



```
1 //EXERCICIO 1
2
3 #include <stdio.h>
4
5 int main()
6 {
7     int vetor[10], maior = 0;
8     int cont, pos;
9
10    for (cont = 0; cont < 10; cont++)
11    {
12        printf("Digite o numero na pos %d: ", cont);
13        scanf("%d", &vetor[cont]);
14    }
15
16    for (cont = 0; cont < 10; cont++)
17    {
18
19        if (cont == 0)
20        {
21            maior = vetor[cont];
22            pos = cont;
23        }
24        else if (vetor[cont] > maior)
25        {
26            maior = vetor[cont];
27            pos = cont;
28        }
29    }
30
31    printf("Vetor:");
32    for (cont = 0; cont < 10; cont++)
33    {
34        printf(" %d -> ", vetor[cont]);
35    }
36    printf("FIM!");
37
38    printf("\nMaior numero: %d - Na Pos: %d", maior, pos);
39    return 0;
40 }
```



```
1  //EXERCICIO 2
2
3  #include <stdio.h>
4
5  int main()
6  {
7      int vetorA[10], vetorB[10], vetorC[10];
8      int cont;
9
10     for (cont = 0; cont < 10; cont++)
11     {
12         printf("Vetor A - POS: %d: ", cont);
13         scanf("%d", &vetorA[cont]);
14     }
15
16     printf("\n");
17
18     for (cont = 0; cont < 10; cont++)
19     {
20         printf("Vetor B - POS: %d: ", cont);
21         scanf(" %d", &vetorB[cont]);
22     }
23
24     printf("\nVetor C:");
25
26     for (cont = 0; cont < 10; cont++)
27     {
28         vetorC[cont] = vetorA[cont] - vetorB[cont];
29         printf(" %d -> ", vetorC[cont]);
30     }
31     printf("FIM!");
32
33     return 0;
34 }
```



```
1 //EXERCICIO 3
2
3 #include <stdio.h>
4
5 int main()
6 {
7     int vetor[20], cont, qnt = 0;
8
9     for (cont = 0; cont < 20; cont++)
10    {
11        printf("Vetor - POS: %d: ", cont);
12        scanf("%d", &vetor[cont]);
13    }
14
15    for (cont = 0; cont < 20; cont++)
16    {
17        for (int j = cont++; j < 20; j++)
18        {
19            if (cont != j)
20            {
21
22                if (vetor[cont] == vetor[j])
23                {
24                    printf(" %d - ", vetor[cont]);
25                    qnt++;
26                }
27            }
28        }
29    }
30
31    printf("\nQuantidade de valores repetidos: %d\n", qnt);
32    return 0;
33 }
```

```
1 //EXERCICIO 4
2
3 #include <stdio.h>
4
5 int main()
6 {
7     int N = 0, cont;
8     float vetorA[N], vetorB[N];
9     float vetor1[N], vetor2[N], vetor3[N], vetor4[N];
10
11     printf("Com quantos elementos deseja fazer as operacoes: ");
12     scanf("%d", &N);
13
14     for (cont = 0; cont < N; cont++)
15     {
16         printf("Vetor A - POS: %d: ", cont);
17         scanf(" %f", &vetorA[cont]);
18     }
19
20     printf("\n");
21
22     for (cont = 0; cont < N; cont++)
23     {
24         printf("Vetor B - POS: %d: ", cont);
25         scanf(" %f", &vetorB[cont]);
26     }
27
28     printf("\nSOMA:");
29     for (cont = 0; cont < N; cont++)
30     {
31         vetor1[cont] = vetorA[cont] + vetorB[cont];
32         printf(" %.2f -> ", vetor1[cont]);
33     }
34
35     printf("\nSUBTRACAO:");
36     for (cont = 0; cont < N; cont++)
37     {
38         vetor2[cont] = vetorA[cont] - vetorB[cont];
39         printf(" %.2f -> ", vetor2[cont]);
40     }
41
42     printf("\nDIVISAO:");
43     for (cont = 0; cont < N; cont++)
44     {
45         vetor3[cont] = vetorA[cont] / vetorB[cont];
46         printf(" %.2f -> ", vetor3[cont]);
47     }
48
49     printf("\nMULTIPLICACAO:");
50     for (cont = 0; cont < N; cont++)
51     {
52         vetor4[cont] = vetorA[cont] * vetorB[cont];
53         printf(" %.2f -> ", vetor4[cont]);
54     }
55
56     return 0;
57 }
58
```

```
1 //EXERCICIO 5
2
3 #include <stdio.h>
4
5 int main()
6 {
7     int matriz[5][5];
8     int col, lin;
9
10    for (lin = 0; lin < 5; lin++)
11    {
12        for (col = 0; col < 5; col++)
13        {
14            if (lin == col)
15            {
16                matriz[lin][col] = 1;
17            }
18            else
19            {
20                matriz[lin][col] = 0;
21            }
22        }
23    }
24
25    for (lin = 0; lin < 5; lin++)
26    {
27        for (col = 0; col < 5; col++)
28        {
29            printf("\nPosicao (%d,%d):%d", lin, col, matriz[lin][col]);
