**ASSIGNMENT-3.4 (QUICK SORT)**

#include<stdio.h>

int Partition(int a[], int l, int h)

{

int pivot = a[l];

int start = l;

int end = h;

int temp\_1, temp\_2;

while(start<end)

{

while(a[start]<=pivot)

{

start++;

}

while(a[end]>pivot)

{

end--;

}

if(start<end)

{

temp\_1=a[start];

a[start]=a[end];

a[end]=temp\_1;

}

}

temp\_2=a[l];

a[l]=a[end];

a[end]=temp\_2;

return end;

}

void quick\_sort(int a[], int l, int h)

{

int loc;

if(l<h)

{

loc=Partition(a,l,h);

quick\_sort(a,l,loc-1);

quick\_sort(a,loc+1,h);

}

}

int main()

{

int a[10],size,i;

printf("Enter the size of array-");

scanf("%d",&size);

if(size>10)

{

printf("Array overflows");

}

else

{

printf("The the size of array-%d",size);

printf("\nEnter the elements of the array-");

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

{

scanf("%d",&a[i]);

}

printf("\nThe elements of the array\n");

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

{

printf("[%d]",a[i]);

}

quick\_sort(a,0,size-1);

printf("\nSorted array\n");

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

{

printf("[%d]",a[i]);

}

}

return 0;

}