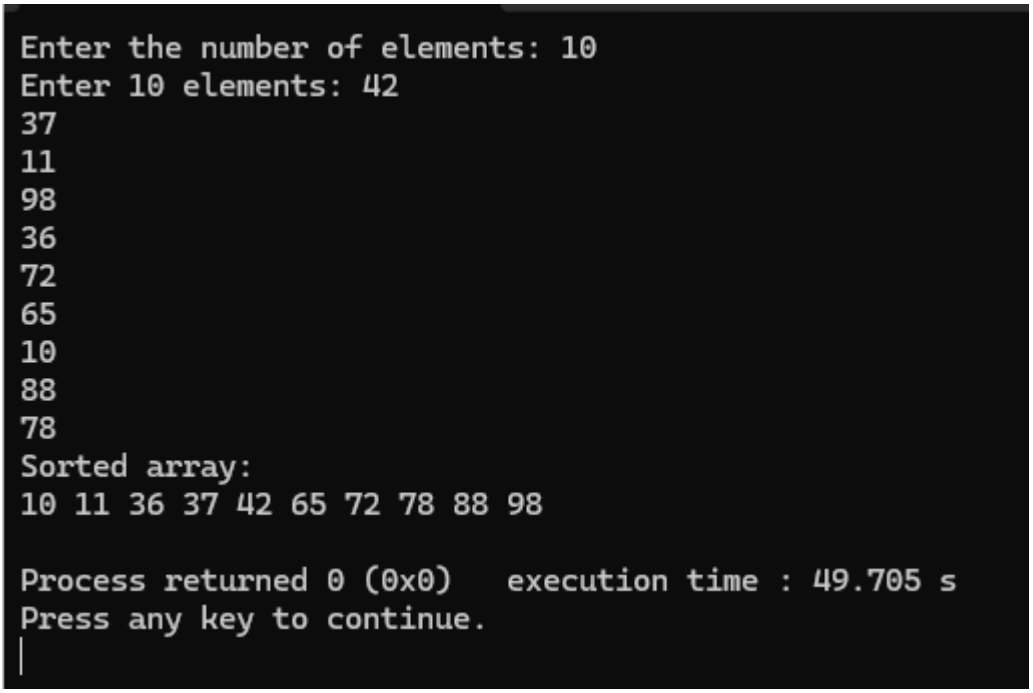
LAB 6- Quick Sort

Sort a given set of N integer elements using quick sort technique.

Code:

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
void swap(int *a, int *b) {
int t = *a;
*a = *b;
*b = t;
}
int partition(int arr[], int low, int high) {
int pivot = arr[high];
int i = (low - 1);
for (int j = low; j <= high - 1; j++) {
if (arr[j] < pivot) {
i++;
swap(&arr[i], &arr[j]);
}
}
swap(&arr[i + 1], &arr[high]);
return (i + 1);
}
void quickSort(int arr[], int low, int high) {
if (low < high) {
int pi = partition(arr, low, high);
quickSort(arr, low, pi - 1);
quickSort(arr, pi + 1, high);
}
}
void printArray(int arr[], int size) {
for (int i = 0; i < size; i++)
printf("%d ", arr[i]);
printf("\n");
}
int main() {
int n;
printf("Enter the number of elements: ");
scanf("%d", &n);
int arr[n];
printf("Enter %d elements: ", n);
for (int i = 0; i < n; i++) {
scanf("%d", &arr[i]);
}
quickSort(arr, 0, n - 1);
printf("Sorted array: \n");
printArray(arr, n);
return 0;
}
```

Output:


Enter the number of elements: 10
Enter 10 elements: 42
37
11
98
36
72
65
10
88
78
Sorted array:
10 11 36 37 42 65 72 78 88 98
Process returned 0 (0x0)   execution time : 49.705 s
Press any key to continue.