

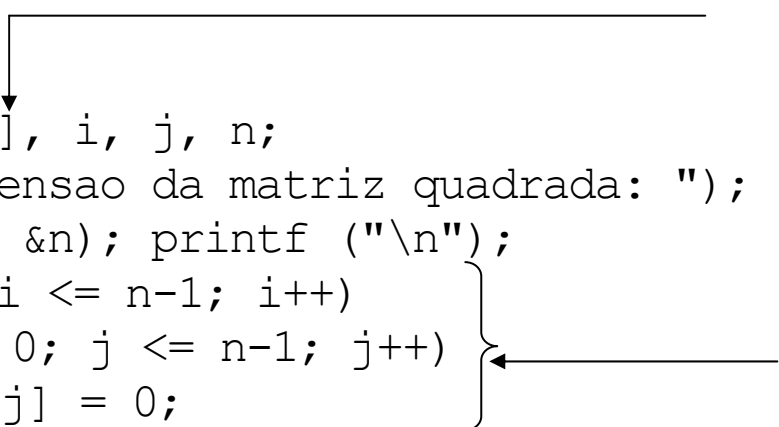
# Capítulo 5: Variáveis Estruturadas

## Programa 5.1: Exercícios básicos com matrizes

```
#include <stdio.h>
void main ()
{
    int A[10][10] = {0}, i, j, n;
    printf ("Dimensao da matriz quadrada: ");
    scanf ("%d", &n); printf ("\n");
    for (i = 0; i <= n-1; i++) {
        for (j = 0; j <= n-1; j++)
            printf ("%4d", A[i][j]);
        printf ("\n");
    }
}
```

- Para não zerar todo o espaço reservado para a matriz:

```
#include <stdio.h>
void main ()
{
    int A[10][10], i, j, n;
    printf ("Dimensao da matriz quadrada: ");
    scanf ("%d", &n); printf ("\n");
    for (i = 0; i <= n-1; i++)
        for (j = 0; j <= n-1; j++)
            A[i][j] = 0;
    for (i = 0; i <= n-1; i++) {
        for (j = 0; j <= n-1; j++)
            printf ("%4d", A[i][j]);
        printf ("\n");
    }
}
```

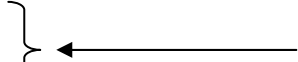
A diagram with two arrows. One arrow starts from the right side of the page and points to the line 'int A[10][10], i, j, n;' in the second code block. The other arrow starts from the right side of the page and points to the inner 'for' loop 'for (j = 0; j <= n-1; j++)' in the same code block, which is enclosed in a curly brace.

- Colocando **5** na diagonal principal, assim:

```

int A[10][10], i, j, n;
printf ("Dimensao da matriz quadrada: ");
scanf ("%d", &n); printf ("\n");
for (i = 0; i <= n-1; i++)
    for (j = 0; j <= n-1; j++)
        if (i == j) A[i][j] = 5;
        else A[i][j] = 0;
for (i = 0; i <= n-1; i++) {
    for (j = 0; j <= n-1; j++)
        printf ("%4d", A[i][j]);
    printf ("\n");
}

```



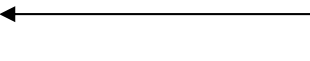
- Ou então assim:

```

int A[10][10], i, j, n;

printf ("Dimensao da matriz quadrada: ");
scanf ("%d", &n); printf ("\n");
for (i = 0; i <= n-1; i++)
    for (j = 0; j <= n-1; j++)
        A[i][j] = 0;
for (i = 0; i <= n-1; i++)
    A[i][i] = 5;
for (i = 0; i <= n-1; i++) {
    for (j = 0; j <= n-1; j++)
        printf ("%4d", A[i][j]);
    printf ("\n");
}

```



- Colocando **5** na primeira linha:

```

printf ("Dimensao da matriz quadrada: ");
scanf ("%d", &n); printf ("\n");
for (i = 0; i <= n-1; i++)
    for (j = 0; j <= n-1; j++)
        A[i][j] = 0;
for (i = 0; i <= n-1; i++)
    A[i][i] = 5;
for (i = 0; i <= n-1; i++) } ←
    A[0][i] = 5;
for (i = 0; i <= n-1; i++) {
    for (j = 0; j <= n-1; j++)
        printf ("%4d", A[i][j]);
    printf ("\n");
}

