

```

#include <iostream>

#include <stdio.h>

#include <string>

#include <stdlib.h>

#include <iomanip>    // std::setw

// #define N 10


using namespace std;


class BinaryHeap
{
public:
    BinaryHeap();                // Construction
    bool less(int i, int j);
    void exch(int i, int j);
    void swim(int k);
    void sink(int k);
    bool isEmpty();
    int size();
    void insert(int v);
    int delMax();
    void ListArray();
        void printT(int x, int id);


private:
    int N = 0;
    int *pq;

```

```

int capacity = 100;

};

// Initialize the class
BinaryHeap::BinaryHeap()
{
    pq = new int[capacity];
    cout << "A new priority queue with " << capacity << " capacity was created...!" << endl ;
}

void BinaryHeap::ListArray()
{
    for (int i=1; i <= N; i++) // Remember we have "size" items
    {
        cout << pq[i] << " , ";
    }
}

void BinaryHeap::swim(int k)
{
    while (k > 1 && less(k/2, k))
    {
        exch(k/2, k);
        k = k/2;
    }
}

bool BinaryHeap::isEmpty()
{ return N == 0; }

```

```
int BinaryHeap::size()
```

```
{ return N; }
```

```
void BinaryHeap::insert(int v)
```

```
{
```

```
    pq[++N] = v;
```

```
    swim(N);
```

```
}
```

```
int BinaryHeap::delMax()
```

```
{
```

```
    int max = pq[1];
```

```
    exch(1, N--);
```

```
    pq[N+1] = NULL;
```

```
    sink(1);
```

```
    return max;
```

```
}
```

```
void BinaryHeap::sink(int k)
```

```
{
```

```
    while (2*k <= N)
```

```
    {
```

```
        int j = 2*k;
```

```
        if (j < N && less(j, j+1)) j++;
```

```
        if (!less(k, j)) break;
```

```
        exch(k, j);
```

```
        k = j;
```

```
    }
```

```
}
```

```
bool BinaryHeap::less(int i, int j)
```

```
{  
    if (pq[i] < pq[j])  
        return true;  
    return false;  
}
```

```
void BinaryHeap::exch(int i, int j)
```

```
{  
    int t = pq[i]; pq[i] = pq[j]; pq[j] = t;  
}
```

```
//1-> 2, 3
```

```
//2-> 4, 5
```

```
//3-> 6, 7
```

```
void BinaryHeap::printT(int x, int id)
```

```
{  
    if (x>N) return;
```

```
    printT(2*x+1,id+10);
```

```
    cout << setw(id) << ' ' << pq[x] << endl;
```

```
    printT(2*x,id+10);
```

```
}
```

```
// The program lunches here
```

```
int main( )
```

```
{  
    BinaryHeap BH;
```

```
        for (int i=0; i < 20; i++)  
            BH.insert( rand()%50 +1);  
  
    BH.ListArray();  
    cout<< " -----\n ";  
    BH.printT(1,10);  
  
}
```