**// Program 1: Write and execute a program to sort a given set of elements using the Quick sort method and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted and plot a graph of the time taken versus n. The elements should be read from a file/can be generated using the random number generator.**

#include<stdio.h>

// Partition Function

int partition(int a[100], int low, int high) {

int key, i, j, temp;

key = a[low];

i = low + 1;

j = high;

while(1) {

while (key >= a[i])

i++;

while (key < a[j])

j--;

if(i < j) {

temp = a[i];

a[i] = a[j];

a[j] = temp;

}

else {

temp = a[j];

a[j] = a[low];

a[low] = temp;

return j;

}

}

}

// Quicksort Function

void quicksort(int a[100], int low, int high) {

int j;

if(low < high) {

j = partition(a, low, high);

quicksort(a, low, j - 1);

quicksort(a, j + 1, high);

}

}

// Main Program

int main() {

int n, a[100], i;

printf("Enter the number of elements:\n");

scanf("%d",&n);

printf("Enter the %d elements:\n", n);

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

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

quicksort(a, 0, n - 1);

printf("The sorted elements are:\n");

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

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

}

**/\***

**C:\TURBOC3\BIN>TC**

**enter the number of elements**

**10**

**346 130 982 90 656 117 595 415 948 126**

**the sorted array is**

**90**

**117**

**126**

**130**

**346**

**656**

**415**

**595**

**948**

**982**

**time taken is 0.000000**

**\*/**