```

- Colocar 5 na última linha e na primeira e última coluna, obtendo:

Dimensao da matriz quadrada: 7

```

5   5   5   5   5   5   5
5   5   0   0   0   0   5
5   0   5   0   0   0   5
5   0   0   5   0   0   5
5   0   0   0   5   0   5
5   0   0   0   0   5   5
5   5   5   5   5   5   5

```

- Colocar 5 somente da diagonal principal para baixo, obtendo:

Dimensao da matriz quadrada: 7

5	0	0	0	0	0	0
5	5	0	0	0	0	0
5	5	5	0	0	0	0
5	5	5	5	0	0	0
5	5	5	5	5	0	0
5	5	5	5	5	5	0
5	5	5	5	5	5	5

- Obter triângulos de Pascal:

1						
1	1					
1	2	1				
1	3	3	1			
1	4	6	4	1		
1	5	10	10	5	1	
1	6	15	20	15	6	1

Os elementos da 1.a coluna e da diagonal principal são iguais a 1.

Os outros elementos (abaixo da diagonal principal) são dados por:  $A[i][j] = A[i-1][j-1] + A[i-1][j]$ .

Não escrever os elementos acima da diagonal principal

Para montar: Linhas:  $i \rightarrow [2 .. n-1]$ , Colunas:  $j \rightarrow [1 .. i-1]$

Para escrever: Linhas:  $i \rightarrow [0 .. n-1]$ , Colunas:  $j \rightarrow [0 .. i]$

## Programa 5.2: Leitura e escrita de matriz

```
#include <stdio.h>

void main ()
{
    int A[10][10], i, j, m, n;

    printf ("Numero de linhas e colunas da matriz: ");
    scanf ("%d%d", &m, &n); printf ("\n");
    for (i = 0; i <= m-1; i++)
        for (j = 0; j <= n-1; j++) {
            printf ("A[%d][%d]: ", i, j);
            scanf ("%d", &A[i][j]);
        }
    printf ("\n");
    printf ("Matriz: \n\n");
    for (i = 0; i <= m-1; i++) {
        for (j = 0; j <= n-1; j++)
            printf ("%4d", A[i][j]);
        printf ("\n");
    }
}
```

- Construir e escrever no programa anterior uma matriz **B**(**m x n**) tal que

$$\begin{aligned}
 B[i][j] &= 0, \text{ se } A[i][j] = 0; \\
 B[i][j] &= 1, \text{ se } A[i][j] > 0 \text{ e} \\
 B[i][j] &= -1, \text{ se } A[i][j] < 0;
 \end{aligned}$$

- Fazer um programa para:
  - Ler as dimensões **m** e **n** de uma matriz **A** destinada a conter somente **zero's** e **um's**;
  - Ler os elementos da matriz **A**; insistir na digitação correta de **zero's** e **um's**;
  - Escrever a matriz **A** lida;
  - Construir uma matriz **B** (**m x n**) tal que cada elemento **B[i][j]** seja a soma dos vizinhos dos elementos de **A[i][j]**.

- **Exemplo:** para  $m = n = 10$ :

**Matriz A**

**Matriz B**

<b>1</b>	<b>1</b>	0	0	1	0	1	1	0	1	<b>1</b>	1	2	3	2	4	3	3	3	0
<b>0</b>	<b>0</b>	0	1	1	0	1	1	0	0	3	4	4	4	3	6	5	6	5	2
0	1	1	0	1	0	1	1	<b>1</b>	<b>0</b>	2	2	2	4	2	5	5	6	3	1
1	0	0	0	0	0	1	1	<b>0</b>	<b>0</b>	2	5	4	4	2	5	4	6	<b>5</b>	<b>3</b>
0	1	1	0	1	0	1	0	<b>1</b>	<b>1</b>	4	4	2	3	1	4	3	6	4	2
1	1	0	0	1	0	0	1	1	0	2	3	3	4	3	5	4	5	5	4
0	0	0	0	<b>1</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>0</b>	4	5	4	4	3	3	4	4	3	2
1	1	1	1	<b>0</b>	<b>0</b>	<b>1</b>	0	0	0	1	3	4	5	<b>5</b>	<b>4</b>	2	3	2	1
0	0	1	1	<b>1</b>	<b>0</b>	<b>0</b>	0	0	0	4	6	6	6	5	4	3	3	2	1
1	1	0	1	1	1	0	1	1	0	1	2	4	4	4	2	2	1	1	1

