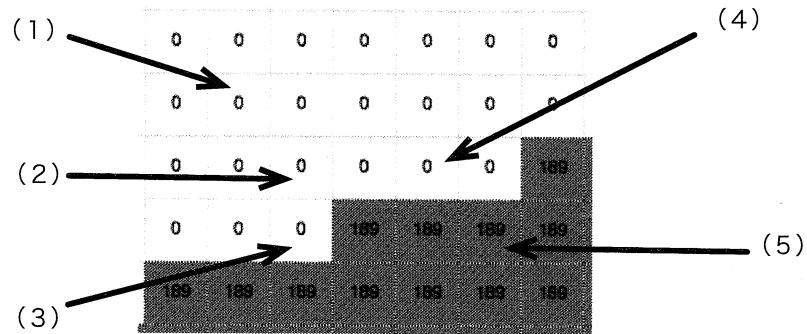
ここではまず、図Ⅲ-2のように、とある画素値（色の濃さ）を持つ画素（図中では★）において、その8近傍（枠で囲われた範囲）との合計9画素の画素値の平均値を求めます。その平均値を、その★の箇所の画面表示する画素値とし、画面全体で各画素値を計算して、ジャギーを目立たなくする方法を考えます。なお、ここでの画素値は大きいほど黒色に近い色を表すものとし、この条件と手法について、次の設問（A）と（B）に答えなさい。



図Ⅲ-2 とある画素の8近傍画素からの色の濃さ計算例

設問 (A)

前述の図Ⅲ-2を参考にして、次の図Ⅲ-3中に示す(1)～(5)の各画素の画素値(色の濃さ)を計算して答えなさい。



図Ⅲ-3 色の濃さを計算する画素値の位置

設問 (B)

上記の手法をコンピュータ上で処理するにあたり、その処理を高速に実行するための工夫(高速化の方法)について検討して答えなさい。なお、その工夫は、ハードウェアに関することでも、ソフトウェアに関することでも構わないが、高速化の方法についてあなたの考えを答えること。

〔IV〕 以下の文章を読んで、設問（A）～ 設問（E）に答えなさい。ただし、この問〔IV〕および次の問〔V〕では、問題冊子の最後に示す記法を使ってプログラムを記述する。

あなたは 2 学期の日直の順番を公平に決めるよう担任の先生に依頼されました。そこであなたはクラスのメンバー全員に日直の順番をランダムに決めるプログラムを作ろうと考えました。そのプログラムは、クラスのメンバー全員に対して、ランダムに順番を決定し、決定した順番に従って出席番号を表示させるプログラムです。プログラムが表示した出席番号順に日直をしてもらおうと考えました。なお、クラスのメンバーにはそれぞれ出席番号が 1 から順に割り振られているものとします。

#### 設問（A）

まずクラスのメンバーが 4 人であったケースについて考えよう。このケースの場合、日直の順番は何通りあるかを答えなさい。ただし、(1, 2, 3, 4)と(2, 3, 4, 1)は異なる順番とする。

#### 設問（B）

次に日直の順番を決めるプログラムの前に、ランダムな値を活用するプログラムの練習問題に取り組みました。練習問題として、1 から 6 までのサイコロの値をランダムに 10 回表示するプログラムを作成しました。作成したプログラムを図 1 に示します。ただし図 IV-1 中の random()は、0 から 65355 までのランダムな整数値(乱数)を戻り値とする関数とする。

なお、このプログラムでは、ランダムにサイコロの値を決定するために、乱数のある数 D で割った余りに 1 を加えたものをサイコロの値としています。この時、図 IV-1 のプログラムの変数 D に代入する値を答えなさい。

```
C = 10  
D =   
for (i = 0; i < C ; i = i + 1)  
    randomNumber = random()  
    j = (randomNumber % D) + 1  
    print(j)  
end
```

図 IV-1 サイコロプログラム

### 設問 (C)

そして、クラスのメンバーが 30 人であった場合の日直の順番をランダムに決定する図 IV-2 のプログラムを作成しました。なお、図 IV-2 中の random() は図 1 中の random() と同じ関数です。

この図 2 のプログラムの手順は 次の(1)から(6)までで表せます。

- (1) 変数  $T$  にクラスのメンバーの人数である 30 を代入する
- (2) 日直の順番を格納する配列  $order$  に初期状態の出席番号を入力する
- (3) まだ順番が決まっていないメンバーからランダムに一人選ぶ
- (4) (3) で選ばれたメンバーの順番を残っている順番の中から最も後の順番とする
- (5) (3) と(4) をすべて人の順番が決まるまで繰り返す
- (6) 決定した順番を表示する。

