LAB - TEST - 1

NAME :- Tushar . A . Pai

USN :- 1BM19CS174

SUBJ :- [redacted] ADA

Sign :- Tushar

Course code :- 19CS 4PC ADA

Quick sort implementation.

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

  int arr[1000000];

void swap (int arr[], int index1, int index2) {
   int temp = arr[index1];
   arr[index1] = arr[index2];
   arr[index2] = temp;
}.
```

```
int partition( int arr[], int start , int end) {
    int pivot = arr[end];
    int boundary = start - 1;
    for (int i = start ; i <= end ; i++) {
        if ( arr [i] <= pivot) {
            boundary ++;
            Swap (arr, i, boundary);
        }
    }
    return boundary;
}

void quicksort ( int arr[] , int start , int end) {
    for(int i=0; i< 800; i++)
    { for (int i=0 ; i< 400; i++) {
            // loop to increase delay.
        }
    }
    if ( start >= end)
        return ;
    int boundary = partition (arr, start , end );
    quicksort( arr, start, boundary -1);
    quicksort ( arr, boundary +1 , end );
}
```


Tushar

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

int main()
{  time_t start, end;
   int n;
   srand(time(0));
   printf("Enter no. of elements \n");
   scanf("%d", &n);
   for(int i=0; i<n; i++){
     arr[i] = rand();
   }
   start = time(NULL);
     quicksort(arr, 0, n-1);
     end = time(NULL);
printf("Array is sorted \n");
printf("The time taken is %0.10f \n", difftime(end,start)/
                                        CLOCKS_PER_SEC);
 return 0;
}
```


Tushar

Modification

(a) To find the k^{th} minimum element.

Algo

Take the value of k from the user.

then printf ("k th smallest element is %d", arr[k-1]);

* Here we are assuming that there are no duplicate values.

Code

```
int k_th_smallest (int k) {
    return arr[k-1];
}
```

Call this after sorting the array.

Tushar