United International University (UIU)

Dept. of Computer Science & Engineering (CSE)

Course Code: **CSI 228**

Course Title: **Algorithms Lab**

1. Implement the Quick sort algorithm. Write a main function to test your code.

Hint:

The quick sort uses divide and conquer just like merge sort but without using additional storage.

The steps are:

1. Select an element q, called a ***pivot***, from the array. In this algorithm, we have chosen the last index as the pivot.
2. The **PARTITION** function finds the location of the pivot in such a way that all the elements smaller than the pivot is on the left side and all the elements greater than the pivot is on the right-hand side. (Items with equal values can go either way).
3. Recursively call the **QUICKSORT** function which performs quicksort on the array on the left side of the pivot and then on the array on the right side, thus, dividing the task into sub-tasks. This is carried out until the arrays can no longer be split.

The pseudo code of quick sort is given below:

```
QUICKSORT(A, p, r)
1  if p < r
2      q = PARTITION(A, p, r)
3      QUICKSORT(A, p, q − 1)
4      QUICKSORT(A, q + 1, r)
```

```
PARTITION(A, p, r)
1  x = A[r]
2  i = p − 1
3  for j = p to r − 1
4      if A[j] ≤ x
5          i = i + 1
6          exchange A[i] with A[j]
7  exchange A[i + 1] with A[r]
8  return i + 1
```

- p = low
- q = position of pivot after partitioning
- r = high

the way we used in binary search and merge sort

Sample Input: int A [] = {10, 15, 5, 19, 16, 188, 99, 14, 27, 35};

2. You are given a sorted array where all elements are unique, except one. For example, {1, 2, 5, 6, 6, 7}. Find the element that is not unique (6) **using divide and conquer**.

[**Hint:** modify the binary search slightly.]
The pseudo code of binary search is given below.

```
Boolean BS(A, key, start, end)
      mid = (start+end)/2
      if(A[mid] == key)
            return true
      else
            if(end <= start)
                  return false
            else
                  if (A[mid] > key)
                        return BS(A, key, start, mid-1)
                  else
                        return BS(A, key, mid+1, end)
```

Figure 1: Binary Search Pseudocode

Sample Input	**Sample Output**
{1, **2, 2, 2, 2**, 30}	2
{1, 2, **25**, **25**, 27, 30}	25
{21, 41, 52, 62, **72, 72**, **72,** 300}	72