Quick Sort Implementation

#include<process.h>

#include<iostream.h>

#include<conio.h>

#include<stdlib.h>

int Partition(int low,int high,int arr[]);

void Quick\_sort(int low,int high,int arr[]);

void main()

{

int \*a,n,low,high,i;

clrscr();

cout<<"/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Quick Sort Algorithm Implementation\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/";

cout<<"Enter number of elements:";

cin>>n;

a=new int[n];

/\* cout<<"enter the elements:";

for(i=0;i<n;i++)

cin>>a;\*/

for(i=0;i<n;i++)

a[i]=rand()%100;

clrscr();

cout<<"Initial Order of elements";

for(i=0;i<n;i++)

cout<<a[i]<<" ";

cout<<"";

high=n-1;

low=0;

Quick\_sort(low,high,a);

cout<<"

Final Array After Sorting:";

for(i=0;i<n;i++)

cout<<a[i]<<" ";

getch();

}

/\*Function for partitioning the array\*/

int Partition(int low,int high,int arr[])

{ int i,high\_vac,low\_vac,pivot/\*,itr\*/;

pivot=arr[low];

while(high>low)

{ high\_vac=arr[high];

while(pivot<high\_vac)

{

if(high<=low) break;

high--;

high\_vac=arr[high];

}

arr[low]=high\_vac;

low\_vac=arr[low];

while(pivot>low\_vac)

{

if(high<=low) break;

low++;

low\_vac=arr[low];

}

arr[high]=low\_vac;

}

arr[low]=pivot;

return low;

}

void Quick\_sort(int low,int high,int arr[])

{

int Piv\_index,i;

if(low<high)

{

Piv\_index=Partition(low,high,arr);

Quick\_sort(low,Piv\_index-1,arr);

Quick\_sort(Piv\_index+1,high,arr);

}

}