

```
#include <iostream>
//VINICIUS MESQUINI 0093/19
```

```
using namespace std;
```

```
void Troca(float* vetor, int i, int j){//troca os valores
```

```
    int aux = vetor[i];
    vetor[i] = vetor[j];
    vetor[j] = aux;
}
```

```
int Pivo(float* vetor, int inicial, int final) { //separa o vetor e marca o pivo
```

```
    int i = inicial + 1; //inicial 0+1=1
    int j = final; //final, tamanho do vetor
    int pivo = vetor[inicial]; //pivo vai ser a posição 1 do vetor
    while (i <= j) { // repetição enquanto o inicio e fim do vetor nao se encontnam
        if (vetor[i] <= pivo) {
            i++; // se a posição do vetor em i é menor que o o pivo,
                // avança o mais uma casa da esquerda para direita
        }
        else if (vetor[j] > pivo) {
            j--; // se nao , se o vetor na posição do j(fim do vetor) for maior
                // que o pivo, diminui uma casa da direita para esquerda no vetor
        }
        else if (i <= j) {
            Troca(vetor, i, j); // troca os valores e aproxima as extremidades
            i++;
            j--;
        }
    }
    Troca(vetor, inicial, j); // coloca o pivo no lugar certo
    return j;
}
```

```
void Quick(float* vetor, int inicial, int final) { // recursividade se o vetor nao for de 0 ou 1 posição
```

```
    if (inicial < final) {
        int j = Pivo(vetor, inicial, final);
        Quick(vetor, inicial, j - 1);
        Quick(vetor, j+1, final);
    }
}
```

```
}
```

```
int main() {
```

```
    float vetor[1000];
    int tam;
```

```
    cout << "Digite a quantidade de posicoes: ";
    cin >> tam;
    for (int i = 0; i < 1000; i++) {
        if (i < tam) {
            cout << "informe um valor: ";
            cin >> vetor[i];
        }
        else {
            break;
        }
    }
}
```

```
    cout << "\n\nvetor dessarrumado: ";
    for (int i = 0; i < tam; i++) {
        cout << vetor[i] << " ";
    }
    cout << "\n\nvetor arrumado: ";
    //shell(vetor, tam);
    Quick(vetor, 0, tam-1 );
    for (int i = 0; i < tam; i++) {
        cout << vetor[i] << " ";
    }
}
```

} }