

Laboratório 9

Operações com vetores; Vetores multidimensionais.

Introdução à Programação C (CMT012)

Prof. Ronald Souza

IC/UFRJ — 18/10/2023

Objetivo

Praticar os conceitos de programação vistos na Aula 9.

Todos os seus programas devem preencher a estrutura abaixo.

```
/*
Autor: <nome do aluno>
Data: <data de hoje>
Descrição: <o que o programa faz>
Entrada: <o que o programa espera receber como entrada>
Saída: <o que o programa retorna para o usuário>
Defesa: <restrições sobre os dados de entrada, se existirem>
*/

#include <stdio.h>

int main() {
    //dicionário de dados

    //corpo do programa

    return 0;
}
```

Atividade 1: Chama-se **Quadrado Mágico** uma matriz quadrada de números em que a soma de cada coluna, de cada linha e das duas diagonais são iguais. Veja o exemplo para uma matriz 3x3:

```
2 7 6
9 5 1
4 3 8
```

*Repare acima que a soma dos elementos de **cada linha**, de **cada coluna** e das **duas diagonais** é sempre igual a 15!*

Tarefa: Escreva um programa em C que receba como entrada uma matriz quadrada de ordem N ($1 < N < 100$) e imprima na tela se trata-se de um quadrado mágico ou não.

Entrada: A entrada contém um inteiro indicando a dimensão da matriz e em seguida os elementos da matriz.

Saída: O seu programa deve imprimir “Verdadeiro”, quando a matriz de entrada for um quadrado mágico, ou “Falso” caso contrário.

Exemplo 1:

ENTRADA

```
5
11 24 7 20 3
4 12 25 8 16
17 5 13 21 9
10 18 1 14 22
23 6 19 2 15
```

SAÍDA

“Verdadeiro”

Exemplo 2:

ENTRADA

```
4
4 14 15 1
9 7 7 12
5 11 10 8
16 2 3 13
```

SAÍDA

“Falso”

Atividade 2: Complete o programa em C abaixo, para que ele realize a **multiplicação** e também a **soma** de duas **matrizes quadradas** e em seguida exiba na tela tanto as matrizes de entrada quanto a matriz resultante de cada operação.

Copie o código abaixo (onde as matrizes de entrada e a resultante são inicializadas) e implemente a função **reportarOpMatrizes()**. **Siga as instruções descritas após o código!**

```
#include <stdio.h>
#include <stdlib.h>
#define DIM 2 //Alterar esse valor ao testar outras dimensões!
void reportarOpMatrizes(int A[DIM][DIM],int B[DIM][DIM],int C[DIM][DIM],char op);
int main() {
    int A[DIM][DIM] = { {-4, 2}, {3, 7} }; //"Comente" ao testar o caso 3x3
    int B[DIM][DIM] = { {1, -2}, {12, 9} };
    //int A[DIM][DIM] = { {3, 4, 5}, {8, -2, 7}, {9, 0, 5} }; //"Descomente" ao testar
    //int B[DIM][DIM] = { {5, 1, 2}, {-8, 2, 2}, {11, 4, 9} };

    int C[DIM][DIM] = { 0 }; //Inicializa com 0 TODOS os elementos de C!

    /* MULTIPLICAÇÃO */
    //Implementar aqui!

    reportarOpMatrizes(A, B, C, 'X');
    printf("\n\n");

    /* SOMA */
    //Implementar aqui!

    reportarOpMatrizes(A, B, C, '+');
    return 0;
}
void reportarOpMatrizes(int A[DIM][DIM],int B[DIM][DIM],int C[DIM][DIM],char op){
    //IMPLEMENTAR!
}
```

Atenção! As operações de multiplicação e soma devem ser implementadas na função **main()**. A função de “reportar” irá apenas exibir na tela os resultados **formatados**.

Teste os casos de entrada!

Após efetuar as operações, sua função de relatório deverá imprimir as duas matrizes de entrada, a operação (multiplicação e, a seguir, soma) e a matriz resultante **exatamente como nos dois exemplos abaixo (2x2 e 3x3)**.

Exemplo 1 (matrizes 2x2):

-4	2	X	1	-2	=	20	26
3	7		12	9		87	57
-4	2	+	1	-2	=	-3	0
3	7		12	9		15	16

Após testar as matrizes 2x2, **“comente”** essas linhas e **“descomente”** as matrizes 3x3. **Lembre-se de alterar a constante DIM para 3!**

Exemplo2 (3x3):

3	4	5	X	5	1	2	=	38	31	59
8	-2	7		-8	2	2		133	32	75
9	0	5		11	4	9		100	29	63
3	4	5	+	5	1	2	=	8	5	7
8	-2	7		-8	2	2		0	0	9
9	0	5		11	4	9		20	4	14

Teste também outros casos por sua própria conta!

Atividade 3 (OBI 2011):

Pedrinho é um garoto fascinado por jogos de tabuleiro. Nas férias de janeiro, ele aprendeu um jogo chamado Campo Minado, que é jogado em um tabuleiro com N células dispostas na horizontal. O objetivo desse jogo é determinar, para cada célula do tabuleiro, o número de minas explosivas nos arredores da mesma (que são a própria célula e as células imediatamente vizinhas à direita e à esquerda, caso existam). Por exemplo, considere um tabuleiro com 5 células, configuradas da seguinte forma (# significa “mina”):

[][#][#][][#]

A primeira célula não possui nenhuma mina explosiva, mas é vizinha de uma célula que possui. Nos arredores da segunda célula temos duas minas (ela mesma e a vizinha à direita), e o mesmo acontece para a terceira e quarta células; a quinta célula só tem uma mina explosiva em seus arredores. A resposta para o caso acima seria:

[1][2][2][2][1]

Pedrinho sabe que você está cursando Prog-I e resolveu lhe pedir para escrever um programa em C que, dado um tabuleiro, imprima o número de minas na vizinhança de cada posição.

A entrada deverá ser lida de teclado. A primeira linha da entrada contém um inteiro N indicando o número de células no tabuleiro. O tabuleiro é dado na próxima linha. O i-ésimo elemento da segunda linha contém 0 se não existe uma mina na i-ésima célula do tabuleiro e 1 caso contrário.

A saída é composta por N valores e deverá ser escrita na tela. O i-ésimo valor da saída contém o número de minas explosivas nos arredores da i-ésima célula do tabuleiro.

Exemplos de entrada e saída:**ENTRADA:**

5
0 1 1 1 0

8
0 1 1 1 0 0 0 1

SAÍDA:

1 2 3 2 1

1 2 3 2 1 0 1 1

Atividade 4:

Crie um **novo tipo de dados** para representar uma **Pessoa** com os campos **idade** e **peso**. Agora, implemente uma função que, dado um vetor de Pessoas, ordene esse vetor em ordem **decrecente** de **idade** e imprima o vetor ordenado na tela.

Finalmente, implemente a função **main**, onde um vetor de Pessoas (máximo de 50) deverá ser preenchido pelo usuário e em seguida ordenado (chamando a função definida anteriormente).