

1. Criar e coletar um vetor [50] inteiro. Calcular e exibir:
 - a. A média dos valores entre 10 e 200;
 - b. A soma dos números ímpares.

Pseudo-Código

Algoritmo: Vetor01

```

inicio
    tipo V = vetor [1..50] inteiros;
    V: vtr;
    inteiro: i;
    real: md;

    s = 0;
    md = 0;

    para(i=1;i<=50;i++)faça
        leia: vtr[i];
        se(vtr[i] >= 10 e vtr[i] <= 200)
            então
                s = s + vtr[i];
                md += 1;
        fimse;
    fimpara;

    md = s / md;
    escreva: md;
    s = 0;

    para(i=1;i<=50;i++)faça
        se(vtr[i] % 2 <> 0)
            então
                s = s + vtr[i];
        fimse;
    fimpara;
    escreva: s;

fim
  
```

Teste de mesa

//Valores coletados no vetor vtr[], valores de 10 a 200 somados e alocados em s, m recebe a quantidade de incidências de valores de 10 a 200.

i	-	1	2	3	..	49	50
vtr[i]	-	23	9	45	..	16	203
s	0	23	23	68	..	84	84
md	0	1	1	2	..	3	3

//Calculo da média dos valores entre 10 e 200 alocado em md.

s	84	0
md	3	28

//Calculo da soma dos números ímpares.

i	1	2	3	..	49	50
vtr[i]	23	9	45	..	16	203
s	23	32	77	..	77	280

Massa de dados

entrada:	23	9	45	..	16	203
saida:	28	280				

2. Criar e coletar um vetor [100] inteiro e exibir:
- O maior e o menor valor,
 - A média dos valores.

Pseudo-Código

algoritmo Vetor2

início

```
    tipo V = vetor [1..100] inteiros;
    V: vtr;
    inteiro: i,s,MA,ME;
    real: md;
```

```
    MA = 0;
    s = 0;
```

```
    para(i=1;i<=100;i++) faça
        leia: vtr[i];
        s += vtr[i];
    fimpara
```

```
    md = i;
    md = s / md; //média dos valores
```

```
    para(i=1;i<=100;i++) faça
        se(vtr[i] > MA)
            então
                MA = vtr[i]; //define maior
    fimse;
    fimpara;
```

```
    ME = MA;
```

```
    para(i=1;i<=100;i++) faça
        se(vtr[i] < ME)
            então
                ME = vtr[i]; //define menor
    fimse;
    fimpara;
```

```
    escreva: MA;
    escreva: ME;
    escreva: md;
```

fim

Teste de mesa

//Valores coletados no vetor vtr[], soma de todos os valores alocado em s.

i	1	2	3	..	99	100
vtr[]	12	345	2	..	76	36
s	12	357	359	..	435	471
md	0	0	0	..	0	0

//Média dos valores.

s	471
md	100 4,71

//Maior maior

i	1	2	3	..	99	100
MA	12	345	345	..	345	345

//Menor maior

i	1	2	3	..	99	100
ME	345	12	2	..	2	2

Massa de dados

entrada:	12	345	2	..	76	36
saida:	345	2	4,71			

3. Criar e coletar valores inteiros nos vetores VT1[3] e VT2[3]. Concatenar esses valores em um 3o vetor (VT3[6]) e mostrar os seus dados.
 P. ex: VT1[1][2][3] VT2[4][5][6] VT3[1][2][3][4][5][6]

Pseudo-Código

```

algoritmo: VetorConcat
início
    tipo V1 = vetor [1..3] inteiros;
    V1: vtrA;

    tipo V2 = vetor [4..6] inteiros;
    V2: vtrB;

    tipo V3 = vetor [1..6] inteiros;
    V3: vtrC;

    para(i=1;i<=6;i++)faça
        se(i<=3)
            então
                leia: vtrA[i];
            senão
                leia: vtrB[i];
        fimse;
    fimpara

    para(i=1;i<=6;i++)faça
        se(i<=3)
            então
                vtrC[i] = vtrA[i];
            senão
                vtrC[i] = vtrB[i];
        fimse;
    fimpara

    para(i=1;i<=6;i++)faça
        escreva: vtrC[i];
    fimpara
fim
  
```

Teste de mesa

i	1	2	3	4	5	6
vtrA	1	2	3	-	-	-
vtrB	-	-	-	4	5	6
vtrC	-	-	-	-	-	-
i	1	2	3	4	5	6
vtrC	1	2	3	4	5	6

Massa de dados

```

entrada:      1 2 3 4 5 6
saida:        1 2 3 4 5 6
  
```

4. Criar e coletar em um vetor [30] real e calcular e exibir:
- A média do grupo;
 - A quantidade de notas acima do grupo;
 - As posições dos valores abaixo da média do grupo.