図 IV-2 中の変数  $j$  に代入する式を答えなさい。

```
T = 30
for (i = 0; i < T; i = i + 1)
    order[i] = i + 1
end
num = T
for (i = 0; i < num; i = i + 1 )
    randomNumber = random()

    j = 

    tmp = order[j]
    order[j] = order[num-i-1]
    order[num-i-1] = tmp
end
for (i = 0; i < T; i = i + 1)
    print(order[i])
end
```

図 IV-2 日直の順番決めプログラム

### 設問 (D)

図 IV-2 のプログラムが想定通りに動作するかを確認するために、図 IV-2 のプログラムに追加修正を行いました。追加修正後のプログラムを図 IV-3 に示します。追加修正した 2 つの点は、1 行目の変数  $T$  に代入する値を 4 に修正した点と、変数  $i$  と  $j$  の値を表示する print 文を 9 行目に追加した点です。なお、8 行目にある四角の空白で示された変数  $j$  に代入する式は設問 (C) の問いと同一の式です。

図 IV-3 のプログラムを実行した結果、図 IV-4 に示す実行結果になりました。図 IV-4 中の四角の空白で示された内容を答えなさい。

```
T = 4
for (i = 0; i < T; i = i + 1)
    order[i] = i + 1
end
num = T
for (i = 0; i < num; i = i + 1 )
    randomNumber = random()

    j = 

    print(i, j)
    tmp = order[j]
    order[j] = order[num-i-1]
    order[num-i-1] = tmp
end
for (i = 0; i < T; i = i + 1)
    print(order[i])
end
```

図 IV-3 日直の順番決めプログラム追加修正版

```
0 2
1 1
2 1
3 0

```

図 IV-4 実行結果



### 設問 (E)

図 IV-2 のプログラムを作ろうとしたところ、当初 1 カ所誤ったプログラムを作成してしまった。誤った箇所は 5 行目であり、「num = T」を誤って、「num = T - 1」としてしまった。

この誤ったプログラムを実行した場合、どのような順番を生成するかを文章で説明しなさい。

[V] 以下の文章を読んで、設問 (A) ～ 設問 (C) に答えなさい。ただし、この問 [V] では、問題冊子の最後に示す記法を使ってプログラムを記述する。

N 次の多項式をプログラムで表すために、要素数 N + 1 の配列を用意し、 $x^k$  の係数を配列の k 番目の要素とする。たとえば、 $2x^2 - x + 3$  は、3, -1, 2 を順に含む要素数 3 の配列で表現できる。

設問 (A)

2 つの多項式を足し合わせて新しい多項式を表すことを考えよう。要素数 M の配列 f と g がそれぞれ多項式を表しているとする。この 2 つの多項式を足し合わせた結果の多項式を、同じく要素数 M の配列 h が表すようにしたい。配列 f と g から配列 h の各要素を計算するプログラムを記述しなさい。ただし、配列 f, g, h と変数 M はすでに用意されており、必要に応じて変数 i, j, t を利用してよいとする (すべてを利用する必要はない)。

設問 (B)

2 つの多項式の間で掛け算を行うことを考えよう。たとえば、多項式  $2x^2 - x + 3$  と  $x + 2$  を掛け合わせると、次のように  $2x^3 + 3x^2 + x + 6$  が得られる。

$$\begin{aligned} (2x^2 - x + 3)(x + 2) &= x \cdot (2x^2 - x + 3) + 2 \cdot (2x^2 - x + 3) \\ &= 2x^3 - x^2 + 3x + 4x^2 - 2x + 6 \\ &= 2x^3 + 3x^2 + x + 6 \end{aligned}$$

同様の結果を、以下のように展開式に従った係数の操作で導くことができる。

|                          |    |       |       |     |   |                         |
|--------------------------|----|-------|-------|-----|---|-------------------------|
|                          |    | $x^3$ | $x^2$ | $x$ | 1 |                         |
| $x \cdot (2x^2 - x + 3)$ | →  | 2     | -1    | 3   | 0 |                         |
| $2 \cdot (2x^2 - x + 3)$ | →  | 0     | 4     | -2  | 6 |                         |
|                          | +) |       |       |     |   |                         |
|                          |    | 2     | 3     | 1   | 6 | → $2x^3 + 3x^2 + x + 6$ |

要素数 K の配列 f が表す多項式と、要素数 L の配列 g が表す多項式を掛け合わせた結果の多項式を、要素数 M の配列 h が表すようにしたい (ただし、 $M = K + L - 1$  である)。上で示した係数の操作に基づいて、配列 f と g から配列 h の各要素を計算するプログラムを記述しなさい。ただし、配列 f, g, h と変数 K, L, M はすでに用意されているとする。また、必要に応じて変数 i, j, t を利用してよい (すべてを利用する必要はない)。

