**M. Ali. Arslan**

**19F-0348**

**Lab-08**

**Task # 01**

#include <iostream>

#include<queue>

using namespace std;

void levelOrder(int\* a, int n)

{

queue<int> q;

q.push(1);

while (!q.empty())

{

int size = q.size();

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

{

int node = q.front();

cout << a[node] << " ";

q.pop();

if (2 \* node <= n)

q.push(2 \* node);

if (2 \* node + 1 <= n)

q.push(2 \* node + 1);

}

}

}

void inorder(int\* a, int n, int i)

{

if (i > n)

return;

inorder(a, n, 2 \* i);

cout << a[i] << " ";

inorder(a, n, 2 \* i + 1);

}

void max\_heap(int\* a, int m, int n) {

int j, t;

t = a[m];

j = 2 \* m;

while (j <= n)

{

if (j < n && a[j + 1] > a[j])

j = j + 1;

if (t > a[j])

break;

else if (t <= a[j])

{

a[j / 2] = a[j];

j = 2 \* j;

}

}

a[j / 2] = t;

return;

}

void build\_maxheap(int\* arr, int n)

{

int k;

for (k = n / 2; k >= 1; k--)

{

max\_heap(arr, k, n);

}

}

int main()

{

int n, i;

int arr[100];

cout << "Enter number of elements: ";

cin >> n;

for (i = 1; i <= n; i++)

{

cin >> arr[i];

}

build\_maxheap(arr, n);

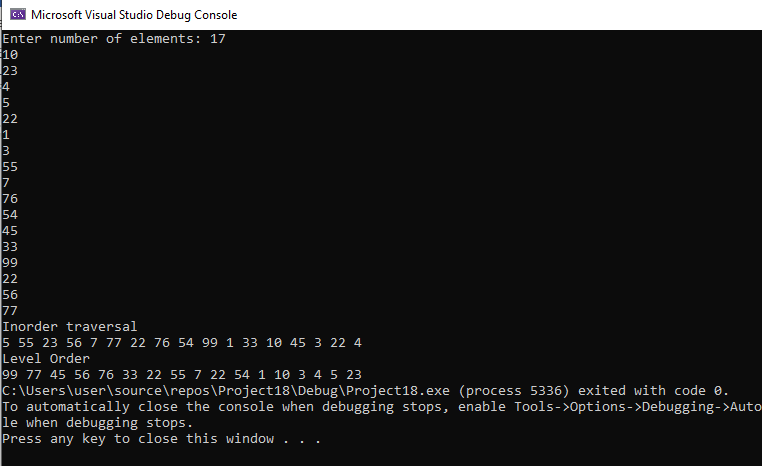
cout << "Inorder traversal \n";

inorder(arr, n, 1);

cout << "\nLevel Order \n";

levelOrder(arr, n);

}



**Task # 02**

#include <iostream>

using namespace std;

void heapify(int arr[], int n, int i);

void heapSort(int arr[], int n);

void displayArr(int arr[], int n);

void input(int arr[], int n);

int main()

{

int n;

cout << "Input the capacity of heap: ";

cin >> n;

int\* arr = new int[n];

input(arr, n);

cout << "\nInput array is \n";

displayArr(arr, n);

heapSort(arr, n);

cout << endl;

cout << "Sorted array is \n";

displayArr(arr, n);

}

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

{

int largest = i;

int leftChild = 2 \* i + 1;

int rightChild = 2 \* i + 2;

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

largest = leftChild;

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

largest = rightChild;

if (largest != i) {

swap(arr[i], arr[largest]);

heapify(arr, n, largest);

}

}

void heapSort(int arr[], int n)

{

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

heapify(arr, n, i);

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

swap(arr[0], arr[i]);

heapify(arr, i, 0);

}

}

void displayArr(int arr[], int n)

{

for (int i = 0; i < n; ++i)

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

cout << endl;

}

void input(int arr[], int n)

{

for (int i = 0; i < n; i++)

{

cout << " Input for Index[" << i << "]: ";

cin >> arr[i];

}

}