- Mudar o programa 5.2 para ler e imprimir uma matriz quadrada.
- Montar e imprimir um vetor de reais contendo a média aritmética dos elementos de cada linha da matriz; o modelo dos resultados é:

	Elementos da Matriz				Media das linhas
-----					
Linha 0:	21	-76	-32	345	64.50
Linha 1:	21	98	-54	0	16.25
Linha 2:	-75	-38	2	-6	-29.25
Linha 3:	23	76	54	-87	16.50
-----					

- Montar e imprimir um vetor de inteiros contendo o maior elemento (em valor relativo) de cada coluna da matriz; calcular e imprimir também a média desses maiores elementos; modelo dos resultados:

	Elementos da Matriz				Media das linhas
-----					
Linha 0:	23	64	-75	19	7.75
Linha 1:	-83	92	49	-38	5.00
Linha 2:	53	-98	21	4	-5.00
Linha 3:	-5	10	-28	58	8.75
-----					
Maior:	53	92	49	58	63.00

- Montar e imprimir um vetor de inteiros contendo os elementos da diagonal principal e da diagonal secundária da matriz; calcular e imprimir também a média dos elementos dessas diagonais; modelo dos resultados:

	Elementos da Matriz				Media das linhas
-----					
Linha 0:	23	64	-75	19	7.75
Linha 1:	-83	92	49	-38	5.00
Linha 2:	53	-98	21	4	-5.00
Linha 3:	-5	10	-28	58	8.75
-----					
Maior:	53	92	49	58	63.00
Diag. princ.:	23	92	21	58	48.50
Diag. secund.:	-5	-98	49	19	-8.75



- **Programa 5.3: Cálculo dos números primos de 1 a n, usando vetores – Fazer fluxograma**

```

#include <stdio.h>
#define TRUE 1
#define FALSE 0
#define N 1001

void main () {
    int n, i, j, d, vet[N];
    scanf ("%d", &n);
    if (n <= 0 || n >= N)
        printf ("Dados incompatíveis");
    else {
        for (i = 1; i <= n; i++)
            vet[i] = TRUE;
        d = 1;
        while (d*d <= n) {
            for (d++; vet[d] == FALSE; d++);
            for (j = d*d; j <= n; j += d)
                vet[j] = FALSE;
        }
    }
    for (i = 1, j = 0; i <= n; i++)
        if (vet[i] == TRUE) {
            printf ("%10d", i); j++;
            if (j % 5 == 0) printf ("\n");
        }
}

```

**Programa 5.4** - Ler um vetor desordenado de números inteiros, eliminar todas as duplicatas e escrever o vetor sem elementos repetidos.

**Programa 5.5** - Ler dois vetores **A** e **B** de números reais com um mesmo número **n** de elementos, formar e escrever outros dois vetores **C** e **D** também de números reais, tais que

$$\begin{aligned} C[i] &= \max (A[i], B[i]) \text{ e} \\ D[i] &= \text{média} (A[i], B[i]) \text{ para} \\ &0 \leq i \leq n-1 \end{aligned}$$

**Programa 5.6** - Ler os elementos de uma matriz **A** e dois inteiros **i** e **j**. Escrever a matriz **A**, trocar todos os elementos da linha **i** com os da linha **j** de **A** e escrever novamente a matriz **A**.

**Programa 5.7** – Ler e escrever uma matriz (20 por 20) de números inteiros. Supondo que ela seja virtualmente subdividida em 25 sub-matrizes (4 por 4), construir e escrever uma outra matriz (5 por 5) na qual cada elemento contenha a soma dos elementos de cada uma das 25 sub-matrizes mencionadas.

