

PESQUISA 2 – 02/06

As orientações de vetores são formas de organizar os vetores para que seja mais fácil compreender e visualizar o resultado que é apresentado ao final do código. No geral ele ordena o valor dos vetores, através de 3 métodos principais: seleção, inserção e bolha.

- **Seleção:** utiliza o *selection sort* ordena o vetor de forma crescente, seleciona o menor número e troca pelo que está na 1ª posição, e segue realizando essa ação até o vetor inteiro esteja em ordem, segundo a fórmula de $n-1$ e $n-2$ para realização do código.

```
#include <stdio.h>
#include <stdlib.h>
void selectionSort(int v[200], int n){
    int i, j, aux, min;
    for(i = 0; i < n-1; i++) {
        min = i;
        for(j = i+1; j < n; j++) {
            if(v[j] < v[min]) {
                min = j;}
        }
        aux = v[i]; v[i] = v[min]; v[min] = aux; //troca
    }
}

int main(){
    int v[200], n, i;
    printf("Entre tamanho desejado do vetor: ");
    scanf("%d", &n);
    printf("Entre os %d elementos do vetor:\n", n);
    for(i = 0; i < n; i++) {
        scanf("%d", &v[i]);
    }
    selectionSort(v, n);
    printf("\n\nVetor ordenado:\n");
    for(i = 0; i < n; i++) {
        printf("%d\t", v[i]);
    }
    printf("\n");
    return 0;
}
```

- **Inserção:** utiliza *insertion sort* para ordenar o vetor em forma crescente, mas percorrendo o vetor da esquerda para a direita, “empurrando” os valores maiores para o lado até que todos estejam em ordem, quase que dividindo os elementos em já ordenados e não ordenados.

```

#include <stdio.h>
#include <stdlib.h>
void insertionSort(int v[200], int n){
    int i, j, x;
    for(i = 1; i < n; i++) {
        x = v[i];
        j = i - 1;
        while(j >= 0 && v[j] > x) {
            v[j+1] = v[j];
            j--;
        }
        v[j+1] = x;
    }
}

int main(){
    int v[200], n, i;
    printf("Entre tamanho desejado do vetor: ");
    scanf("%d", &n);
    printf("Entre os %d elementos do vetor:\n", n);
    for(i = 0; i < n; i++) {
        scanf("%d", &v[i]);
    }
    insertionSort(v, n);
    printf("\n\nVetor ordenado:\n");
    for(i = 0; i < n; i++) {
        printf("%d\t", v[i]);
    }
    printf("\n");
    return 0;
}

```

- **Bolha:** utiliza o *bubble sort* e realizar a ordenação do vetor através da comparação dos elementos, se estiverem fora de ordem os elementos são trocados de posição, como se os números “flutuassem” de uma posição para a outra.

```

#include <stdio.h>
#include <stdlib.h>
void bubbleSort(int v[200], int n){
    int i, j, aux;
    for(i = n-1; i > 0; i--) {
        for(j = 0; j < i; j++) {
            if(v[j] > v[j+1]) {aux = v[j]; v[j] = v[j+1]; v[j+1] = aux; //troca}
        }
    }
}

```

```
int main(){
int v[200], n, i;
printf("Entre tamanho desejado do vetor: ");
scanf("%d", &n);
printf("Entre os %d elementos do vetor:\n", n);
for(i = 0; i < n; i++) {
    scanf("%d", &v[i]);
}
bubbleSort(v, n);
printf("\n\nVetor ordenado:\n");
for(i = 0; i < n; i++) {
    printf("%d\t", v[i]);
}
printf("\n");
return 0;
}
```

Bibliografia

<https://www.treinaweb.com.br/blog/conheca-os-principais-algoritmos-de-ordenacao/>

http://www.facom.ufu.br/~anilton/EQQ09_PD_EngQuimica/ordenacao.pdf

http://ww2.inf.ufg.br/~hebert/disc/aed1/AED1_04_ordenacao1.pdf

<https://www.devmedia.com.br/algoritmos-de-ordenacao-analise-e-comparacao/28261>

https://paginas.fe.up.pt/~lpreis/prog2_06_07/AulasTeoricas/4.vectorOrd.pdf