**Agnim Gupta**

**2028083**

**A-23,CSSE**

**Question 1**

#include <iostream>

using namespace std;

void heap(int arr[], int n, int i)

{

    int largest = i;

    int l = 2 \* i + 1;

    int r = 2 \* i + 2;

    if (l < n && arr[l] > arr[largest])

        largest = l;

    if (r < n && arr[r] > arr[largest])

        largest = r;

    if (largest != i) {

        swap(arr[i], arr[largest]);

        heap(arr, n, largest);

    }

}

void heapSort(int arr[], int n)

{

    for (int i = n / 2 - 1; i >= 0; i--)

        heap(arr, n, i);

    for (int i = n - 1; i > 0; i--) {

        swap(arr[0], arr[i]);

        heap(arr, i, 0);

    }

}

void printArray(int arr[], int n)

{

    for (int i = 0; i < n; ++i)

        cout << arr[i] << " ";

    cout << "\n";

}

int main()

{

    int arr[] = { 66,43,120,100,55,89 };

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

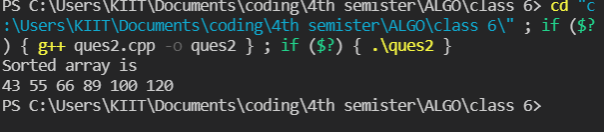
    heapSort(arr, n);

    cout << "Sorted array is \n";

    printArray(arr, n);

}

**Output**

****

**Question 2**

//Priority Queue

#include <bits/stdc++.h>

using namespace std;

struct item {

    int value;

    int priority;

};

item pr[100000];

int size = -1;

void enqueue(int value, int priority)

{

    size++;

    pr[size].value = value;

    pr[size].priority = priority;

}

int peek()

{

    int highestPriority = INT\_MIN;

    int ind = -1;

    for (int i = 0; i <= size; i++) {

        if (highestPriority == pr[i].priority && ind > -1

            && pr[ind].value < pr[i].value) {

            highestPriority = pr[i].priority;

            ind = i;

        }

        else if (highestPriority < pr[i].priority) {

            highestPriority = pr[i].priority;

            ind = i;

        }

    }

    return ind;

}

void dequeue()

{

    int ind = peek();

    for (int i = ind; i < size; i++) {

        pr[i] = pr[i + 1];

    }

    size--;

}

int main()

{

    enqueue(10, 2);

    enqueue(14, 4);

    enqueue(16, 4);

    enqueue(12, 3);

    int ind = peek();

    cout << pr[ind].value << endl;

    dequeue();

    ind = peek();

    cout << pr[ind].value << endl;

    dequeue();

    ind = peek();

    cout << pr[ind].value << endl;

    return 0;

}

**Output**