## Programa 5.8: Leitura de cadeias de caracteres

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

```
void main ( ) {  
    char cad [10]; int i; char xxx [] = "abcde";  
  
    for (i = 1; i <= 2; i++) {  
        printf ("%d", i);  
        printf ("\tDigite cad: "); scanf ("%s", cad);  
        printf ("\n\tcad: %s; xxx: %s; i: %d;", cad, xxx, i);  
        printf ("\n\n\tDigite xxx: "); scanf ("%s", xxx);  
        printf ("\n\tcad: %s; xxx: %s; i: %d;", cad, xxx, i);  
        printf ("\n\n");  
    }  
}
```

Rode o programa para:

- 1) 12345      qwertgioup
- 2) x            4

Rode para:

- 1) 123456789012345      x
- 2) x                      4

Rode para:

- 1) 12345      abababababababababababababbababaabb
- 2) x            t

Mude **int i;** para **char i;**

## Programa 5.9: Comprimento de cadeias de caracteres

```
#include <stdio.h>

void main () {
    char cad[10]; int i, j; char xxx[] = "abcde";

    for (i = 1; i <= 2; i++) {
        printf ("%d", i);
        printf ("\tDigite cad: "); scanf ("%s", cad);
        printf ("\n\tcad: %s; tamanho: %d; xxx: %s; tamanho: %d;",
            cad, strlen(cad), xxx, strlen(xxx));
        printf ("\n\n\tDigite xxx: "); scanf ("%s", xxx);
        printf ("\n\tcad: %s; tamanho: %d; xxx: %s; tamanho: %d;",
            cad, strlen(cad), xxx, strlen(xxx));
        printf ("\n\n");
        printf ("\tcad: ");
        for (j = 0; j <= 50; j++) printf ("%c", cad[j]);
        printf ("\n\n");
        printf ("\txxx: ");
        for (j = 0; j <= 50; j++) printf ("%c", xxx[j]);
        printf ("\n\n");
    }
}
```

## Programa 5.10: Cópia, concatenação e comparação de

## cadeias

```
#include <stdio.h>
#include <string.h>
typedef char cadeia [20];

void main () {
    int i, cmp;
    cadeia a, b, x = "xxxxx", y = "yyy";
    for (i = 0; i <= 19; i++)
        printf ("%4c%4c%4c%4c\n", a[i], b[i], x[i], y[i]);
    strcpy (a, x); strcpy (b, y);
    printf ("a = %s b = %s x = %s y = %s\n\n", a, b, x, y);
    strcat (a, b);
    printf ("a = %s b = %s\n\n", a, b);
    cmp = strcmp (a, b);
    if (cmp == 0) printf ("a e b sao iguais");
    else if (cmp < 0) printf ("a eh menor que b");
    else printf ("a eh maior que b");
}
```

## Exercício: Cadeias de caracteres binários

a) Programa para digitar frases e repetir as frases digitadas:

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <string.h>

void main ( ) {
    char c, frase[50];
    printf ("Trabalho com frase? (s/n): ");
    c = getche ();
    while (c == 's' || c == 'S') {
        printf ("\n\n\tDigite uma frase: ");
        fflush (stdin);
        gets (frase);
        printf ("\n\tFrase processada: %s", frase);
        printf ("\n\nTrabalho com frase? (s/n): ");
        c = getche ();
    }
}
```

b) Incluir o comprimento da frase

Função **strlen (frase)**: número de caracteres da frase

```
printf ("\n\tComprimento: %d", strlen (frase));
```

c) Escrever o primeiro e o terceiro caractere da frase:

Primeiro caractere: frase[0]

Terceiro caractere: frase[2]

```
printf ("\n\tPrimeiro caractere: %c", frase[0]);  
printf ("\n\tTerceiro caractere: %c", frase[2]);
```

d) Escrever os 10 primeiros caracteres da frase:

```
#include <stdio.h>  
#include <stdlib.h>  
#include <conio.h>  
#include <string.h>
```

```
void main ( ) {  
    int i, n;  
    char c, frase[50];  
    printf ("Trabalho com frase? (s/n): ");  
    c = getche ();  
    while (c == 's' || c == 'S') {  
        printf ("\n\n\tDigite uma frase: ");  
        fflush (stdin);  
        gets (frase); n = strlen (frase);  
        printf ("\n\tFrase processada: %s", frase);  
        printf ("\n\tComprimento: %d", n);  
        printf ("\n\tDez primeiros caracteres: ");  
        for (i = 0; i <= 9; i++)  
            printf ("%c", frase[i]);  
        printf ("\n\nTrabalho com frase? (s/n): ");  
        c = getche ();  
    }  
}
```

e) Trocar as letras minúsculas por maiúsculas:

ASCII das letras:

A = 65	a = 97
B = 66	b = 98
C = 67	c = 99
⋮	⋮
Z = 90	z = 122

<b>Maiúscula = Minúscula - 32</b>
-----------------------------------

Antes de

