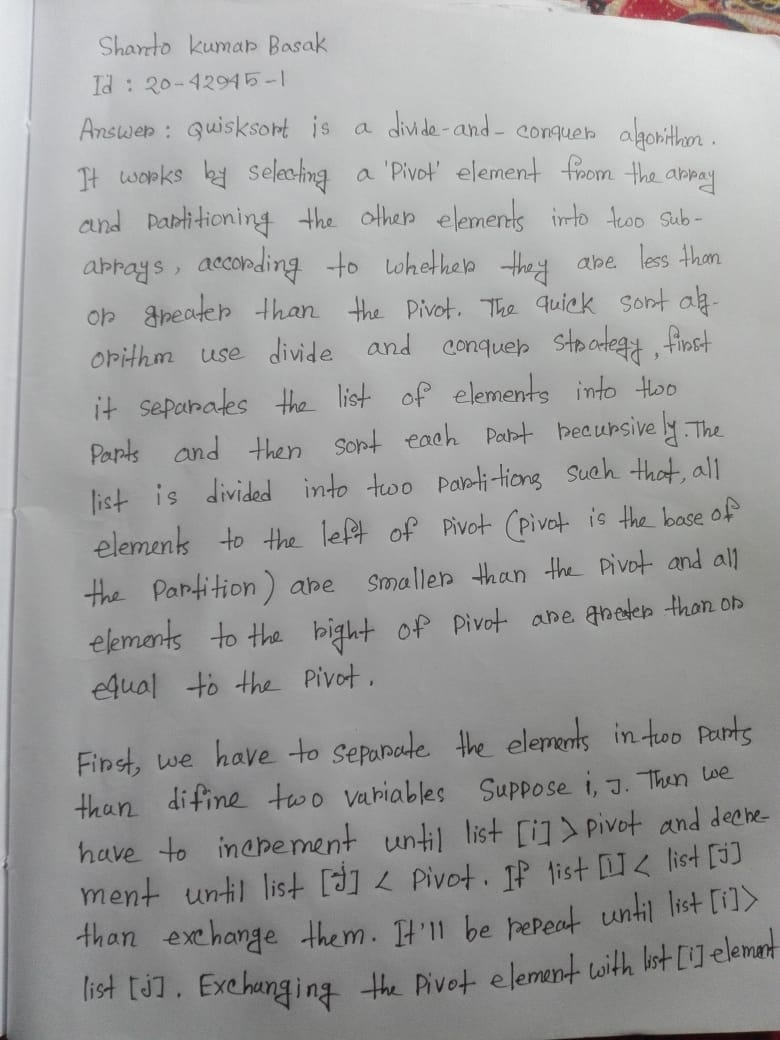
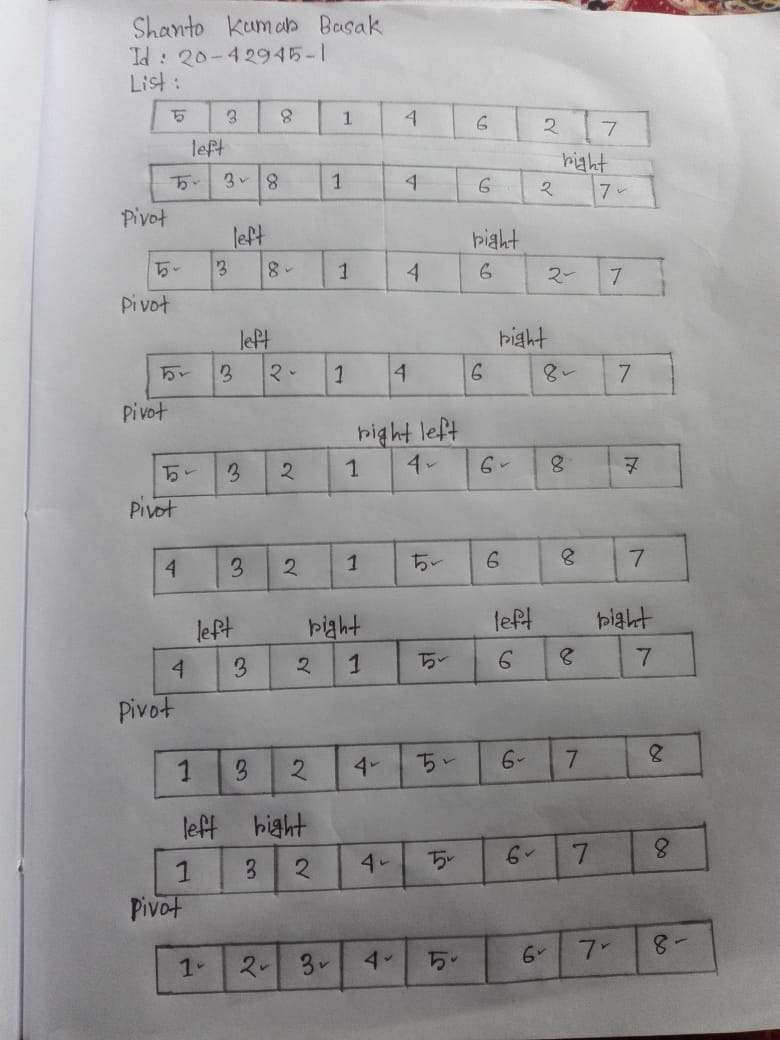
**Algorithms**

**Mid Assignment**

**1. Explain how Quick sort sorts**

****

**2. Use diagrams to provide a simulation for Quick Sort.**



**3. Write a pseudo code for Quick Sort**

**Answer :**

**Quick Sort Pseudocode:**

Partition(int A,int low, int high)

   i=low+1

   j=high

 Pivot=low;

i<j:

A[i]<=pivot->i++

A[j]>pivot->j—

Else

Swap with pivot;

Pivot=A[j]

Quicksort (int A, int low, int high):

Low<high:

       partition=Partition(A, low, high)

      Quicksort (A, low, partition-1)

         Quicksort (A, partition+1, high)

END.

**4. Implement and add the code In this section**

**Answer :**

#include <iostream>

using namespace std;

void swap(int\* a, int\* b)

{

int t = \*a;

\*a = \*b;

\*b = t;

}

int partition (int arr[], int left, int right)

{

int pivot = arr[right];

int i = (left - 1);

for (int j = left; j <= right - 1; j++)

{

if (arr[j] < pivot)

{

i++;

swap(&arr[i], &arr[j]);

}

}

swap(&arr[i + 1], &arr[right]);

return (i + 1);

}

void quickSort(int arr[], int left, int right)

{

if (left < right)

{

int p = partition(arr, left, right);

quickSort(arr, left, p - 1);

quickSort(arr, p + 1, right);

}

}

void printArray(int arr[], int size)

{

int i;

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

cout << arr[i] << " ";

cout << endl;

}

int main()

{

int arr[] = {11, 8, 3, 22, 4, 5};

int n = sizeof(arr) / sizeof(arr[0]);

quickSort(arr, 0, n - 1);

cout << "Sorted array: \n";

printArray(arr, n);

return 0;

}