

令和7年度(令和7年4月入学)
博士前期課程(修士課程)一般入試(第I期)
令和6年度(令和6年秋入学)
博士前期課程(修士課程)外国人留学生特別入試
情報工学専攻

数 学 (120分)

〔注意事項〕

1. 監督者の指示があるまで、問題冊子(この冊子)を開いてはいけません。
2. 配布物は、この問題冊子1部、解答用紙3枚と計算用紙1枚です。
3. 解答用紙には志望専攻名、受験番号を記入する欄がそれぞれ1箇所ずつあります。監督者の指示に従って、すべての解答用紙(合計3枚)の志望専攻名欄と受験番号欄に志望専攻名と受験番号を記入しなさい。
4. 解答は、問題番号に対応する解答用紙の指定された場所に書きなさい。解答を解答用紙の裏面に書いてはいけません。解答用紙、計算用紙の追加、交換はしません。
5. 問題は全部で3問あり、2ページにわたって印刷されています。落丁・乱丁および印刷の不鮮明な箇所などがあれば、手をあげて監督者に知らせなさい。
6. 問題冊子の白紙と余白は、計算などに使用してもよろしい。
7. 解答用紙は、持ち帰ってはいけません。
8. 問題冊子と計算用紙は、持ち帰りなさい。

- 1** (1) a を実数とする。数ベクトル空間 \mathbb{R}^3 において、5つのベクトル

$$\begin{pmatrix} 2 \\ 1 \\ 2 \end{pmatrix}, \begin{pmatrix} 4 \\ 0 \\ 8 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 2 \\ -1 \\ 6 \end{pmatrix}, \begin{pmatrix} a \\ 1 \\ 1 \end{pmatrix}$$

が生成する部分空間を V とする。 V の次元 $\dim V$ を求めよ。

- (2) s, t , および b を実数とする。実2次正方行列

$$A = \begin{pmatrix} 5 & 2 \\ -2 & 1 \end{pmatrix}$$

を考える。実2次正則行列 P が等式

$$P^{-1}AP = \begin{pmatrix} s & 0 \\ b & t \end{pmatrix}$$

を満たすとする。

- (a) 数ベクトル空間 \mathbb{R}^2 の一次変換 $T_A \left(\begin{pmatrix} x \\ y \end{pmatrix} \right) = A \begin{pmatrix} x \\ y \end{pmatrix}$ の固有値をすべて求めよ。
- (b) s と t の値を求めよ。
- (c) P, b を一組求めよ。

- 2** (1) 広義積分 $\int_0^{\infty} \frac{dx}{x^2 + x + 1}$ を求めよ。

- (2) xy 平面上の関数 $f(x, y) = x^4 + 2xy + y^2$ の極値をすべて求めよ。

- 3 X と Y は独立な確率変数であり, X の密度関数 $f_X(x)$ および Y の密度関数 $f_Y(y)$ はそれぞれ

$$f_X(x) = \begin{cases} e^{-x} & (x \geq 0) \\ 0 & (x < 0) \end{cases}, \quad f_Y(y) = \begin{cases} \frac{1}{2} & (-1 \leq y \leq 1) \\ 0 & (y < -1 \text{ または } y > 1) \end{cases}$$

である。

- (1) X の期待値 $E(X)$ および分散 $V(X)$ を求めよ。
- (2) Y の期待値 $E(Y)$ および分散 $V(Y)$ を求めよ。
- (3) 2つの確率変数 $X+Y$ と $X-Y$ の共分散 $\text{Cov}(X+Y, X-Y)$ を求めよ。また, $X+Y$ と $X-Y$ が独立であるか独立でないかを判定せよ。
- (4) 実数 a が $-1 \leq a \leq 1$ を満たすとき, 確率 $P(X+Y \leq a)$ を求めよ。

(以上)

令和7年4月入学 一般入試（第I期）

April Admission for the 2025 Academic Year for Japanese Students (1st period)

令和6年秋入学 外国人留学生特別入試 Fall Admission for the 2024 Academic Year for International Students

京都工芸繊維大学 大学院工芸科学研究科 博士前期課程（修士課程） 情報工学専攻 試験問題

Question booklet of Entrance Examination for Master's Program of Information Science,

Graduate School of Science and Technology, Kyoto Institute of Technology (KIT)

