

MAC 2166 – Introdução à Computação
ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO
PRIMEIRO SEMESTRE DE 2022

Terceira Prova – 12 de julho de 2022

Nome do aluno: _____

NUSP: _____ Turma: _____

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 **5 questões**. Verifique antes de começar a prova se o seu caderno de questões está completo.
5. Não é permitido o uso de folhas avulsas para rascunho.
6. Não é permitido o uso de artefatos eletrônicos.
7. Não é permitido 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	
4	
5	
Total	

Q1. (3.0 pontos) Simule a execução do programa abaixo, usando seu número USP como entrada. Basta indicar a saída do programa, no espaço indicado.

```
#include <stdio.h>

#define MAX 15

void t(int v[MAX], int i, int j);
int f(int v[MAX]);
void g(int v[MAX], int N);
void p(int v[MAX], int N);

int main()
{
    int N, v[MAX];

    N = f(v);
    p(v, N);
    g(v, N);
    p(v, N);

    return 0;
}

void t(int v[], int i, int j) {
    int x = v[i];
    v[i] = v[j];
    v[j] = x;
}

int f(int v[]) {
    int i = 0, N = 0, NUSP;
    scanf("%d", &NUSP);
    while (NUSP > 0) {
        v[i] = NUSP % 10;
        NUSP = NUSP / 10;
        N++; i++;
    }
    return N;
}

void g(int v[], int N) {
    int i;
    while (N > 0) {
        for (i = 0; i < N - 1; ++i)
            t(v, i, i + 1);
        p(v, N);
        N--;
    }
}

void p(int v[], int N) {
    int i;
    for (i = 0; i < N; ++i)
        printf("%d ", v[i]);
    printf("\n");
}
```

Saída:

Q2. (2.0 pontos) Faça uma função de protótipo

```
int crescente(int v[], int N);
```

que devolve 1 (ou **TRUE**) se

$$v[0] \leq v[1] \leq \dots \leq v[N-1]$$

e devolve 0 (ou **FALSE**) caso contrário. Chamamos de *crescentes* os vetores v que satisfazem a condição acima.

Q3. (2.0 pontos) Dizemos que uma matriz é *linha-crescente* se suas linhas são todas crescentes (no sentido da Questão **Q2.**). Faça uma função de protótipo

```
int l_crescente(int A[][NMAX], int M, int N);
```

que recebe uma matriz A com M linhas e N colunas e que devolve 1 (ou **TRUE**) se A é linha-crescente e devolve 0 (ou **FALSE**) caso contrário. Você deve usar a função da Questão **Q2.**, mesmo que você não a tenha escrito.

Q4. (2.0 pontos) Escreva um programa que lê inteiros M e N com $0 < M, N < 1000$ e uma matriz de inteiros A com M linhas e N colunas e que imprime

Matriz 'e linha-crescente

se A for linha-crescente e que imprime

Matriz nao 'e linha-crescente

caso contrário. Por exemplo, para entrada

3 4

```
0  1  2  3
-3 -2  0  1
-1 -1 -1 -1
```

seu programa deve imprimir que a matriz é linha-crescente. Para entrada

3 4

```
0  1  2  3
-3 -2  0 -1
-1 -1 -1 -1
```

seu programa deve imprimir que a matriz não é linha crescente. Seu programa deve usar a função da questão anterior, mesmo que você não a tenha escrito. Não é necessário reescrever aquela função aqui.

Q5. (2.0 pontos) Escreva uma função de protótipo

```
void conte(char s[], int *m, int *M);
```

que recebe em `s` uma string e que devolve em `*m` e em `*M`, respectivamente, o número de letras minúsculas em `s` e o número de letras maiúsculas em `s`. Por exemplo, se `s` contém a string

Foguete da NASA

então depois da execução da chamada

```
conte(s, &mi, &ma);
```

a variável `mi` deve valer 8 e a variável `ma` deve valer 5.

