Primeiramente é criado um vetor X, de cinco posições;

O vetor X é povoado com dados passados pelo usuário;

Logo após, é chamada a função **quicksort()**, passando como parâmetros o vetor, sua posição inicial e final;

Na função quicksort(), é criada uma variável do tipo inteiro **q**. Logo após é criado um If com a condição de a posição inicial ser menor que a final. Dentro desse if, a variável inteira **q** recebe o valor retornado da função **particao()**, e acontecem duas chamadas de recursão da própria função quicksort().

Na função particao(), que recebe como parâmetros o vetor X, sua posição inicial e final, são criadas as variáveis pivô, i e j;

O pivô, recebe o valor que está no meio do vetor, as variáveis i e j recebem o valor inicial-1 e final+1, respectivamente; logo após isso temos um **while(i<j)**, e dentro dele a variável j recebe j-1 enquanto X[j]>pivô e a variável i recebe i+1 enquanto X[i]<pivô. Ainda dentro do while temos um if (i<j), que chama a função **troca(X, i, j)**; feito o while, a função particao() retorna o valor de j;

A função troca(), faz a ordenação dos valores passados para ela como parâmetro dentro do vetor;

Graças às chamadas de recursão das funções quicksort() e particao(), o procedimento se repete até que todo o vetor tenha sido separado e ordenado de acordo com o pivô;

Após o vetor ter sido ordenado, voltamos ao main(), onde um for com um cout mostra os valores do vetor na tela, e para finalizar a execução, um getch().