CCM0118 – Computação I

Curso de Ciências Moleculares

Segundo Semestre de 2024

Prova de Recuperação – 4/2/2025

Nome completo: _____

NUSP: _____

Assinatura: _____

Instruções:

1. Não destaque as folhas deste caderno.
2. Preencha o cabeçalho acima.
3. A prova pode ser feita a lápis. Cuidado com a legibilidade.
4. A prova consta de **3 questões**. Verifique antes de começar a prova se o seu caderno de questões está completo. **A prova vale 12 pontos**.
5. Não é permitido o uso de folhas avulsas para rascunho.
6. Não é permitido o uso de artefatos eletrônicos.
7. Não é permitida a consulta a livros, apontamentos ou colegas.
8. Não é necessário apagar rascunhos no caderno de questões.

DURAÇÃO DA PROVA: 2 horas

Questão	Nota
1	
2	
3	
Total	

Q1 (5.0 pontos) Considere inicialmente o seguinte programa. Lembre que, dada uma matriz m, a expressão $m[i]$ refere-se a sua i-ésima linha.

```
public class Q1Aux
{
    public static int[] l(int m) {
        int[] a = new int[m];
        for (int i = 0; i < m; i++)
            a[i] = i + 1;
        StdOut.print("l(" + m + "): ");
        show(a);
        return a;
    }

    public static void show(int[] a) {
        int N = a.length;
        for (int i = 0; i < N; i++)
            StdOut.print(a[i] + " ");
        StdOut.println();
    }

    public static void show(int[][] m) {
        int N = m.length;    // number of rows in the matrix m
        for (int i = 2; i < N; i++)
            show(m[i]);
    }

    public static void main(String[] args)
    {
        int[][] m = new int[4][];   // m has 4 rows, each of unspecified length
        m[2] = l(2);   // row 2 of m is an array returned by a call of l()
        m[3] = l(3);   // same for row 3
        StdOut.println("\nMatrix m (rows 2 and 3):");
        show(m);
    }
}
```

Quando executado, Q1Aux.java tem a seguinte saída:

```
$ java-introcs Q1Aux
l(2): 1 2
l(3): 1 2 3

Matrix m (rows 2 and 3):
1 2
1 2 3
$
```

O programa acima ilustra o uso de matrizes com linhas de tamanhos variados: a matriz m construída no programa tem as linhas 2 e 3 com 2 e 3 elementos, respectivamente.

Considere agora o seguinte programa.

```
public class Q1
{
    public static int[] digit(int N) {
        int[] r = new int[10];
        int i = 0;
        while (N > 0) {
            r[i] = N % 10;
            N /= 10;
            i++;
        }
        show(r);
        int [] d = new int[i];
        for (int j = 0; j < i; j++)
            d[j] = r[i - j - 1];
        show(d);
        return d;
    }

    public static int[][] modCount(int[] a) {
        int[][] m = new int[11][];
        for (int b = 2; b <= 10; b++)
            m[b] = modCount(a, b); // b-ésima linha de m
        return m;
    }

    public static int[] modCount(int[] a, int b) {
        int N = a.length;
        int[] l = new int[b];
        for (int i = 0; i < N; i++)
            l[a[i] % b]++;
        return l;
    }

    public static void show(int[] a) {
        int N = a.length;
        for (int i = 0; i < N; i++)
            StdOut.print(a[i] + " ");
        StdOut.println();
    }

    public static void show(int[][] m) {
        int N = m.length;
        for (int i = 2; i < N; i++)
            show(m[i]);
    }
```

```
    public static void main(String[] args)
    {
        int NUSP = Integer.parseInt(args[0]);
        int[] digit = digit(NUSP);
        int[][] modCount = modCount(digit);
        show(modCount);
    }
}
```

Diga qual será a saída do programa acima quando seu número USP é fornecido como argumento de linha de comando. Por exemplo, se seu NUSP fosse 12345678, você teria de dizer qual é a saída da execução

```
$ java-introcs Q1 12345678
```

Observação. Seu NUSP não é 12345678. Use seu NUSP.

