

Create an implementation of quicksort based on the following split program.

```
int Count=0;

int Split(int * List,int Size)
{
    Count += Size;
    int rv = 0;
    int Temp;
    for (int i = 1 ; i<Size ; i++)
    {
        if (List[i]<List[0])
        {
            rv++;
            Temp = List[i];
            List[i] = List[rv];
            List[rv] = Temp;
        }
    }
    Temp = List[0];
    List[0] = List[rv];
    List[rv] = Temp;
    return rv;
}

void QuickSort(int * List, int Size)
{
    if (Size>2)
    {
        int Pivot = Split(List,Size);
        QuickSort(List,Pivot);
        QuickSort(List+Pivot+1,Size-Pivot-1);
    }
}
```

In your main routine, declare an array of 100 elements, and sort the array twice using the above QuickSort algorithm.

Initialize your array using random numbers, using the rand function. The following line uses rand to initialize x. You will need to use a for loop.

```
int x = rand( );
```

Initialize Count to zero and run QuickSort. Then print

```
cout<< "Count="<<Count<<end;
```

Next, initialize your array to the integers 0-99 in sequence like this:

```
for (long i = 0 ; i<100 ; i++)  
{  
    YourArray[i] = i;  
}
```

Then set "Count" to zero and run QuickSort again.

Then print again

```
cout<< "Count=" << Count;
```

What does this say about QuickSort?