

**MAC 110 – Introdução à Computação**  
**BACH. EM CIÊNCIA DA COMPUTAÇÃO – PRIM SEMESTRE DE 2019**  
**Segunda Prova – 29 de junho de 2019**

Nome do aluno: \_\_\_\_\_

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 4 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 a consulta a livros, apontamentos ou colegas.
7. Não é necessário apagar rascunhos no caderno de questões.

**DURAÇÃO DA PROVA: 2 horas**

Questão	Nota
1	
2	
3	
4	
Total	

1. (valor 1.5 pontos)

Simule a execução do programa abaixo, destacando o que vai ser impresso, ou seja, destaque como ficaria a tela do computador quando o programa fosse executado. Use na simulação o seu NUSP como entrada.

```
#include<stdio.h>
#define MAX 5

int f(int v[MAX], int *m, int n)
{
    int i;
    n--;
    *m = *m - n;
    if (n > 0 && n < MAX) v[n] = *m * *m;
    else v[MAX - 1] = *m * *m;
    printf("=====> (f) m = %3d n = %3d ", *m, n);
    for (i = 0; i < MAX; i++) printf("v[%d]=%d ", i,v[i]);
    printf("\n");
    return (n + *m + v[2]);
}

int main()
{
    int n, i, i1, i2, a[MAX];

    printf("Digite seu NUSP:\n");
    scanf("%d", &n);
    for (i = 0; i < MAX; i++){
        a[i] = n % 10;
        n = n / 10;
    }
    i1 = a[2];
    i2 = a[3];
    printf("(1) i1 = %3d i2 = %3d ", i1, i2);
    for (i = 0; i < MAX; i++) printf("a[%1d]=%2d ", i, a[i]);
    printf("\n");
    a[0] = f(a, &i1, i2);
    printf("(2) i1 = %3d i2 = %3d ", i1, i2);
    for (i = 0; i < MAX; i++) printf("a[%1d]=%2d ", i, a[i]);
    printf("\n");
    i2 = f(a, &a[0], a[1]);
    printf("(2) i1 = %3d i2 = %3d ", i1, i2);
    for (i = 0; i < MAX; i++) printf("a[%1d]=%2d ", i, a[i]);
    printf("\n");
    return 0;
}
```





2. (valor 2.0+1.0 pontos)

a. Faça uma função de protótipo

```
void elimina(int v[MAX], int *n, int i, int f, int k)
```

que recebe um vetor com  $n$  elementos e elimina do intervalo do vetor entre os índices  $i$  e  $f$  (incluindo estes dois índices) os múltiplos de  $k$ , mantendo a ordem dos elementos que permanecem no vetor, e atualizando o valor de  $n$ .



b. Faça um programa que implementa a ideia do crivo de Eratóstenes para determinar todos os números primos menores que um valor  $0 < m \leq 1000$  dado. A ideia do crivo é, em cada passo, eliminar os múltiplos de cada primo encontrado. Veja o exemplo abaixo para  $m = 25$ :

O vetor é inicializado com todos os inteiros de 2 a 25:

2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25

Em seguida, eliminamos do vetor os múltiplos de 2 (o primeiro primo):

2 **3** 5 7 9 11 13 15 17 19 21 23 25

Depois os múltiplos de 3 (que ainda não foram eliminados):

2 3 **5** 7 11 13 17 19 23 25

de 5:

2 3 5 7 11 13 17 19 23

Seguimos o processo até o fim do vetor, e encontramos todos os primos até 25.

**Utilize a função do item a. mesmo que você não a tenha feito. Você não precisa reescrever a função neste item.**



3. (valor 2.5 pontos)

Dada uma matriz  $A_{m \times n}$ ,  $0 < m, n < 1000$ , tal que todos os elementos da matriz são zeros ou uns, determine a maior submatriz quadrada dessa matriz composta apenas de 1's.

Exemplo:

Dada a matriz  $A_{15 \times 10}$  abaixo,

$$\begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix}$$

Seu programa deverá imprimir que a maior matriz quadrada composta apenas por 1's tem tamanho  $5 \times 5$ . Você não precisa dizer onde está a submatriz, mas apenas o seu tamanho.



4. (valor 2.0 + 1.0)  
a. Faça uma função de protótipo

```
int diagonal_nula (float A[MAX][MAX], int n, int k)
```

que recebe uma matriz real  $A_{n \times n}$ ,  $n > 0$ , e um inteiro  $0 \leq k \leq n - 1$  e devolve 1 se a matriz  $A$  tem a  $k$ -ésima diagonal abaixo da diagonal principal composta apenas por zeros e 0 caso contrário. Quando  $k = 0$  a função verifica se a diagonal principal é nula.

Exemplos: Para a matriz  $5 \times 5$  abaixo,

$$\begin{pmatrix} 3 & 2 & 1 & 4 & 8 \\ 0 & 1 & 3 & 2 & 1 \\ 3 & 0 & 8 & 0 & 1 \\ 0 & 0 & 0 & 1 & 3 \\ 9 & 0 & 0 & 0 & 4 \end{pmatrix}$$

a chamada da função

```
diagonal_nula (A, 5, 1)
```

deve devolver 1, enquanto que

```
diagonal_nula (A, 5, 4)
```

deve devolver 0.



b. Faça um programa que leia um inteiro  $0 < n < 1000$  e uma matriz real  $A_{n \times n}$  e calcule o maior inteiro  $0 \leq m \leq n$  tal que a matriz tem **banda nula**  $m$ . Dizemos que uma matriz  $A$  tem banda nula  $m$  se as  $m$  primeiras diagonais abaixo da diagonal principal são nulas, contando de baixo para cima.

Exemplo: Para a matriz  $5 \times 5$  abaixo,

$$\begin{pmatrix} 3 & 2 & 1 & 4 & 8 \\ 0 & 1 & 3 & 2 & 1 \\ \mathbf{0} & 2 & 8 & 0 & 1 \\ \mathbf{0} & \mathbf{0} & 0 & 1 & 3 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & 6 & 4 \end{pmatrix}$$

seu programa deve imprimir

A matriz tem banda nula 3.