

MAC 2166 – Introdução à Computação

ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO

PRIMEIRO SEMESTRE DE 2023

Segunda Prova – 4 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. **(2.0 pontos)** Simule a execução do programa abaixo, destacando o que vai ser impresso. Você deve dar seu número USP como entrada.

```
#include <stdio.h>

void print_array(int v[], int n);

void reflect_copy(int a[], int b[], int n);

int main() {
    int nusp, n, u[10], v[10];

    scanf("%d", &nusp);

    u[0] = 0;
    for (n = 1; nusp > 0; n++) {
        u[n] = nusp % 10;
        nusp /= 10;
    }

    printf("n = %d\n", n);
    print_array(u, n);

    reflect_copy(v, u, n);

    print_array(v, n);
    print_array(u, n);

    reflect_copy(u, u, n);

    print_array(u, n);

    return 0;
}

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

void reflect_copy(int a[], int b[], int n) {
    int i;
    for (i = 0; i < n; i++)
        a[i] = b[n-i-1];
}
```

Rascunho

Saída do programa

2. (1.0 + 1.5 + 2.0 pontos) Seja M uma matriz N por N de inteiros. Dizemos que uma entrada M_{ij} de M é um **superzero** se o “bloco” 3 por 3 de M centrado em M_{ij} contém somente entradas nulas. Por exemplo, na matriz M abaixo, a entrada M_{62} é um superzero (indexamos as linhas e colunas da matriz começando com 0):

$$M = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & \boxed{\begin{matrix} 0 & 0 & 0 \\ 0 & \textcircled{0} & 0 \\ 0 & 0 & 0 \end{matrix}} & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & \textcircled{0} & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & \boxed{\begin{matrix} 0 & 0 & 0 \end{matrix}} & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

A matriz M acima tem um total de 4 entradas que são superzeros (M_{54} , M_{55} e M_{56} também são superzeros).

Nesta questão, você deve escrever um programa que lê o valor de N e uma matriz inteira N por N e que determina quantas entradas de M são superzeros. De fato, parte do programa é dado, e sua tarefa é implementar as funções usadas nesse programa para completá-lo. (Este programa supõe que $N \leq \text{NMAX}$.)

```
#include <stdio.h>

#define NMAX 100

void leia_matriz(int M[] [NMAX], int *N);
int n_s_zeros(int M[] [NMAX], int N);
int s_zero(int M[] [NMAX], int N, int i, int j);

int main()
{
    int N;
    int M[NMAX] [NMAX];

    leia_matriz(M, &N);

    printf("Numero de superzeros: %d\n", n_s_zeros(M, N));

    return 0;
}
```


3. **(1.0 + 1.5 + 1.5 pontos)** Um grupo de estudantes faz uma prova. A prova tem 100 testes de múltipla escolha, cada teste valendo um ponto, de forma que a nota de cada estudante é um inteiro que varia de 0 a 100 (não há notas fracionárias). Nesta questão, você deve escrever um programa que lê a lista de notas desses estudantes e imprime a média, a entropia e a maior moda dessa lista de notas (as definições de entropia, média e moda são dadas no item relacionado).

Parte do programa é dado (veja abaixo). Em cada item desta questão, você deve escrever a implementação das funções que ocorrem no programa abaixo para completá-lo. Supomos no programa abaixo que há no máximo `NMAX` estudantes e `RANGE` é a nota máxima possível.

```
#include <stdio.h>
#include <math.h>

#define NMAX 1000000
#define RANGE 100

int read_scores(int v[]);
void get_freq(int freq[], int v[], int N);
double entropy(int freq[], int N);
double average_mode(int freq[], int N, int *mode);

int main() {
    int N, v[NMAX], freq[RANGE + 1], mode;
    double ave, ent;

    N = read_scores(v);
    get_freq(freq, v, N);

    ent = entropy(freq, N);
    ave = average_mode(freq, N, &mode);

    printf("Entropy: %f\n", ent);
    printf("Average: %f\n", ave);
    printf("Mode: %d\n", mode);

    return 0;
}

int read_scores(int v[]) {
    int x, N = 0;
    scanf("%d", &x);
    while (x >= 0) {
        v[N] = x;
        N++;
        scanf("%d", &x);
    }
    return N;
}
```


