Heap Deletion-

```
// Online C++ compiler to run C++ program online
#include <iostream>
using namespace std;

class MaxHeap{
    int * arr;
    int size;         //total elements in the heap
    int total_size;   //total size of an array

    public:
    MaxHeap(int n){
        arr=new int [n];
        size=0;
        total_size=n;
    }

    void Insert(int value){
        if(size==total_size){
            cout<<"overflow"<<endl;
            return;
        }
        arr[size]=value;
        int i=size;
        size++;

        while(i > 0 && arr[(i-1)/2] < arr[i]){
            swap(arr[i],arr[(i-1)/2]);
            i=(i-1)/2;
        }

        }
        void print(){
            for(int i=0;i<size;i++){
                cout<<arr[i]<<" ";

            }
              cout<<endl;
```

```
    }

    void Heapify(int index)
{
  int largest = index;
  int left = 2 * index + 1;
  int right = 2 * index + 2;

  // Largest will store the index of the element which
  // is greater between parent, left child and right child

  if (left < size && arr[left] > arr[largest])
    largest = left;

  if (right < size && arr[right] > arr[largest])
    largest = right;

  if (largest != index)
  {
    swap(arr[index], arr[largest]);
    Heapify(largest);
  }
}

void Delete()
{
  if (size == 0)
  {
    cout << "Heap Underflow"<<endl;;
    return;
  }

  cout << arr[0] << " deleted from the heap";
  arr[0] = arr[size - 1];
  size--;

  if (size == 0){
    return;
  }
```

```
    Heapify(0);
}

    };

int main() {
  MaxHeap a(9);
  a.Insert(14);
  a.Insert(24);
  a.Insert(12);
  a.Insert(11);
  a.Insert(25);
  a.Insert(8);
  a.Insert(35);
  a.print();
  a.Delete();

    return 0;
}
```

Output-

```
35 24 25 11 14 8 12
35 deleted from the heap
```