```
printf ("\n\tFrase processada: %s", frase);

for (i = 0; i <= n-1; i++)
    if (frase[i] >= 'a' && frase[i] <= 'z')
        frase[i] = frase[i] - 32;
```

f) Pedir frases contendo somente letras:

```
printf ("\n\n\tFrase contendo soh letras: ");
fflush (stdin);
gets (frase); n = strlen (frase);
valida = 1;
for (i = 0; i <= n-1 && valida == 1; i++)
    if (isalpha (frase[i]) == 0)
        valida = 0;
printf ("\n\tFrase processada: %s", frase);
printf ("\n\tComprimento: %d", n);
if (valida == 1)
    printf ("\n\tfrase valida");
else
    printf ("\n\tfrase invalida");
```



g) Pedir frases contendo somente dígitos binários:

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <string.h>

void main ( ) {
    int i, n;
    char c, binario[50], valido;
    printf ("Trabalho com binarios? (s/n): ");
    c = getche ();
    while (c == 's' || c == 'S') {
        printf ("\n\n\tNumero binario: ");
        fflush (stdin);
        gets (binario); n = strlen (binario);
        valido = 1;
        for (i = 0; i <= n-1 && valido == 1; i++)
            if (binario[i] != '0' && binario[i] != '1')
                valido = 0;
        printf ("\n\tBinario digitado: %s", binario);
        printf ("\n\tComprimento: %d", n);
        if (valido == 1)
            printf ("\n\tbinario valido");
        else
            printf ("\n\tbinario invalido");
        printf ("\n\nTrabalho com binarios? (s/n): ");
        c = getche ();
    }
}
```

**h)** Converter os binários para base 10:

```
if (valido == 0)
    printf ("\n\tbinario invalido");
else {
    base10 = 0;
    for (i = 0; i <= n-1; i++)
        base10 = 2 * base10 + binario[i] - '0';
    printf ("\n\tNa base 10: %ld", base10);
}
```

**i)** Converter para a base 10 somente os binários de 16 bits.

**j)** Considerando que o binário de 16 bits é o complemento de 2 de algum número, calcular e imprimir esse número.

**k)** Achar a soma de todos os números dados por seu complemento de 2 em 16 bits.

## Programa 5.11: Tipos enumerativos

```
#include <stdio.h>
#define N 10
enum diasemana {dom, seg, ter, qua, qui, sex, sab};
typedef enum diasemana diasem;
typedef char nome[N];

void main () {
    diasem hoje, ontem, amanha;
    nome nomedia[7] = {"domingo", "segunda", "terca",
        "quarta", "quinta", "sexta", "sabado"};
    do {
        printf ("Dia de hoje: ");
        scanf ("%d", &hoje);
        if (hoje >= dom && hoje <= sab) {
            ontem = (hoje + 6) % 7;
            amanha = (hoje + 1) % 7;
            printf ("hoje = %s; ontem = %s; amanha = %s;\n\n",
                nomedia[hoje], nomedia[ontem], nomedia[amanha]);
        }
    } while (hoje >= dom && hoje <= sab);
}
```

Faça mudanças para Domingo ser 1 e Sábado ser 7.

## Programa 5.12: Alterações no programa anterior

```

#include <stdio.h>
#define N 10
enum diasemana {dom = 1, seg, ter, qua, qui, sex, sab};
typedef enum diasemana diasem;
typedef char nome[N];

void main () {
    diasem hoje, ontem, amanha;
    nome nomedia[8] = {"", "domingo", "segunda", "terca",
        "quarta", "quinta", "sexta", "sabado"};
    do {
        printf ("Dia de hoje: ");
        scanf ("%d", &hoje);
        if (hoje >= dom && hoje <= sab) {
            ontem = (hoje + 5) % 7 + 1;
            amanha = hoje % 7 + 1;
            printf ("hoje = %s; ontem = %s; amanha = %s;\n\n",
                nomedia[hoje], nomedia[ontem], nomedia[amanha]);
        }
    } while (hoje >= dom && hoje <= sab);
}

```

Faça modificações para ler cadeias de caracteres e não números.

