

アルゴリズムとデータ構造 2024 第1回 演習課題 「プログラミングI・IIの復習」

- 10個の要素を持つ配列 `data` の各要素に、0以上9以下の整数が1つつランダムな順番で格納してある。
 - (a) このとき、`data[0]`が6である確率 (`data[0]`に6が入る確率) を述べよ。
 - (b) `data[0]`の値を“得点”とする場合、得点の「期待値」(無限回試行したときの平均値) の計算式を示せ。
 - (c) (b)の式を計算し、期待値を求めよ。
 - (d) `data` からランダムに1つの要素を選んでその値を得点とする場合、得点の「期待値」を求めよ。

2. `int` 型の配列 `a` を引数に取り (配列の長さは任意), その中身を逆の順番に入れ替える `Java` のメソッドを作成せよ。さらに、長さも要素も異なるいくつかの配列を与える `main` メソッドを追加して、プログラムの動作をテストせよ。

```
public static void reverse(int[] a) {
    int n = a.length;
    for (int i = 0; i < n / 2; i++) {

    }
}
```

3. `int` 型の配列 `a` と `b` を引数に取り (配列の長さは任意), 両者の長さが等しく内容が完全に等しければ真, そうでなければ偽を返す `Java` のメソッドを作成せよ。さらに、長さや要素を変えた何組かの配列を与える `main` メソッドを追加して、それが正しい真偽値を返すかテストせよ。

```
public static boolean equals(int[] a, int[] b) {

}
```

4. 下記の `Java` プログラムは、何時間何分かを表すデータをキーボードから2つ読み込み、合計時間を計算して別の変数に正しい形式で格納してから、それを確認のために画面に表示するものである。空欄を補って完成させよ。
(例: 5時間32分+2時間43分=8時間15分)

```
import java.util.Scanner;

class Jikan {

    public String toString() {
        return hours + "時間" + mins + "分";
    }
}

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        Jikan j1 = new
        Jikan j2 = new
        Jikan j3 = new

        System.out.print("時間-> ");
        j1.hours = sc.
        System.out.print("分 -> ");
        j1.mins = sc.

        System.out.print("時間-> ");
        j2.hours = sc.
        System.out.print("分 -> ");
        j2.mins = sc.

        System.out.println(
            j1 + "+" + j2 + "=" + j3);
    }
}
```

5. 下記のプログラムの実行結果を予想し、確認せよ。さらに、「System.out.print(n);」という処理を移動させるだけで、「12345」と表示されるように修正せよ。

```
public class Main {
    public static void method(int n) {
        if (n == 0) return;
        System.out.print(n);
        method(n - 1);
    }
    public static void main(String [] args) {
        method(5);
    }
}
```

6. 下記のプログラムに適切なコードを補充して、double 型の配列 data を引数に取り (data の要素数は 1 個以上)、data の中で最大値を持つ要素の添字 (インデックス) を返すメソッドを作成し、動作をテストせよ。

```
public static int searchMax(double[] data) {

    for (int i = 1; i < data.length; i++) {
        if (data[i] > data[imax]) {

        }
    }
    return imax;
}
```

7. double 型の配列を引数に取り、その内容をランダムにシャッフルする処理を実現したい。代表的な方法は、配列の中からランダムに 2 つの要素を選び、それらの値を交換する処理を何回も繰り返すものである。下記のコードの空欄を補ってメソッドを完成させ、適切な配列を与える main メソッドも追加して、プログラムの動作をテストせよ。

```
import java.util.Random;

public class Main {
    static Random rnd = new Random();

    public static void shuffle(double[] data, int ntimes) {
        for (int count = 0; count < ntimes; count++) {
            int i = rnd.nextInt(data.length);
            int j = rnd.nextInt(data.length);

        }
    }
}
```

8. 1 以上 999 以下の int 型の整数 n をキーボードから読み込み、n を漢数字で表した文字列を画面に表示するプログラムを作成せよ。n に対する計算によって 1 桁ずつ取り出して、下記の配列を利用して漢字に変換するとよい。
(例: 「987」 → 「九百八十七」, 「500」 → 「五百」, 「103」 → 「百三」)

```
char [] suji = { '零', '一', '二', '三', '四', '五', '六', '七', '八', '九' };
```

9. 下記に示したのは、ビット演算 (ビットごとの論理積) を利用して、引数の整数を 2 で割った余りを返すメソッドである。これを参考にして、同様にビットごとの論理積を用いて引数の整数を 4 で割った余りを返すメソッド (名前は mod4) を作成せよ。さらに、引数として 0 から 9 までの整数を渡して動作をテストせよ。整数がコンピュータの内部の 2 進数でどう表されているかをよく考えて作成するとよい。

```
public static int mod2(int n) {
    return n & 1;
}
```