Pseudo-Código

algoritmo: VetorMedia

início

```
    tipo V = vetor [1..30] reais;  
    V: vtr, vtrPosicoes;  
    real: s,md;  
    inteiro: i,j,nAcima;
```

```
    s ← 0;  
    nAcima ← 0;  
    j ← 1;
```

```
    para(i=1;i<=30;i++)faça  
        leia: vtr[i];  
        s ← s + vtr[i];  
    fimpara;
```

```
    md ← i;  
    md ← s / md;
```

```
    escreva: md; //a. A média do grupo
```

```
    para(i=1;i<=30;i++)faça  
        se(vtr[i] > md)  
            então  
                nAcima += 1;  
        fimse;  
    fimpara;
```

```
    escreva: nAcima; //A quantidade de notas acima do grupo
```

```
    para(i=1;i<=30;i++)faça  
        se(vtr[i] < md)  
            então  
                vtrPosicoes[j] ← i;  
                j += 1;  
        fimse;  
    fimpara;
```

```
    para(i=1;i<=j;i++)faça  
        escreva: vtrPosicoes[i]; //As posições dos valores abaixo da média do grupo.  
    fimpara;
```

fim

5. Criar e coletar em um vetor [20] inteiro. Calcule e exiba, segundo:

$$\sum_{i=1}^{10} (A[i] - A[21-i])$$

Pseudo-Código

algoritmo:

inicio

 tipo V = vetor [1..20] de inteiros;

 V: vtr;

 inteiro: i;

 j ← 21;

 para(i=1;i<=20;i++)faça

 leia: vtr[i]; //coleta valores int no vetor vtr[]

 fimpara;

 para(i=1;i<=10;i++)faça

 s ← s + (vtr[i] - vtr[j-i]);

 fimpara;

 escreva: s;

fim

6. Criar e coletar em um vetor [20] com números aleatórios. Classificar este vetor em ordem crescente e mostre os dados.

Pseudo-Código

algoritmo: ordVetor

inicio

```
    tipo V =  vetor [1..20] inteiro;
    V: vtr;
    inteiro: i,j,aux;

    aux ← 0;

    para(i=1;i<=20;i++)faça
        vtr[i] ← random();
    fimpara;

    para(i=1;i<=20;i++)faça
        escreva = vtr[i];
    fimpara;

    para(i=2;i<=20;i++)faça
        para(j=20;j>=1;j--)faça
            se(vtr[j-1] > vtr[j])
                então
                    aux = vtr[j-1];
                    vtr[j-1] = vtr[j];
                    vtr[j] = aux;
                fimse;
        fimpara;
    fimpara;

    para(i=1;i<20;i++)faça
        escreva: vtr[i];
    fimpara;

    escreva: Fbusca();
```

fim

7. A partir do exercício 6 (vetor classificado) solicitar um valor qualquer e verificar a sua existência no vetor (utilizar pesquisa binária).

Pseudo-Código

Algoritmo: Fbusca(inteiro vrt[]) inteiro

inicio

 inteiro: inicial, meio, final, busca;
 boolean: nroLocalizado;

 inicial \leftarrow 1;
 final \leftarrow 20;
 nroLocalizado \leftarrow Falso;

 enquanto ((inicial \leq final) e não nroLocalizado) faça

 meio \leftarrow (inicial + final) / 2;

 se(vrtB[meio] = busca)

 então

 nroLocalizado \leftarrow Verdadeiro;

 fimse;

 se(vrtB[meio] > busca)

 então

 final \leftarrow meio - 1;

 senão

 inicial \leftarrow meio + 1;

 fimse;

 fimenquanto

 retorna: nroLocalizado;

fim

8. Criar e carregar uma matriz [4][3] inteiro com quantidade de produtos vendidos em 4 semanas. Calcular e exibir:
- A quantidade de cada produto vendido no mês;
 - A quantidade de produtos vendidos por semana;
 - O total de produtos vendidos no mês.

Pseudo-Código

algoritmo: qtdProdutos

início

```
    tipo M = matriz [1..4, 1..3] de inteiros;  
    M: mtz;  
    inteiro: i,j,s;
```

```
    //Lê os valores de cada posição na Matriz  
    para(i=1;i<=4;i++)faça  
        para(j=1;j<=3;j++)faça  
            leia: mtz[i, j];  
        fimpara;  
    fimpara;
```

```
    //a. A quantidade de cada produto vendido no mês;  
    para(i=1;i<=3;i++)faça  
        para(j=1;j<=4;j++)faça  
            s += mtz[j, i];  
        fimpara;  
        escreva: s;  
        s = 0;  
    fimpara;
```

```
    //b. A quantidade de produtos vendidos por semana;  
    para(i=1;i<=4;i++)faça  
        para(j=1;j<=3;j++)faça  
            s += mtz[i,j];  
        fimpara;  
        escreva: s;  
        s = 0;  
    fimpara;
```

```
    //c. O total de produtos vendidos no mês.  
    para(i=1;i<=4;i++)faça  
        para(j=1;j<=3;j++)faça  
            s += mtz[i,j];  
        fimpara;  
    fimpara;  
    escreva: s;  
    s = 0;
```

fim

9. Criar e carregar uma matriz [4][4] com valores aleatórios, sendo que a diagonal principal terá seus dados carregados no programa segundo:

```
1
  4
    16
      64
```

