

MAC 2166 – Introdução à Computação

ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO

PRIMEIRO SEMESTRE DE 2023

Prova Substitutiva – 11 de julho de 2023

Nome completo: _____

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 3 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	
Total	

1. **(3.0 pontos)** Simule a execução do programa abaixo, destacando o que vai ser impresso. Você deve dar seu número USP como entrada.

```
#include <stdio.h>

int main()
{
    int NUSP, D = 0, i;
    int freq[10], dig[10];

    scanf("%d", &NUSP);

    i = 0;
    while (NUSP > 0) {
        dig[i] = NUSP % 10;
        i++;
        NUSP /= 10;
        D++;
    }

    printf("D = %d\n", D);

    for (i = 0; i < D; i++)
        printf("%d ", dig[D - 1 - i]);
    printf("\n");

    for (i = 0; i < D; i++)
        printf("%d ", dig[i]);
    printf("\n");

    for (i = 0; i < 10; i++)
        freq[i] = 0;

    for (i = 0; i < D; i++)
        freq[dig[i]]++;

    for (i = 0; i < 10; i++)
        while (freq[i] > 0) {
            printf("%d ", i);
            freq[i]--;
        }

    printf("\n");

    return 0;
}
```

Rascunho

Saída do programa

2. **(3.0 pontos)** Para todo x real com $|x| < 1$, vale que

$$\arcsin x = \sum_{k=0}^{\infty} \frac{(2k)!}{4^k (k!)^2 (2k+1)} x^{2k+1} = x + \frac{1}{6}x^3 + \frac{3}{40}x^5 + \dots \quad (1)$$

Se tomamos

$$a_k = \frac{(2k)!}{4^k (k!)^2 (2k+1)} \quad (2)$$

para todo $k \geq 0$, a série em (1) torna-se

$$\arcsin x = \sum_{k=0}^{\infty} a_k x^{2k+1}. \quad (3)$$

Ademais, para todo $k \geq 1$, temos

$$a_k = \frac{(2k-1)^2}{2k(2k+1)} a_{k-1}. \quad (4)$$

Escreva um programa que lê um valor x com $|x| < 1$ dado pelo usuário e um natural N , e que determina uma aproximação para $\arcsin x$ somando as parcelas em (1) com $0 \leq k \leq N$ (isto é, as primeiras $N + 1$ parcelas da série em (1)). Seu programa deve, obrigatoriamente, usar a relação (4) para determinar as parcelas a serem somadas.

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

3. **(4.0 pontos)** Seja P uma matriz M por M de zeros e uns e sejam A e B matrizes inteiras N por N . A **composta** $P(A, B)$ é definida como sendo a matriz que obtemos ao substituir os zeros de P por A e os uns de P por B . Por exemplo, sejam P , A e B como abaixo:

$$P = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad A = \begin{bmatrix} 2 & 4 & 2 & 6 \\ 8 & 2 & 6 & 4 \\ 8 & 0 & 2 & 2 \\ 4 & 4 & 0 & 4 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 5 & 3 & 7 \\ 5 & 5 & 7 & 9 \\ 3 & 1 & 5 & 1 \\ 3 & 1 & 5 & 9 \end{bmatrix} \quad (5)$$

Note que, no exemplo acima, temos $M = 3$ e $N = 4$. Ademais,

$$P(A, B) = \begin{bmatrix} \boxed{\begin{matrix} 1 & 5 & 3 & 7 \\ 5 & 5 & 7 & 9 \\ 3 & 1 & 5 & 1 \\ 3 & 1 & 5 & 9 \end{matrix}} & \begin{matrix} 2 & 4 & 2 & 6 \\ 8 & 2 & 6 & 4 \\ 8 & 0 & 2 & 2 \\ 4 & 4 & 0 & 4 \end{matrix} & \begin{matrix} 1 & 5 & 3 & 7 \\ 5 & 5 & 7 & 9 \\ 3 & 1 & 5 & 1 \\ 3 & 1 & 5 & 9 \end{matrix} \\ \begin{matrix} 1 & 5 & 3 & 7 \\ 5 & 5 & 7 & 9 \\ 3 & 1 & 5 & 1 \\ 3 & 1 & 5 & 9 \end{matrix} & \boxed{\begin{matrix} 2 & 4 & 2 & 6 \\ 8 & 2 & 6 & 4 \\ 8 & 0 & 2 & 2 \\ 4 & 4 & 0 & 4 \end{matrix}} & \begin{matrix} 2 & 4 & 2 & 6 \\ 8 & 2 & 6 & 4 \\ 8 & 0 & 2 & 2 \\ 4 & 4 & 0 & 4 \end{matrix} \\ \begin{matrix} 2 & 4 & 2 & 6 \\ 8 & 2 & 6 & 4 \\ 8 & 0 & 2 & 2 \\ 4 & 4 & 0 & 4 \end{matrix} & \begin{matrix} 1 & 5 & 3 & 7 \\ 5 & 5 & 7 & 9 \\ 3 & 1 & 5 & 1 \\ 3 & 1 & 5 & 9 \end{matrix} & \begin{matrix} 2 & 4 & 2 & 6 \\ 8 & 2 & 6 & 4 \\ 8 & 0 & 2 & 2 \\ 4 & 4 & 0 & 4 \end{matrix} \end{bmatrix}. \quad (6)$$

