```
/* Q3. Inversion count for array indicates how far (or close) the array is from being sorted.
  If the array is already sorted, then the inversion count is 0; but if the array is sorted in reverse order,
  the inversion count is the maximum. Given an array arr[]. The task is to find the inversion count of
arr[].
  Two elements arr[i] and arr[j] form an inversion if a[i] > a[j] and i<j.
*/


#include <iostream>
using namespace std;

// Function to get inversion count
int inversionCount(int arr[], int n)
{
  int inverCount = 0;
  for(int i=n-1; i>0; --i)
  {
    for(int j=0; j<i; ++j)
    {
      if(arr[j]>arr[i]){
        ++inverCount;
        // cout<<endl<<inverCount<<". "<<arr[i]<<" "<<arr[j];
      }
    }
  }
  return inverCount;
}

int main()
{
  int arr[] = {5,20,18,20,5,23,-8,100};
  // int arr[] = {-8,5,5,18,20,20,23,100};

  cout<<"Given array: ";
  for(int i=0; i<8; ++i)
    cout<<arr[i]<<" ";
  cout<<"\nInversion Count: "<<inversionCount(arr, 8);
  return 0;
}
```