

EXERCÍCIOS COM VETORES

Exercício 1

Carregar um vetor de entrada com N números inteiros. Em seguida gerar e mostrar dois vetores resultantes a partir do vetor de entrada: o primeiro deve conter os números positivos; o segundo deve conter os negativos.

Exercício 1

```
int vetor[100], positivos[100], negativos[100], k, n, i = 0, j = 0;
printf("Quantos numeros?: "); scanf("%i",&n);
for(k=0;k<n;k++){
    printf("vetor[%i]: ",k);
    scanf("%i",&vetor[k]);
}
for(k=0;k<n;k++){
    if (vetor[k] >= 0){
        positivos[i] = vetor[k];    i += 1;
    }
    else{
        negativos[j] = vetor[k];    j += 1;
    }
}
printf("\nPositivos: [ ");
for(k=0;k<i;k++) printf("%i ",positivos[k]);
printf("]\n");
printf("\nNegativos: [ ");
for(k=0;k<j;k++) printf("%i ",negativos[k]);
printf("]\n");
```

Exercício 2

Armazenar N valores reais em um vetor, fornecidos pelo usuário, e depois ler um número que corresponde a um índice no vetor e imprimir o valor correspondente.

Exercício 2

```
float num[100]; int n, indice;
printf("Quantos numeros?: "); scanf("%i", &n);
printf("\n");
for (int k=0; k<n; k++){
    printf("num[%i]: ", k); scanf("%f", &num[k]);
}
printf("\n");
indice = -1;
while (indice < 0 || indice > n-1){
    printf("Digite um indice: "); scanf("%i", &indice);
}
printf("\nValor = %5.2f\n", num[indice]);
```

Exercício 3

Armazenar N números inteiros fornecidos pelo usuário. Depois ler um número e, se for encontrado, imprimir o índice desse valor no vetor; caso contrário, imprimir uma mensagem.

Exercício 3

```
float num[100],numero;
int i=0,n;
printf("Quantos numeros?: "); scanf("%i",&n);
printf("\n");
for (int k=0; k<n; k++){
    printf("num[%i]: ",k); scanf("%f", &num[k]);
}
printf("\nProcurar o numero: ");  scanf("%f", &numero);
while (i < n && numero != num[i])  i++;
if (i==n)
    printf("\nO numero %5.2f NAO se encontra no vetor.\n", numero);
else
    printf("\n%5.2f encontra-se na posicao %d do vetor.\n", numero, i);
```

Exercício 4

Faça um programa que, no momento de carregar um vetor com N números inteiros, já o carregue de maneira ordenada crescente.

Exercício 4

```
int n, i, j, k, num;
printf("Quantos numeros?: "); scanf("%i",&n);
int vetor[n];
for(k=0; k<n; k++){
    printf("\nNumero: "); scanf("%i",&num);
    if (k == 0) // insere primeiro valor
        vetor[k] = num;
    else{ // insere demais valores
        i = 0;
        while (i < k && num > vetor[i])        i += 1;
        if (i < k){
            j = k;
            while (j > i){ // realiza deslocamento
                vetor[j] = vetor[j-1];
                j -= 1;
            }
        }
        vetor[i] = num; //insere elemento
    }
}
for(k=0; k<n; k++) //mostra vetor ordenado
    printf("\n%i",vetor[k]);
```

Exercício 5

Faça uma **função** que retorne ao programa principal os quatro primeiros números perfeitos. Sabe-se que um número é perfeito quando é igual à soma de seus divisores (exceto ele mesmo). Exemplo: os divisores de 6 são 1, 2 e 3 e $1+2+3 = 6$, logo, 6 é perfeito.

Exercício 5

Faça uma **função** que retorne ao programa principal os quatro primeiros números perfeitos. Sabe-se que um número é perfeito quando é igual à soma de seus divisores (exceto ele mesmo). Exemplo: os divisores de 6 são 1, 2 e 3 e $1+2+3 = 6$, logo, 6 é perfeito.

```
//Programa Principal
int n = 4, v[n];
perfeitos(n, v);
printf("\n\nOs %i primeiros Nos. perfeitos = ", n);
for(int k=0; k<n; k++) printf("%i ", v[k]);
```

Exercício 5

```
void perfeitos(int n, int v[]){  
    // ENCONTRA OS QUATRO PRIMEIROS PERFEITOS  
    int soma;  
    int num = 1; //números sequenciais  
    int k = 0; //quantidade de perfeitos  
    while (k < n){  
        soma = 0;  
        for (int divide=1; divide < num/2 + 1; divide++)  
            //procura divisores de num  
            if (num%divide == 0)  
                soma += divide; //soma os divisores  
        if (soma == num){  
            v[k] = num;  
            k += 1;  
            printf("\nMais um perfeito = %i", num);  
        }  
        num += 1;  
    }  
}
```

```
//Programa Principal  
int n = 4, v[n];  
perfeitos(n, v);  
printf("\n\nOs %i primeiros Nos. perfeitos = ", n);  
for(int k=0; k<n; k++) printf("%i ", v[k]);
```

Exercício 6

Colocar em ordem crescente um vetor contendo N números inteiros fornecidos pelo usuário.