## Programa 5.13: Melhoramentos na interface do programa anterior

```
#include <stdio.h>
#include <string.h>
#define N 10

enum diasemana {dom = 1, seg, ter, qua, qui, sex, sab};
typedef enum diasemana diasem;
typedef char nome[N];

void main () {
    diasem hoje, ontem, amanha;
    nome nomehoje;
    nome nomedia[8] = {"", "domingo", "segunda", "terca",
        "quarta", "quinta", "sexta", "sabado"};
    do {
        printf ("Dia de hoje: ");
        scanf ("%s", nomehoje);
        for (hoje = 1; hoje <= 7; hoje++)
            if (strcmp (nomehoje, nomedia[hoje]) == 0) break;
        if (hoje >= dom && hoje <= sab) {
            ontem = (hoje + 5) % 7 + 1;
            amanha = hoje % 7 + 1;
            printf ("hoje = %s; ontem = %s; amanha = %s;\n\n",
                nomedia[hoje], nomedia[ontem], nomedia[amanha]);
        }
    } while (hoje >= dom && hoje <= sab);
}
```

## Programa 5.14: Estruturas

```
#include <stdio.h>
#define N 30
typedef char cadeia[N];

enum escolaridade { nenh, pgrau, sgrau, sup };
typedef enum escolaridade escolaridade;

struct data {
    int dia, mes, ano;
};
typedef struct data data;

struct habitante {
    cadeia nome; data dnasc; escolaridade esc;
};
typedef struct habitante habitante;

void main () {
    int i, n; habitante hab [10];
    scanf ("%d", &n);
    for (i = 0; i < n; i++)
        scanf ("%s%d%d%d",
            hab[i].nome, &hab[i].dnasc.dia,
            &hab[i].dnasc.mes, &hab[i].dnasc.ano, &hab[i].esc);
    for (i = 0; i < n; i++)
        printf ("%31s%4d/%4d/%4d%7d\n",
            hab[i].nome, hab[i].dnasc.dia,
            hab[i].dnasc.mes, hab[i].dnasc.ano, hab[i].esc);
}
```

## **Exercício: Estruturas para moradores de um prédio**

```
typedef char cad30[30]; typedef char cad15[15];

enum escolaridade {nenhuma, basica, media, superior};
typedef enum escolaridade escolaridade;

struct zerodez { int nv; cad15 video; };
typedef struct zerodez zerodez;

struct onze17 {
    int nnam; char leulivro; cad30 intpref; };
typedef struct onze17 onze17;

struct dezoitomais {
    char casado, trabalha; int nfilhos;
    escolaridade esc; cad15 profissao; };
typedef struct dezoitomais dezoitomais;

union particularidades {
    zerodez infoz10; onze17 info1117;
    dezoitomais info18;
};
typedef union particularidades particularidades;

struct morador {
    cad30 nome; int idade; particularidades info;
};
typedef struct morador morador;

void main ( ) {
    morador Mor[10]; int i, n;
```

- Exemplo de uma execução:

Numero de moradores: 3

Morador 1

Nome: Mario

Idade: 8

Numero de video-games que possui: 15

video-game preferido: Pokemon

Morador 2

Nome: Alberto

Idade: 16

Numero de namorad(as/os) que ja teve: 5

Ja leu algum livro? (s/n): s

Interprete musical preferido: Oasis

Morador 3

Nome: Marina

Idade: 28

Eh casad(o/a)? (s/n): s

Trabalha? (s/n): s

Numero de filhos: 3

Escolaridade (0, 1, 2, 3): 3

Profissao: Psicologa

MORADORES DO PREDIO:

1) Mario

Idade: 8 anos

Possui 15 video-games

Seu video-game preferido: Pokemon

2) Alberto

Idade: 16 anos

Teve 5 namoradas(os)

Se leu algum livro: s

Interprete musical preferido: Oasis

3) Marina

Idade: 28 anos

Se eh casad(o/a): s

Se trabalha: s

Tem 3 filhos

Grau de Escolaridade: 3

Profissao: Psicologa