

- 1) Faça uma função que receba uma matriz quadrada de ordem n e devolva a somatória da diagonal principal.

```
1  #include <stdio.h>
2
3  // Função que calcula a soma da diagonal principal da matriz
4  int somatorioDiagonal(int n, int matriz[][n]) {
5      int i;
6      int soma = 0;
7      for (i = 0; i < n; i++) {
8          soma += matriz[i][i];
9      }
10     return soma;
11 }
12
13 int main(void) {
14     int n,i,j;
15
16     printf("Digite o tamanho da matriz quadrada (n): ");
17     scanf("%d", &n);
18
19     int matriz[n][n];
20
21     // Leitura dos elementos da matriz
22     printf("Digite os elementos da matriz:\n");
23     for (i = 0; i < n; i++) {
24         for (j = 0; j < n; j++) {
25             printf("Elemento [%d][%d]: ", i, j);
26             scanf("%d", &matriz[i][j]);
27         }
28     }
29
30     // Calcula a soma da diagonal principal
31     int soma = somatorioDiagonal(n, matriz);
32     printf("A soma da diagonal principal e: %d\n", soma);
33
34     return 0;
35 }
36
```

- 2) Faça um programa que: a) Leia uma matriz quadrada de ordem n de elementos do tipo inteiro. b) Divida cada elemento de uma linha pelo elemento da diagonal principal desta matriz. c) Imprima a nova matriz.

```
1  #include <stdio.h>
2
3  int main(void) {
4      int n,i,j;
5      printf("Digite a ordem da matriz quadrada: ");
6      scanf("%d", &n);
7
8      int matriz[n][n];
9
10     // Leitura dos elementos da matriz
11     printf("Digite os elementos da matriz:\n");
12     for ( i = 0; i < n; i++) {
13         for ( j = 0; j < n; j++) {
14             printf("Elemento [%d][%d]: ", i, j);
15             scanf("%d", &matriz[i][j]);
16         }
17     }
18
19     // Nova matriz para armazenar os resultados da divisao
20     float novaMatriz[n][n];
21
22     // Processa cada linha dividindo cada elemento pelo elemento da diagonal principal
23     for ( i = 0; i < n; i++) {
24         int divisor = matriz[i][i];
25         if (divisor == 0) {
26             printf("Erro: divisao por zero na linha %d (elemento diagonal = %d).\n", i, divisor);
27             return 1; // Encerra o programa se houver tentativa de divisao por zero
28         }
29         for ( j = 0; j < n; j++) {
30             novaMatriz[i][j] = (float) matriz[i][j] / divisor;
31         }
32     }
33
34     // Impressao da nova matriz
35     printf("\nMatriz apos a divisao dos elementos de cada linha pelo elemento da diagonal:\n");
36     for ( i = 0; i < n; i++) {
37         for ( j = 0; j < n; j++) {
38             // %.2f formata o valor float com duas casas decimais
39             printf("%.2f ", novaMatriz[i][j]);
40         }
41         printf("\n");
42     }
43
44     return 0;
```

- 3) Construa uma função que receba uma matriz quadrada de ordem n e devolve a soma dos elementos que estão abaixo da diagonal principal.

```
1  #include <stdio.h>
2
3  // Função que calcula a soma dos elementos abaixo da diagonal principal da matriz
4  int somaAbaixoDaDiagonal(int n, int matriz[][n]) {
5      int i,j;
6      int soma = 0;
7      // Percorre as linhas da matriz
8      for ( i = 0; i < n; i++) {
9          /* Para cada linha, os elementos abaixo da diagonal principal
10           * estão nas colunas de índice menor que o índice da linha.
11           * Assim, o laço abaixo percorre de j = 0 até j < i.
12           */
13         for ( j = 0; j < i; j++) {
14             soma += matriz[i][j];
15         }
16     }
17     return soma;
18 }
19
20 int main(void) {
21     int n,i,j;
22
23     printf("Digite a ordem da matriz quadrada: ");
24     scanf("%d", &n);
25
26     int matriz[n][n];
27
28     // Leitura dos elementos da matriz
29     printf("Digite os elementos da matriz:\n");
30     for ( i = 0; i < n; i++) {
31         for ( j = 0; j < n; j++) {
32             printf("Elemento [%d][%d]: ", i, j);
33             scanf("%d", &matriz[i][j]);
34         }
35     }
36
37     // Calcula a soma dos elementos abaixo da diagonal principal
38     int soma = somaAbaixoDaDiagonal(n, matriz);
39     printf("A soma dos elementos abaixo da diagonal principal e: %d\n", soma);
40
41     return 0;
42 }
43
```

4) Faça um programa que dada uma matriz quadrada de ordem n de elementos do tipo inteiro, verifica se a soma dos elementos das diagonais são iguais.

```
1  #include <stdio.h>
2
3  int main(void) {
4      int n,i,j;
5      printf("Digite a ordem da matriz quadrada: ");
6      scanf("%d", &n);
7
8      int matriz[n][n];
9
10     // Leitura dos elementos da matriz
11     printf("Digite os elementos da matriz:\n");
12     for ( i = 0; i < n; i++) {
13         for ( j = 0; j < n; j++) {
14             printf("Elemento [%d][%d]: ", i, j);
15             scanf("%d", &matriz[i][j]);
16         }
17     }
18
19     int somaPrincipal = 0;
20     int somaSecundaria = 0;
21
22     // Calcula a soma da diagonal principal e da diagonal secundária
23     for ( i = 0; i < n; i++) {
24         somaPrincipal += matriz[i][i];
25         somaSecundaria += matriz[i][n - 1 - i];
26     }
27
28     // Exibe os resultados
29     printf("\nSoma da diagonal principal: %d\n", somaPrincipal);
30     printf("Soma da diagonal secundaria: %d\n", somaSecundaria);
31
32     if (somaPrincipal == somaSecundaria) {
33         printf("As somas das diagonais sao iguais.\n");
34     } else {
35         printf("As somas das diagonais sao diferentes.\n");
36     }
37
38     return 0;
39 }
40
```

5) Leia uma matriz de elementos do tipo inteiro com m linhas e n colunas e imprima a soma dos números pares e a soma dos ímpares.

```
1  #include <stdio.h>
2
3  int main(void) {
4      int m, n, i, j;
5
6      // Leitura do número de linhas e colunas
7      printf("Digite o numero de linhas (m): ");
8      scanf("%d", &m);
9      printf("Digite o numero de colunas (n): ");
10     scanf("%d", &n);
11
12     // Declaração da matriz usando Variable Length Arrays (C99)
13     int matriz[m][n];
14
15     // Leitura dos elementos da matriz
16     printf("Digite os elementos da matriz:\n");
17     for ( i = 0; i < m; i++) {
18         for ( j = 0; j < n; j++) {
19             printf("Elemento [%d][%d]: ", i, j);
20             scanf("%d", &matriz[i][j]);
21         }
22     }
23
24     // Variáveis para armazenar as somas dos pares e ímpares
25     int somaPares = 0;
26     int somaImpares = 0;
27
28     // Processamento: percorre a matriz e acumula as somas
29     for ( i = 0; i < m; i++) {
30         for ( j = 0; j < n; j++) {
31             if (matriz[i][j] % 2 == 0) {
32                 somaPares += matriz[i][j];
33             } else {
34                 somaImpares += matriz[i][j];
35             }
36         }
37     }
38
39     // Impressão dos resultados
40     printf("\nSoma dos numeros pares: %d\n", somaPares);
41     printf("Soma dos numeros impares: %d\n", somaImpares);
42
43     return 0;
44 }
```