### 設問 (C)

要素数  $M$  の配列  $\mathbf{a}$  が多項式  $P(x)$  を表現しているとして、変数  $x$  に具体的な値を代入して多項式の値を計算したい。ただし  $x^k$  の部分の計算は、次のように行くと乗算の回数を減らすことができる。まず、多項式  $P(x)$  を考える。

$$P(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_2 x^2 + a_1 x + a_0$$

これを次のように変形すると、乗算の回数が  $n$  回になることがわかる。

$$P(x) = ((\cdots ((a_n x + a_{n-1})x + a_{n-2})x + \cdots + a_2)x + a_1)x + a_0$$

- (1) 多項式  $Q(x) = 2x^4 + x^3 - x^2 + 5x - 4$  について、上のような式の変形を行なった結果を示しなさい。また、式の変形後、多項式の値を求めるために乗算を何回行う必要があるか答えなさい。
- (2) この式の変形を利用する方法によって多項式の値を求めるプログラムを記述しなさい。多項式を表す配列  $\mathbf{a}$  と変数  $M$  はすでに用意されており、変数  $x$  にも具体的な値が代入されているとする。結果の値は、プログラムが終了した時に変数  $y$  に格納されているようにしなさい。なお、必要に応じて変数  $i$ ,  $j$ ,  $t$  を利用してよい (すべてを利用する必要はない)。

## プログラムの記法の説明

間〔IV〕〔V〕でプログラムの記述に用いる記法について説明する。

### 文

---

|                                       |   |
|---------------------------------------|---|
| 変数 = 式                                | 変数に式の値を代入する。以下の説明ではこの文を「代入文」と呼ぶ。  |
| for (代入文 1; 条件式; 代入文 2)<br>...<br>end | まず、代入文 1 を実行し、条件式を評価する。条件式が偽であれば何もしない。条件式が真のとき、end までの命令を実行し、次に代入文 2 を実行する。その後、条件式が真である間、end までの命令と代入文 2 を実行する。                 |
| while 条件式<br>...<br>end               | 条件式が真である間、end までの命令を実行する。   |
| if 条件式<br>...<br>else<br>...<br>end   | 条件式が真のとき、else までの命令を実行し、偽のとき、else から end までの命令を実行する。else を記述しない場合、条件式が真のときに end までの命令を実行し、条件式が偽のときは何もしない。                       |
| break                                 | for 文、または while 文（これらをループ文と呼ぶ）の内部でのみ利用できる。実行すると繰り返しの処理を打ち切り、ループ文の次の文の実行に移る。ループ文の中でループ文が使われている（ネストしている）時は、一番内側のループ文の処理だけが打ち切られる。 |
| print(式)                              | 式の値を表示する。   |

---

### 配列

配列名 [式]           含まれる要素の数があらかじめ決められた配列を利用できる。配列 **a** の要素数が **N** (**N** は正の整数) のとき、配列 **a** は 0 番目から (**N**-1) 番目までの要素を持ち、**i** 番目の要素は **a[i]** という記法で表現する。

### 比較演算子

|        |                  |
|--------|------------------|
| A == B | A と B の値が等しい。    |
| A != B | A と B の値が等しくない。  |
| A <= B | A の値が B の値以下である。 |
| A < B  | A の値が B の値より小さい。 |
| A >= B | A の値が B の値以上である。 |
| A > B  | A の値が B の値より大きい。 |

## 算術演算子

|   |  |
|---|--|
| + | 加算（足し算）を行う。  |
| - | 減算（引き算）を行う。  |
| * | 乗算（掛け算）を行う。  |
| / | 除算（割り算）を行う。ただし、結果は実数で表される。                                       |
| % | 剰余算を行う。剰余算は割り算の余りを求める。例えば <code>7%3</code> は <code>1</code> となる。 |

複数の算術演算子が混在した式では `*` と `/` と `%` の計算が `+` や `-` よりも先に行われる。算術演算子と比較演算子が混在した式では、算術演算子が先に計算される。また、式の中で `( )` を使い、計算の順序を示すことができる。

## プログラムの例

```
s = 0
i = 1
while s <= 100
    s = s + i
    print(s)
    i = i + 1
end
```

(1) `1+2+3+...` と加算を繰り返して、その値を表示する。合計が `100` を超えたら終わる。

(2) 与えられた要素数 `n` の配列 `a` の内容を、同じ大きさの配列 `b` にコピーする。ただし、配

```
for (i = 0; i < n; i = i + 1)
    if a[i] >= 0
        b[i] = a[i]
    else
        b[i] = 0
    end
end
```

列 `a` の要素で負の数があれば、`b` には代わりに `0` を代入する。