Os blocos 4 por 4 marcados em $P(A, B)$ correspondem ao 0 na posição central de P e ao 1 no canto superior esquerdo de P . Note que 0 foi substituído por A e 1 foi substituído por B .

Nesta questão, você deve escrever um programa que, dados P , A e B , calcula e imprime $P(A, B)$. Parte do programa é dado; sua tarefa é escrever as funções que faltam para completar esse programa.

Importante. Você deve supor que as matrizes P , A e B são dadas como no exemplo abaixo:

```
3
1 0 1
1 0 0
0 1 0

4
2 4 2 6
8 2 6 4
8 0 2 2
4 4 0 4

1 5 3 7
5 5 7 9
3 1 5 1
3 1 5 9
```

Isto é, a entrada consiste do valor de M , seguido das entradas de P , seguido do valor de N , seguido das entradas de A , seguido das entradas de B .

Segue o programa a ser completado:

```
#include <stdio.h>

#define MMAX 10
#define NMAX 10

void leia_matrizes(int P[][MMAX], int *M, int A[][NMAX], int B[][MMAX], int *N);
void calcule_composta(int C[][MMAX * NMAX],
                     int P[][MMAX], int M, int A[][NMAX], int B[][NMAX], int N);

void insira(int C[][MMAX * NMAX], int ii, int jj, int X[][NMAX], int N);
void imprima_matriz(int C[][MMAX * NMAX], int T);

int main() {
    int M, N;
    int P[MMAX][MMAX], A[NMAX][NMAX], B[NMAX][NMAX],
        C[MMAX * NMAX][MMAX * NMAX];

    leia_matrizes(P, &M, A, B, &N);
    calcule_composta(C, P, M, A, B, N);

    imprima_matriz(C, M * N);

    return 0;
}

void imprima_matriz(int C[][MMAX * NMAX], int T) {
    int i, j;

    for (i = 0; i < T; i++) {
        for (j = 0; j < T; j++)
            printf("%d ", C[i][j]);
        printf("\n");
    }
}
```

Nas páginas seguintes você deve escrever as três funções que faltam.

(a) Escreva uma função de protótipo

```
void leia_matrizes(int P[][MMAX], int *M,
                  int A[][NMAX], int B[][MMAX], int *N);
```

que lê as matrizes P , A e B , dadas como discutido acima. Esta função deve colocar em *M e *N os valores de M e N lidos.

[illegible]

(b) Escreva uma função de protótipo

```
void insira(int C[][MMAX * NMAX], int ii, int jj, int X[][NMAX], int N);
```

que insere na matriz **C** a matriz **N** por **N** inteira **X**, “a partir da posição (ii, jj)” de **C**. Note que, por exemplo, uma vez escrita esta função, para inserir a matriz *A* do exemplo no começo desta questão na parte central de uma matriz 12 por 12 inteira **C** (como em (6)), basta executar

```
insira(C, 4, 4, A, 4);
```

Para inserir B na parte superior esquerda de $\mathbb{C}$ (como em (6)), basta executar

```
insira(C, 0, 0, B, 4);
```

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

(c) Escreva uma função de protótipo

```
void calcule_composta(int C[][MMAX * NMAX],
                     int P[][MMAX], int M, int A[][NMAX], int B[][NMAX], int N);
```

que recebe matrizes **P**, **A** e **B**, onde **P** é **M** por **M** e **A** e **B** são **N** por **N** e que armazena na matriz **C** a composição **P(A, B)**. Você pode supor que **C** é grande o suficiente para comportar **P(A,B)**. Use, obrigatoriamente, a função **insira()** do item anterior, mesmo que você não a tenha escrito.