Saída

Rascunho

Q2 (4.0 pontos) Considere o seguinte programa.

```
public class Q2Ex
{
    public static void l1(int N) {
        int M = 2 * N;
        for (int y = -M; y <= M; y++) {
            for (int x = -M; x <= M; x++)
                if (Math.abs(x) + Math.abs(y) <= N)
                    StdOut.print("* ");
                else
                    StdOut.print(". ");
            StdOut.println();
        }
    }

    public static void main(String[] args)
    {
        int N = Integer.parseInt(args[0]);
        l1(N);
    }
}
```

Segue um exemplo de execução de Q2Ex.java:

```
$ java-introcs Q2Ex 2
. . . . . .
. . . . . .
. . . * . . .
. . * * * . .
. * * * * * .
. . * * * . .
. . . * . . .
. . . . . .
. . . . . .
$
```

Ao executá-lo com argumentos 0, 1 e 3, as saídas de Q2Ex.java são

```
*
```

```
. . . . .
. . * . .
. * * * .
. . * . .
. . . . .
```

```
. . . . . .
. . . . . .
. . . * . . .
. . * * * . .
. * * * * * .
. . * * * . .
. . . * . . .
. . . . . .
. . . . . .
```

```
. . . . . . . . . . . . . . .
. . . . . . . . . . . . . . .
. . . . . . . . . . . . . . .
. . . . . * . . . . . .
. . . . . * * * . . . . .
. . . . * * * * * . . . .
. . . * * * * * * * . . .
. . . . * * * * * . . . .
. . . . . * * * . . . . .
. . . . . * . . . . . .
. . . . . . . . . . . . . . .
. . . . . . . . . . . . . . .
. . . . . . . . . . . . . . .
```

Considere agora o seguinte programa incompleto:

```
public class Q2
{
    public static void linf(int N) {

        // a completar

    }

    public static void main(String[] args)
    {
        int N = Integer.parseInt(args[0]);
        linf(N);
    }
}
```

Implemente a função `linf()` do programa acima de forma que, ao executarmos Q2.java com parâmetros 0, 1, 2 e 3, obtemos as seguintes saídas:

```
*
```

```
. . . . .
. * * * .
. * * * .
. * * * .
. . . . .
```

```
. . . . . . . . . .
. . . . . . . . . .
. . * * * * * * . .
. . * * * * * * . .
. . * * * * * * . .
. . * * * * * * . .
. . . . . . . . . .
. . . . . . . . . .
```

```
. . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . .
. . . * * * * * * * * . . .
. . . * * * * * * * * . . .
. . . * * * * * * * * . . .
. . . * * * * * * * * . . .
. . . * * * * * * * * . . .
. . . * * * * * * * * . . .
. . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . .
```

Escreva a seguir a função `linf()` que falta por completo (incluindo seu cabeçalho).

(Rascunho)

Q3 (3.0 pontos) Sejam s e t strings. Um *embaralhamento* (*shuffle*) de s e t é um string que podemos obter intercalando-se os caracteres de s e t, de forma arbitrária, mas respeitando-se a ordem dos caracteres em s e em t. Por exemplo, se s = AB e t = abc, há 10 embaralhamentos possíveis: ABabc, AaBbc, AabBc, AabcB, aABbc, aAbBc, aAbcB, abABc, abAcB, abcAB. Note que, em todos os embaralhamentos, as letras em s e em t ocorrem na ordem original, de forma s e t são subsequências de todos os embaralhamentos. Note também que se s tem m caracteres e t tem n caracteres, então todo embaralhamento de s e t tem $m + n$ caracteres.

Considere agora o seguinte programa, que imprime todos os embaralhamentos de dois strings dados como argumentos de linha de comando.

```
public class Q2
{
    public static void shuffle(String s, String t) {
        shuffle("", s, t);
    }

    public static void shuffle(String prefix, String s, String t) {

       // a completar

    }

    public static void main(String[] args)
    {
        String s = args[0];
        String t = args[1];
        shuffle(s, t);
    }
}
```

Nesta questão, você deve completar a implementação da função `shuffle()` do programa acima. Seguem alguns exemplos de execução do programa Q2.java:

```
$ java-introcs Q2 A a
Aa
aA
$ java-introcs Q2 A bc
Abc
bAc
bcA
$ java-introcs Q2 AB bc
ABbc
AbBc
AbcB
bABc
bAcB
bcAB
$
```

Sugestão. A função

```
public static void shuffle(String prefix, String s, String t)
```

que você implementará deve imprimir `prefix` + `t` se s é o string vazio. Analogamente, ela deve imprimir `prefix` + `s` se t é o string vazio. Suponha agora que ambos s e t sejam não-vazios. Seja s_0 o primeiro caractere de s e seja t_0 o primeiro caractere de t. A função shuffle deve imprimir todos os embaralhamentos de s e t que começam com s_0 seguido de todos os embaralhamentos de s e t que começam com t_0.

Observação. Nesta questão, você pode supor que s e t não têm caracteres comuns (isto é, nenhum caractere de s ocorre em t e vice-versa).

Escreva a seguir a função `shuffle()` que falta por completo (incluindo seu cabeçalho).

* * *

(Rascunho)

(Rascunho)