30        }
31    }
32
33    return 0;
34 }
35
```

```
1 //EXERCICIO 6
2
3 #include <stdio.h>
4
5 int main()
6 {
7     int matriz[4][4];
8     int col, lin, maior = 0, lin_maior, col_maior;
9
10    for (lin = 0; lin < 4; lin++)
11    {
12        for (col = 0; col < 4; col++)
13        {
14            printf("\nDigite linha e coluna (%d,%d):", lin, col);
15            scanf("%d", &matriz[lin][col]);
16        }
17    }
18
19    for (lin = 0; lin < 4; lin++)
20    {
21        for (col = 0; col < 4; col++)
22        {
23            if (matriz[lin][col] == 0)
24            {
25                maior = matriz[lin][col];
26                lin_maior = lin;
27                col_maior = col;
28            }
29            else if (matriz[lin][col] > maior)
30            {
31                maior = matriz[lin][col];
32                lin_maior = lin;
33                col_maior = col;
34            }
35        }
36    }
37
38    for (lin = 0; lin < 4; lin++)
39    {
40        for (col = 0; col < 4; col++)
41        {
42            printf("\nPosicao (%d,%d):%d", lin, col, matriz[lin][col]);
43        }
44    }
45
46    printf("\nPosicao do maior(%d,%d): %d", lin_maior, col_maior, maior);
47
48    return 0;
49 }
50
```

```

1 //EXERCICIO 7
2
3 #include <stdio.h>
4
5 int main()
6 {
7     float matriz[3][6];
8     float soma_col_impar = 0, soma_col = 0, soma_col_par = 0, qnt_par = 0;
9     int cont, lin, col;
10
11     for (lin = 0; lin < 3; lin++)
12     {
13         for (col = 0; col < 6; col++)
14         {
15             printf("\nDigite os valores (%d,%d): ", lin, col);
16             scanf("%f", &matriz[lin][col]);
17         }
18     }
19
20     for (lin = 0; lin < 3; lin++)
21     {
22         for (col = 0; col < 6; col++)
23         {
24             if (col % 2 != 0)
25             {
26                 soma_col_impar += matriz[lin][col];
27             }
28         }
29     }
30
31     for (lin = 0; lin < 3; lin++)
32     {
33         for (col = 0; col < 6; col++)
34         {
35             if (col % 2 == 0)
36             {
37                 soma_col_par += matriz[lin][col];
38                 qnt_par++;
39             }
40         }
41     }
42
43     printf("\nSoma elementos Colunas impares: %.2f", soma_col_impar);
44     printf("\nMedia elementos Colunas Pares: %.2f\n", soma_col_par / qnt_par);
45
46     for (lin = 0; lin < 3; lin++)
47     {
48         for (col = 0; col < 2; col++)
49         {
50             if (col == 0)
51             {
52                 matriz[lin][5] = 0;
53             }
54             matriz[lin][5] += matriz[lin][col];
55         }
56     }
57
58     for (lin = 0; lin < 3; lin++)
59     {
60         for (col = 0; col < 6; col++)
61         {
62             printf("%.2f ", matriz[lin][col]);
63         }
64         printf("\n");
65     }
66
67     return 0;
68 }

```

```
1 //EXERCICIO 8
2
3 #include <stdio.h>
4
5 int main()
6 {
7     int matriz[50][50], N, M;
8     int soma = 0, media, qnt_par = 0, qnt_impar = 0;
9     int cont, lin, col;
10
11     printf("Digite as dimensões (N x M): ");
12     scanf("%d%d", &N, &M);
13
14     for (lin = 0; lin < N; lin++)
15     {
16         for (col = 0; col < M; col++)
17         {
18             printf("Digite os valores (%d, %d): ", lin, col);
19             scanf(" %d", &matriz[lin][col]);
20         }
21     }
22
23     for (lin = 0; lin < N; lin++)
24     {
25         for (col = 0; col < M; col++)
26         {
27             soma += matriz[lin][col];
28         }
29     }
30
31     media = soma / N * M;
32
33     for (lin = 0; lin < N; lin++)
34     {
35         for (col = 0; col < M; col++)
36         {
37             if (matriz[lin][col] % 2 == 0 && matriz[lin][col] > media)
38             {
39                 qnt_par++;
40             }
41             else if (matriz[lin][col] % 2 != 0 && matriz[lin][col] > media)
42             {
43                 qnt_impar++;
44             }
45         }
46     }
47
48     printf("\nElementos PARES acima da media: %d", qnt_par);
49     printf("\nElementos IMPARES acima da media: %d", qnt_impar);
50     printf("\nElementos TOTAIS acima da media: %d", qnt_par + qnt_impar);
51
52     return 0;
53 }
```