

Aim:

Write a C program to implement Binary Heap operations.

Source Code:

binaryheap.c

```
// Type your content here...
#include <stdio.h>
#include <stdlib.h>
#include <limits.h>

typedef struct {
    int*arr;
    int capacity;
    int size;
} BinaryHeap;

BinaryHeap* createHeap(int capacity) {
    BinaryHeap *heap = (BinaryHeap*)malloc(sizeof(BinaryHeap));
    heap->capacity = capacity;
    heap->size = 0;
    heap->arr = (int*)malloc(capacity* sizeof(int));
    return heap;
}

void swap(int *a, int *b) {
    int temp = *a;
    *a = *b;
    *b = temp;
}

void heapifyup (BinaryHeap *heap, int index) {
    while (index != 0 && heap->arr[(index -1)/2]>heap->arr[index]){
        swap(&heap->arr[(index-1)/2], &heap->arr[index]);
        index = (index -1)/2;
    }
}

void insert(BinaryHeap *heap, int value) {
    if (heap->size == heap->capacity){
        printf("Heap is full\n");
        return;
    }
    heap->arr[heap->size] = value;
    heap->size++;
    heapifyup(heap, heap->size -1);
}

void heapifyDown(BinaryHeap *heap, int index) {
    int smallest = index;
    int left = 2* index +1;
    int right = 2 * index+2;
```

```

if (left < heap->size && heap->arr [left] < heap ->arr [smallest])
smallest = left;
if (right < heap->size && heap-> arr[right]<heap->arr[smallest])
smallest = right;
if (smallest != index) {
swap(&heap->arr[index], &heap->arr[smallest]);
heapifyDown(heap, smallest);
}
}
int extractMin(BinaryHeap *heap) {
if (heap->size <=0)
return INT_MAX;

if (heap->size == 1){
heap->size--;
return heap->arr[0];
}
int min = heap->arr[0];
heap->arr[0] = heap->arr[heap->size -1];
heap->size--;
heapifyDown(heap, 0);

return min;
}
void printHeap(BinaryHeap *heap) {
for (int i = 0; i<heap->size; i++)
printf ("%d ", heap->arr[i]);
printf("\n");
}
int getMin(BinaryHeap *heap) {
if (heap->size <=0)
return INT_MAX;
return heap->arr[0];
}
int main() {
int capacity, element;

printf("Enter the capacity of the Binary Heap: ");
scanf("%d", &capacity);
BinaryHeap *heap = createHeap(capacity);
printf ("Enter elements to add to the heap (-1 to stop):\n");
while (1){
scanf("%d", &element);
if (element == -1) break;
insert(heap, element);
}
printf("Heap after insertions:\nHeap: ");
printHeap(heap );
printf("Min value is %d\n", getMin(heap));
free(heap->arr);
free(heap);
return 0;
}

```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter the capacity of the Binary Heap: 10
Enter elements to add to the heap (-1 to stop): 10 5 20 3 8 15 30 -1
Heap after insertions:
Heap: 3 5 15 10 8 20 30
Min value is 3

Test Case - 2
User Output
Enter the capacity of the Binary Heap: 8
Enter elements to add to the heap (-1 to stop): 4 1 2 6 7 3 8 5 -1
Heap after insertions:
Heap: 1 4 2 5 7 3 8 6
Min value is 1

