

CCM0118 – Computação I

CURSO DE CIÊNCIAS MOLECULARES

SEGUNDO SEMESTRE DE 2024

Prova 1 – 3/10/2024

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 (4.0 pontos) Diga qual será a saída do programa abaixo quando executado com seu número USP como argumento de linha de comando.

Importante: seu rascunho deve indicar como você chegou a sua resposta.

```
public class Q1
{
    public static void main(String[] args)
    {
        int NUSP = Integer.parseInt(args[0]);
        int DMAX = 10;
        int[] digits = new int[DMAX];

        int d = 0;
        while (NUSP > 0) {
            digits[d] = NUSP % 10;
            NUSP /= 10;
            d++;
        }

        for (int i = 0; i < d; i++)
            StdOut.print(digits[i] + " ");
        StdOut.println();

        int BASE = 10;
        int[] count = new int[BASE + 1];
        for (int i = 0; i < d; i++)
            count[digits[i] + 1]++;

        for (int i = 0; i < BASE; i++)
            StdOut.print(count[i + 1] + " ");
        StdOut.println();

        for (int r = 0; r < BASE; r++)
            count[r + 1] += count[r];

        for (int i = 0; i < BASE; i++)
            StdOut.print(count[i + 1] + " ");
        StdOut.println();

        int[] aux = new int[DMAX];
        for (int i = 0; i < d; i++) {
            aux[count[digits[i]]] = digits[i];
            count[digits[i]]++;
        }

        for (int i = 0; i < d; i++)
            StdOut.print(aux[i] + " ");
        StdOut.println();
    }
}
```

Rascunho

--

Saída do programa

--

Q2 (4.0 pontos) Suponha que o vetor a contenha uma permutação dos inteiros $0, 1, \dots, N - 1$ ($N \geq 1$). Fixe $0 \leq i < N$. Considere a lista de valores $i, a[i], a[a[i]], \dots, a^{(k)}[i], \dots$, onde $a^{(k)}[i]$ denota o valor $a[a[\dots[i]\dots]]$ com k ocorrências de $a[\dots]$ (por exemplo, $a^{(2)}[i] = a[a[i]]$). Como estamos supondo que a contém uma permutação de $0, 1, \dots, N - 1$, para algum $k > 0$, vale que $a^{(k)}[i] = i$. O menor $k > 0$ tal que $a^{(k)}[i] = i$ é o *período* de i em a .

Considere o seguinte programa incompleto:

```
public class Q2
{
    public static void main(String[] args)
    {
        int N = Integer.parseInt(args[0]);
        int[] a = new int[N];

        for (int i = 0; i < N; i++)
            a[i] = i;

        for (int i = 0; i < N; i++) {
            int r = i + (int) (Math.random() * (N - i));
            int t = a[r];
            a[r] = a[i];
            a[i] = t;
        }

        for (int i = 0; i < N; i++)
            StdOut.print(i + " ");
        StdOut.println();
        for (int i = 0; i < N; i++)
            StdOut.print(a[i] + " ");
        StdOut.println();

        // a completar

    }
}
```

Nesta questão, você deve escrever trechos de código a serem inseridos no ponto indicado por “a completar” no programa acima para resolver os seguintes itens.

- (a) Neste item, você deve escrever um trecho de código para que o programa Q2.java determine e imprima o período de 0 na permutação a . Seu programa deve comportar-se como ilustram os seguintes exemplos:

```
$ java-introcs Q2 10
0 1 2 3 4 5 6 7 8 9
7 2 5 0 9 3 4 1 8 6
period of 0: 6
$ java-introcs Q2 10
```

- ```
$ java-introcs Q2 10
0 1 2 3 4 5 6 7 8 9
7 2 5 0 9 3 4 1 8 6
Periods of all elements:
0 1 2 3 4 5 6 7 8 9
```

\$

[illegible]

$$m(t) \quad M(T)$$

Nesta questão, você pode supor que a sequência que será dada a seu programa terá no máximo 10000 elementos.

```
$ cat seq0.txt
3 1 4 1 5 9 2 6 5 3 5 8 9 7 9 3 2 3 8 5
$ java-introcs Q3 < seq0.txt
1 (2) 9 (3)
$ cat seq1.txt
8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
$ java-introcs Q3 < seq1.txt
8 (15) 8 (15)
$
```





**Dica.** Talvez seja de ajuda considerar o seguinte programa.

```
public class Help
{
 public static void main(String[] args)
 {
 int NMAX = 10000;
 int[] a = new int[NMAX];

 int N = 0;
 while (!StdIn.isEmpty()) {
 a[N] = StdIn.readInt();
 N++;
 }

 for (int i = 0; i < N; i++)
 StdOut.print(a[i] + " ");
 StdOut.println();
 }
}
```

Exemplo de execução:

```
$ cat seq0.txt
3 1 4 1 5 9 2 6 5 3 5 8 9 7 9 3 2 3 8 5
$ java-introcs Help < seq0.txt
3 1 4 1 5 9 2 6 5 3 5 8 9 7 9 3 2 3 8 5
$
```

\* \* \*

(Rascunho)