NUM
7
9
5
3
1

Exercício 6

```
int n;
printf("Quantos numeros? "); scanf("%i",&n);
int k,troca, num[n];
for(k=0; k<n; k++){
    printf("\nnum[%d]: ",k); scanf("%i",&num[k]);
}
for(int i=0; i<n; i++)
    for(int j=0; j<n-i-1; j++)
        if (num[j] > num[j+1]){
            troca = num[j];
            num[j] = num[j+1];
            num[j+1] = troca;
        }
for(k=0; k<n; k++)
    printf("%i ", num[k]);
```

Exercício 7

Faça um programa que **carregue dois vetores** com N elementos numéricos cada um. Gere e mostre um terceiro vetor com $2*N$ posições, que será composto pela intercalação dos vetores anteriores. Depois ordene o vetor resultante de maneira crescente e mostre-o.

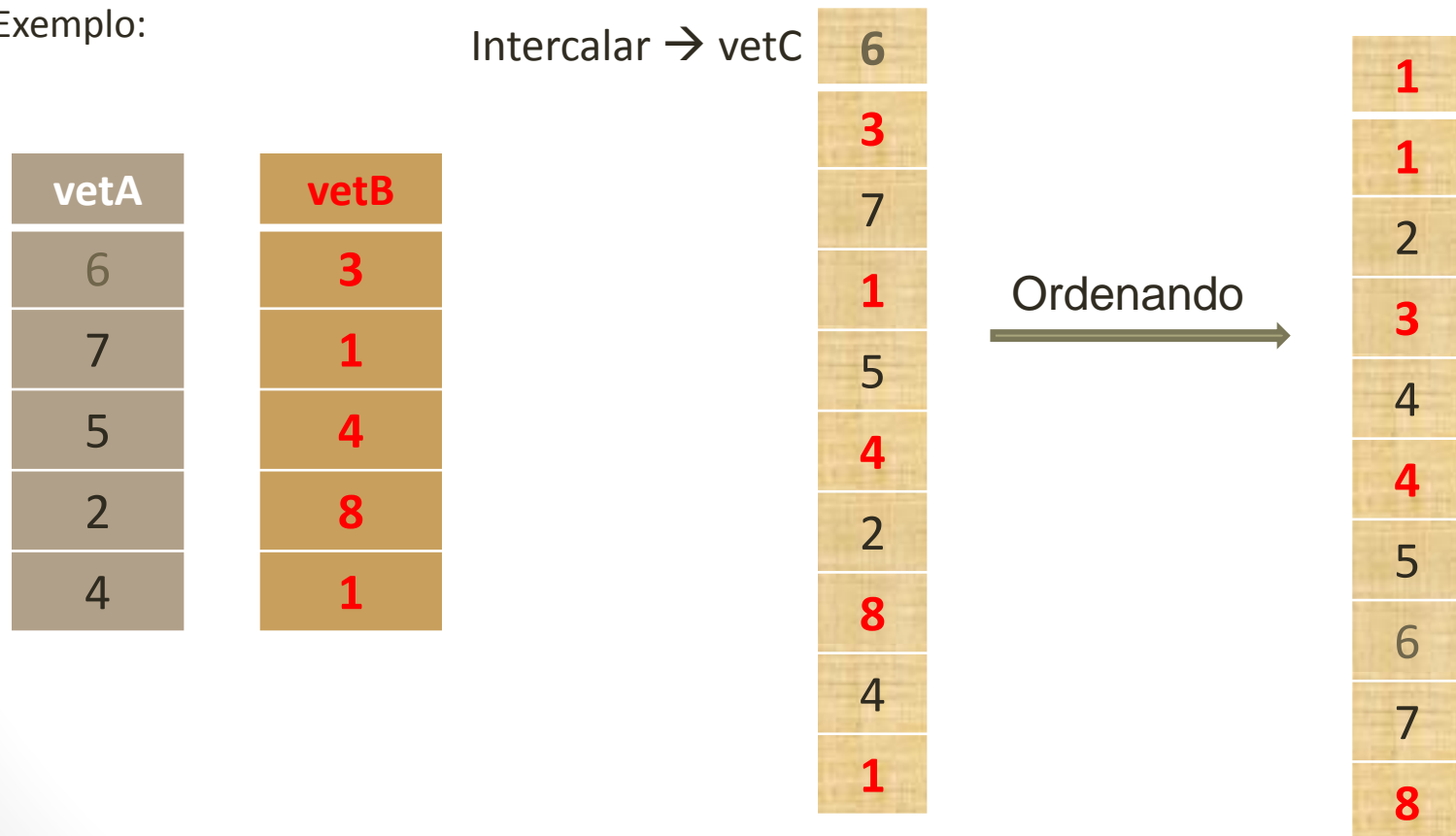
Exemplo:

