**Practical-3**: Implementation of operations on binary heaps.

#include<iostream.h>

void swap(int \*x, int \*y);

class MinHeap

{

int \*harr;

int capacity;

int heap\_size;

public:

MinHeap(int capacity);

void MinHeapify(int );

int parent(int i) { return (i-1)/2; }

int left(int i) { return (2\*i + 1); }

int right(int i) { return (2\*i + 2); }

int extractMin();

void decreaseKey(int i, int new\_val);

int getMin() { return harr[0]; }

void deleteKey(int i);

void insertKey(int k);

};

MinHeap::MinHeap(int cap)

{

heap\_size = 0;

capacity = cap;

harr = new int[cap];

}

void MinHeap::insertKey(int k)

{

if (heap\_size == capacity)

{

cout << "\nOverflow: Could not insertKey\n";

return;

}

heap\_size++;

int i = heap\_size - 1;

harr[i] = k;

while (i != 0 && harr[parent(i)] > harr[i])

{

swap(&harr[i], &harr[parent(i)]);

i = parent(i);

}

}

void MinHeap::decreaseKey(int i, int new\_val)

{

harr[i] = new\_val;

while (i != 0 && harr[parent(i)] > harr[i])

{

swap(&harr[i], &harr[parent(i)]);

i = parent(i);

}

}

int MinHeap::extractMin()

{

if (heap\_size <= 0)

return INT\_MAX;

if (heap\_size == 1)

{

heap\_size--;

return harr[0];

}

int root = harr[0];

harr[0] = harr[heap\_size-1];

heap\_size--;

MinHeapify(0);

return root;

}

void MinHeap::deleteKey(int i)

{

decreaseKey(i, INT\_MIN);

extractMin();

}

void MinHeap::MinHeapify(int i)

{

int l = left(i);

int r = right(i);

int smallest = i;

if (l < heap\_size && harr[l] < harr[i])

smallest = l;

if (r < heap\_size && harr[r] < harr[smallest])

smallest = r;

if (smallest != i)

{

swap(&harr[i], &harr[smallest]);

MinHeapify(smallest);

}

}

void swap(int \*x, int \*y)

{

int temp = \*x;

\*x = \*y;

\*y = temp;

}

int main()

{

MinHeap h(11);

h.insertKey(3);

h.insertKey(2);

h.deleteKey(1);

h.insertKey(15);

h.insertKey(5);

h.insertKey(4);

h.insertKey(45);

cout <<"Ectracted Min: "<< h.extractMin() << "\n";

cout << "GetMin after ExtractMin (New Min): "<< h.getMin() << "\n";

h.decreaseKey(2, 1);

cout << "New Min after decreaseKey: "<<h.getMin();

return 0;

}

**Output:**

****