**2) Sort a given set of N integer elements using the Quick Sort technique and compute its time taken**

#include<stdio.h>

void quickSort(int[], int, int);

int partition(int[], int, int);

void swap(int\*, int\*);

int main()

{

int n,i;

printf("Enter Array Size\n");

scanf("%d",&n);

int arr[n];

printf("Enter Array Elements\n");

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

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

quickSort(arr,0,n-1);

printf("After the QuickSort\n");

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

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

printf("\n");

return 0;

}

void quickSort(int arr[], int start, int end)

{

if(start < end)

{

int pIndex = partition(arr, start, end);

quickSort(arr, start, pIndex-1);

quickSort(arr, pIndex+1, end);

}

}

int partition(int arr[], int start, int end)

{

int pIndex = start;

int pivot = arr[end];

int i;

for(i = start; i < end; i++)

{

if(arr[i] < pivot)

{

swap(&arr[i], &arr[pIndex]);

pIndex++;

}

}

swap(&arr[end], &arr[pIndex]);

return pIndex;

}

void swap(int \*x, int \*y)

{

int t = \*x;

\*x = \*y;

\*y = t;

}