vetA		vetB
6		3
7		1
5		4
2		8
4		1

Exercício 7

Faça um programa que carregue dois vetores com N números inteiros cada um. Gere e mostre um terceiro vetor com $2*N$ posições, que será composto pela **intercalação** dos vetores anteriores. Depois ordene o vetor resultante de maneira crescente e mostre-o.

Exemplo:



Exercício 7

```
int i,j,k,n,base,aux;
printf("Quantos numeros em cada vetor? ");  scanf("%i",&n);
int vetA[n], vetB[n], tamC = 2*n, vetC[tamC];
printf("\n");
// carregue dois vetores
for(k=0; k<n; k++){
    printf("vetA[%d]: ",k);  scanf("%i", &vetA[k]);
}
printf("\n");
for(k=0; k<n; k++){
    printf("vetB[%d]: ",k);  scanf("%i", &vetB[k]);
}
// intercalar

// ordenar
```

Exercício 7

```
int i,j,k,n,base,aux;
printf("Quantos numeros em cada vetor? "); scanf("%i",&n);
int vetA[n], vetB[n], tamC = 2*n, vetC[tamC];
printf("\n");
// carregue dois vetores
for(k=0; k<n; k++){
    printf("vetA[%d]: ",k); scanf("%i", &vetA[k]);
}
printf("\n");
for(k=0; k<n; k++){
    printf("vetB[%d]: ",k); scanf("%i", &vetB[k]);
}
// intercalar

// ordenar
```

// intercalar

i = -1;

for(k=0; k<n; k++){

i += 1; vetC[i] = vetA[k];

i += 1; vetC[i] = vetB[k];

}

printf("\nvetC intercalado: ");

for(k=0; k<tamC; k++)

printf("%i ",vetC[k]);

printf("\n");

// ordenar

Exercício 7

```
int i,j,k,n,base,aux;
printf("Quantos numeros em cada vetor? "); scanf("%i",&n);
int vetA[n], vetB[n], tamC = 2*n, vetC[tamC];
printf("\n");
// carregue dois vetores
for(k=0; k<n; k++){
    printf("vetA[%d]: ",k); scanf("%i", &vetA[k]);
}
printf("\n");
for(k=0; k<n; k++){
    printf("vetB[%d]: ",k); scanf("%i", &vetB[k]);
}
// intercalar
i = -1;
for(k=0; k<n; k++){
    i += 1; vetC[i] = vetA[k];
    i += 1; vetC[i] = vetB[k];
}
printf("\nvetC intercalado: ");
for(k=0; k<tamC; k++)
    printf("%i ",vetC[k]);
printf("\n");
```

```
// ordenar
base = tamC - 1;
for(i=0; i<tamC - 1; i++){
    j = 0;
    while (j < base){
        if (vetC[j] > vetC[j+1]){
            aux = vetC[j];
            vetC[j] = vetC[j+1];
            vetC[j+1] = aux;
        }
        j += 1;
    }
    base -= 1;
}
```

Exercício 7

```
int i,j,k,n,base,aux;
printf("Quantos numeros em cada vetor? "); scanf("%i",&n);
int vetA[n], vetB[n], tamC = 2*n, vetC[tamC];
printf("\n");
// carregue dois vetores
for(k=0; k<n; k++){
    printf("vetA[%d]: ",k); scanf("%i", &vetA[k]);
}
printf("\n");
for(k=0; k<n; k++){
    printf("vetB[%d]: ",k); scanf("%i", &vetB[k]);
}
// intercalar
i = -1;
for(k=0; k<n; k++){
    i += 1; vetC[i] = vetA[k];
    i += 1; vetC[i] = vetB[k];
}
printf("\nvetC intercalado: ");
for(k=0; k<tamC; k++)
    printf("%i ",vetC[k]);
printf("\n");
```

```
// ordenar
base = tamC - 1;
for(i=0; i<tamC - 1; i++){
    j = 0;
    while (j < base){
        if (vetC[j] > vetC[j+1]){
            aux = vetC[j];
            vetC[j] = vetC[j+1];
            vetC[j+1] = aux;
        }
        j += 1;
    }
    base -= 1;
}
printf("\nvetC ordenado: ");
for (k=0;k<tamC;k++)
    printf("%i ",vetC[k]);
```