

**Data Structure and Algorithms**

**ASSIGNMENT NO 09**

**SUBMITTED BY:**

Hasaan Ahmad SP22-BSE-017

**SUBMITTED TO: Sir Syed Ahmad Qasim**

Activities:

// Design an Array Based Priority Queue. Write the enqueue method.

#include <iostream>

using namespace std;

int Heap[1000], sz = -1;

int parent(int i)

{

    return (i - 1) / 2;

}

void shiftUp(int i)

{

    if (i == 0)

        return;

    if (Heap[parent(i)] < Heap[i])

    {

        swap(Heap[parent(i)], Heap[i]);

        shiftUp(parent(i));

    }

}

void insert(int k)

{

    Heap[++sz] = k;

    shiftUp(sz);

}

int shiftDown(int i)

{

    int left = 2 \* i + 1;

    int right = 2 \* i + 2;

    int largest = i;

    if (left <= sz && Heap[left] > Heap[largest])

        largest = left;

    if (right <= sz && Heap[right] > Heap[largest])

        largest = right;

    if (largest != i)

    {

        swap(Heap[i], Heap[largest]);

        shiftDown(largest);

    }

    return largest;

}

int leftChild(int i)

{

    return 2 \* i + 1;

}

int rightChild(int i)

{

    return 2 \* i + 2;

}

int extractMax()

{

    int ret = Heap[0];

    Heap[0] = Heap[sz--];

    shiftDown(0);

    return ret;

}

int main()

{

    cout << "Enter the number of elements: ";

    int n;

    cin >> n;

    for (int i = 0; i < n; i++)

    {

        int x;

        cin >> x;

        insert(x);

    }

    for (int i = 0; i <= sz; i++)

        cout << Heap[i] << " ";

    cout << endl;

    cout << "Enter the number of elements to extract: ";

    int k;

    cin >> k;

    for (int i = 0; i < k; i++)

        cout << extractMax() << " ";

    cout << endl;

    for (int i = 0; i <= sz; i++)

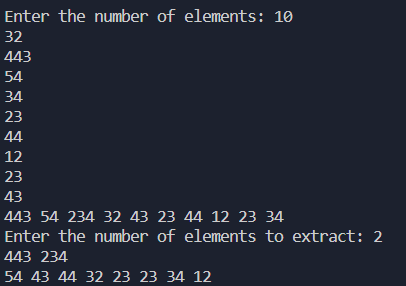
        cout << Heap[i] << " ";

    cout << endl;

    return 0;

}

**Output:**



**Graded Lab Task:**

/\*

Design a Hospital Registration system for patients using Priority Queue.

a. Assume a struct that you think is appropriate for such a system.

b. Take thee implemented system of Priority Queue

\*/

#include <iostream>

using namespace std;

struct patient

{

    string name;

    int priority;

    string condition;

};

int Heap[1000], sz = -1;

int parent(int i)

{

    return (i - 1) / 2;

}

void shiftUp(int i)

{

    if (i == 0)

        return;

    if (Heap[parent(i)] < Heap[i])

    {

        swap(Heap[parent(i)], Heap[i]);

        shiftUp(parent(i));

    }

}

void insert(int k)

{

    Heap[++sz] = k;

    shiftUp(sz);

}

int shiftDown(int i)

{

    int left = 2 \* i + 1;

    int right = 2 \* i + 2;

    int largest = i;

    if (left <= sz && Heap[left] > Heap[largest])

        largest = left;

    if (right <= sz && Heap[right] > Heap[largest])

        largest = right;

    if (largest != i)

    {

        swap(Heap[i], Heap[largest]);

        shiftDown(largest);

    }

    return largest;

}

int leftChild(int i)

{

    return 2 \* i + 1;

}

int rightChild(int i)

{

    return 2 \* i + 2;

}

int extractMax()

{

    int ret = Heap[0];

    Heap[0] = Heap[sz--];

    shiftDown(0);

    return ret;

}

int main()

{

    cout << "Enter the number of patients: ";

    int n;

    cin >> n;

    patient p[n];

    for (int i = 0; i < n; i++)

    {

        cout << "Enter the name of patient " << i + 1 << ": ";

        cin >> p[i].name;

        cout << "Enter the priority of patient " << i + 1 << ": ";

        cin >> p[i].priority;

        cout << "Enter the condition of patient " << i + 1 << ": ";

        cin >> p[i].condition;

        insert(p[i].priority);

    }

    cout << "The patients are: " << endl;

    for (int i = 0; i < n; i++)

    {

        cout << "Name: " << p[i].name << endl;

        cout << "Priority: " << p[i].priority << endl;

        cout << "Condition: " << p[i].condition << endl;

    }

    cout << "The patients in order of priority are: " << endl;

    for (int i = 0; i < n; i++)

    {

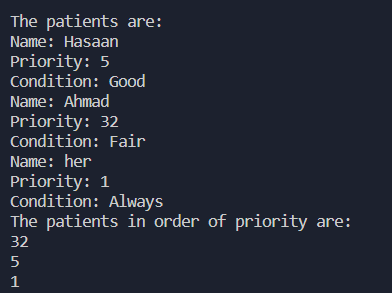
        cout << extractMax() << endl;

    }

    return 0;

}

**Output:**

****