専門科目 Special Subject

[注意事項 Cautions]

- この問題冊子は合図があるまで中を開かないでください。この中身は以下の2題であり、2題とも必須です。落丁・乱丁および印刷の不鮮明な箇所などがあれば、手を挙げて監督者に知らせなさい。
Do not open this question booklet until permitted by the proctor. Answer all two subject parts listed below. Raise your hand and inform the proctors of any missing pages, disarranged pages, unclear printing, etc.
プログラミング（1） Programming (1)
プログラミング（2） Programming (2)
- 配布物は、この問題冊子1部、解答用紙2枚、および下書き用紙1枚です。
The proctors distribute this question booklet, two answer sheets, and a memo sheet.
- 机の上には受験票以外に、次のものを置いてもよろしい。
You can put the following goods in addition to your exam admission ticket.
(1) 黒鉛筆とシャープペンシル、鉛筆キャップ Black pencils and mechanical pencils, pencil covers
(2) プラスチック製の消しゴム Plastic erasers
(3) 電動でない小型の鉛筆削り（カッターナイフは可） Small-sized non-electric pencil sharpeners (including a small cutter)
(4) 秒針音がしない小型の時計（辞書、電卓、通信等の機能があるものは不可）
Small-sized silent watches or clocks without any additional dictionary, calculator, communication, etc.
(5) 眼鏡、ハンカチ、目薬、無地のマスク、ティッシュペーパー（中身だけを取り出したもの）
Glasses, handkerchiefs, eye drops, plain masks, tissues without package
これら以外については監督者の了解を受けてください。
Ask the proctors for permission to use any goods other than the above.
- プログラミング（1）とプログラミング（2）を別の解答用紙に解答してください。解答用紙2枚すべての上欄指定枠内に、問題科目名、志望専攻名、受験番号を忘れずに記入してください。解答用紙の裏面に解答を書いても構いません。解答用紙と下書き用紙の追加配布はしません。
Use a separate answer sheet for each subject part. Fill in the subject-part name, the major of Master's Program, and your examinee's number in the designated boxes on all two answer sheets. You can use both sides of the answer sheet. No additional sheet is available.
- この問題冊子はバラしても構いません。 You can unbind this booklet.
- 試験終了後も退出の許可があるまで退室はできません。中途退室できません。
Do not leave the room after the exam until permitted by the proctor. Also, you do not during the exam.
- 問題冊子と下書き用紙は持ち帰ってください。
Bring this question booklet and the memo sheet when you leave the room after the exam.

プログラミング(1) [1/2]

問1 C言語で記述された下記 Program1 を実行したとき、標準出力(standard output)に出力される内容を示せ。

Program1

```
#include <stdio.h>

typedef struct {
    char c;
    int i;
} data, *data_p;

void f(data d, data_p pd) {
    pd->c += d.i++;
    pd->i += ++d.i;
}

int main(void) {
    data d[] = { {'a', 1}, {'b', 2}, {'c', 3} };
    f(*d, d + 1);
    f(*d, d + 2);
    printf("%c, %c, %c\n", d[0].c, d[1].c, d[2].c);
    printf("%d, %d, %d\n", d[0].i, d[1].i, d[2].i);
    return 0;
}
```

問2 C言語で記述された下記 Program2 は、文字列(string) s に含まれる英字(alphabetic character)の大文字(upper case)を小文字(lower case)に変換(convert)し、小文字を大文字に変換する関数(function) case_change である。下記の空欄 (あ) ~ (う) を埋めて、関数 case_change を完成させよ。

Program2

```
void case_change(char *s) {
    while( (あ) ) {
        if(*s >= 'A' && *s <= 'Z')
            *s += (い) - (う);
        else if(*s >= 'a' && *s <= 'z')
            *s += (う) - (い);
        s++;
    }
}
```

[次ページに続く]

プログラミング(1) [2/2]

問3 C言語で記述された下記 Program3 に関する以下の間に答えよ。

Program3

```
int a(int n) {
    if(n < 3) return (え);
    return (お);
}

int b(int n) {
    int i, x = 1;
    for(i=2; i<=n; i++) (か);
    return x;
}
```

