Quick Sort

```c
#include <stdio.h>
#include<time.h>
#include<stdlib.h>

int n;
long a[100000];

void swap(long* a, long* b) {
    long t = *a;
    *a = *b;
    *b = t;
}

long partition(long arr[], long low, long high) {
    long 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(long arr[], long low, long high) {
    if (low < high) {
        long pi = partition(arr, low, high);

        quickSort(arr, low, pi - 1);
        quickSort(arr, pi + 1, high);
    }
}
```

```c
void printArray() {
    printf("\nSorted Array\n");
    for (int i = 0; i < n; i++) {
        printf("%d ", a[i]);
    }
    printf("\n");
}

int main() {
    clock_t start, end;
    printf("Enter the size of the array to be sorted: ");
    scanf("%d",&n);
    printf("\nEnter the array elements: ");
    srand(time(NULL));
    for(int i = 0 ; i < n ; i++){
        a[i] = rand() % 100000;
        // scanf("%d",&a[i]);
    }
    start = clock();
    quickSort(a, 0, n - 1);
    end = clock();
    printArray();
    double time_taken = ((double) (end-start) )/CLOCKS_PER_SEC;
    printf("\nTime taken to sort: %lf",time_taken);
    return 0;
}
```