Pseudo-Código

algoritmo: qtdProdutos

início

 tipo M = matriz [1..4, 1..4] de inteiros;

 M: mtz;

 inteiro: i,j;

 i = 1;

 j = 1;

 aux = 1;

 mtz[i, j] = aux;

 para(i=1;i<=4;i++)faça

 para(j=2;j<=4;j++)faça

 se(i == j)

 entao

 aux *= 4;

 mtz[i, j] = aux;

 senao

 mtz[i, j] = rand();

 fimse;

 fimpara;

 fimpara;

 para(i=1;i<=4;i++)faça

 para(j=1;j<=4;j++)faça

 escreva: mtz[i, j];

 fimpara;

 fimpara;

10. Criar uma matriz [8][8] onde o programa irá carregar segundo:

casa	1	2	3	4	..	*Exibir a soma
valor	1	2	4	8	..	dos valores

Pseudo-Código

algoritmo: qtdProdutos

início

 tipo M = matriz [1..8, 1..8] de inteiros;

 M: mtz;

 inteiro: i,j;

 aux = 1;

 s = 1;

 para(i=1;i<=8;i++)faça

 para(j=1;j<=8;j++)faça

 mtz[i, j] = aux;

 fimpara;

 aux *= 2;

 s += aux;

 fimpara;

 para(i=1;i<=8;i++)faça

 para(j=1;j<=8;j++)faça

 escreva: mtz[i, j];

 fimpara;

 fimpara;

fim

aux	1	2	4	8	16	32	64	128	..
s	1	3	7	15	31	63	127	255	..
i	1	1	1	1	1	1	1	1	..
j	1	2	3	4	5	6	7	8	..
mtz[i, j]	1	2	4	8	16	32	64	128	..

11. Criar uma matriz [8][8] inteiro e o programa irá carregar segundo:

```
1 1 1 1 1 1 1 1
1 2 2 2 2 2 2 1
1 2 3 3 3 3 2 1
1 2 3 4 4 3 2 1
1 2 3 4 4 3 2 1
1 2 3 3 3 3 2 1
1 2 2 2 2 2 2 1
1 1 1 1 1 1 1 1
```

Pseudo-Código

algoritmo: qtdProdutos

início

```
    tipo M = matriz [1..8, 1..8] de inteiros;
    M: mtz;
    inteiro: i,j;

    para(i=1;i<=8;i++)faça
        para(j=1;j<=8;j++)faça
            se((i==1 ou i==8) ou (j==1 ou j==8))
                então
                    mtz[i, j] = 1;
            senao
                se((i==2 ou i==7) ou (j==2 ou j==7))
                    então
                        mtz[i, j] = 2;
                senao
                    se((i==3 ou i==6) ou (j==3 ou j==6))
                        então
                            mtz[i, j] = 3;
                senao
                    se((i==4 ou i==5) ou (j==4 ou j==5))
                        então
                            mtz[i, j] = 4;
            fimse;
        fimse;
    fimpara;
fimpara;

para(i=1;i<=8;i++)faça
    para(j=1;j<=8;j++)faça
        escreva: mtz[i, j];
    fimpara;
fimpara;
```

12. Carregar códigos das peças em um tabuleiro de xadrez, onde:

Código:	1	2	3	4	5	6	7
Peças:	Peão	Torre	Bispo	Cavalo	Rainha	Rei	Vazio

Calcular e mostrar a soma das peças do tabuleiro,

Não pode utilizar Estrutura de Decisão e Escolha Caso na contagem das peças

Pseudo-Código

algoritmo: carregaTabuleiro

início

```
    tipo M = matriz [1..8, 1..8] de inteiros;
    M: mtz;
    inteiro: i,j;

    aux = 0;
    s = 0;

    para(i=1;i<=8;i++)faça
        para(j=1;j<=8;j++)faça
            se(i>=3 e i<=6)
                então
                    mtz[i, j] = 7;
            senao
                se(i==2 ou i==7)
                    então
                        mtz[i, j] = 1;
                senao
                    se((i==1 ou i==8) e (j==1 ou j==8))
                        então
                            mtz[i, j] = 2;
                    senao
                        se((i==1 ou i==8) e (j==2 ou j==7))
                            então
                                mtz[i, j] = 4;
                    senao
                        se((i==1 ou i==8) e (j==3 ou j==6))
                            então
                                mtz[i, j] = 3;
                    senao
                        se((i==1 ou i==8) e j==4)
                            então
                                mtz[i, j] = 6;
                    senao
                        se((i==1 ou i==8) e j==5)
                            então
                                mtz[i, j] = 5;
                        fimse;
                    fimse;
                fimse;
            fimse;
        fimse;
    fimpara;

    para(i=1;i<=8;i++)faça
        para(j=1;j<=8;j++)faça
            aux = 7 - mtz[i, j];
            s += aux;
        fimpara;
    fimpara;
```