- (a) 関数 a は、引数(argument) n に対して、以下に示す a_n の値を返す関数である。

$$a_1 = 1, a_2 = 2, a_n = 4(a_{n-1} - a_{n-2}) \quad (n > 2)$$

空欄 (え) ~ (お) を埋めて、関数 a を完成させよ。ただし、引数 n の値は、1 以上 30 以下とする。

- (b) 関数 a と関数 b の引数 n の値が等しいとき、戻り値も等しくなるよう、空欄 (か) を埋めて、関数 b を完成させよ。ただし、関数 b の内部で関数 a を呼び出してはならない。

問4 Java 言語で記述された下記 Program4 に関する以下の間に答えよ。なお、String クラス(class)の charAt メソッド(method)は、引数を n とすると、(n+1) 番目の文字(character)を char 型で返すメソッドである。文字列を構成する文字は、先頭から順に、1 番目、2 番目…、と数える。

Program4

```
class Main {
    private static String processString(String s) {
        return processString(s, s.length() - 1);
    }

    private static String processString(String s, int i) {
        if(i < 0) return "";
        return s.charAt(i) + processString(s, i - 1);
    }

    public static void main(String[] args) {
        System.out.println(processString("program"));
    }
}
```

- (a) 下線部の static が必要である理由を簡潔に説明せよ。なお、下線は説明のために付したものであり、実際のプログラムには含まれない。
- (b) processString メソッドのように、同じ名前のメソッドを複数定義し、引数の数やデータ型に応じて使い分ける仕組みを何と呼ぶか答えよ。また、この仕組みを利用するメリットを簡潔に説明せよ。
- (c) Program4 を実行したとき、標準出力に出力される内容を示せ。

プログラミング (2) [1/2]

問1 Javaで記述されている Program 1 と Program 2 に関する以下の問に答えよ。

(a) Program 1 は、クイックソートアルゴリズム (quick sort algorithm) を用いたデータの整列プログラムである。空欄 (あ) ~ (お) をすべて埋めて Program 1 を完成せよ。

(b) Program 1 で実現しているクイックソートアルゴリズムの時間計算量 (time complexity) について考える。それが最悪時間計算量 (worst time complexity) となる場合のデータを例示しているのは Program 2 である。空欄 (か) と (き) を埋めて Program 2 を完成せよ。但し、worstCaseArray の要素数は 6 とする。

Program 1

```
public class QuickSort {
    public static void quickSort(int[] array, int low, int high) {
        if (low < high) {
            int pivotIndex = (あ); // ピボットの位置を取得します
            // ピボット(pivot)の左側の部分配列を再帰的(recursively)に処理
            (い);
            // ピボット(pivot)の右側の部分配列を再帰的(recursively)に処理
            (う);
        }
    }
    private static int partition(int[] array, int low, int high) {
        int pivot = array[high];
        int i = low - 1;
        for (int j = low; j < high; j++) {
            if (array[j] < pivot) {
                i++;
                (え);
            }
        }
        (お); // ピボット(pivot)を移動
        return i + 1;
    }
    private static void swap(int[] array, int i, int j) {
        int temp = array[i];
        array[i] = array[j];
        array[j] = temp;
    }
    public static void main(String[] args) {
        int[] array = {34, 7, 23, 32, 5, 62};
        quickSort(array, 0, array.length - 1);
        for (int num : array) {
            System.out.print(num + " ");
        }
    }
}
```

[次ページに続く]

プログラミング (2) [2/2]

Program 2

```
public class QuickSortWorstCase {
    public static void main(String[] args) {
        int[] worstCaseArray = (か) ; // 最悪の場合のデータ例
        QuickSort.(き);
        for (int num : worstCaseArray) {
            System.out.print(num + " ");
        }
    }
}
```

問 2 クイックソートアルゴリズム (quick sort algorithm) の時間計算量 (time complexity) に関する以下の問に答えよ。

- (a) クイックソートアルゴリズムの最良時間計算量 (best time complexity)、平均時間計算量 (average time complexity)、および最悪時間計算量 (worst time complexity) を答えよ。
- (b) クイックソートアルゴリズムの最良時間計算量、平均時間計算量、および最悪時間計算量は、それぞれどのような場合で発生するかについて説明せよ。
- (c) クイックソートアルゴリズムの最悪時間計算量の軽減方法について説明せよ。