Task 1

Java code for heap

```
package heap;

import java.util.Arrays;
import java.util.NoSuchElementException;

import static java.lang.Integer.MIN_VALUE;
import static java.lang.Math.pow;

public class heap {
    private double x;
    private int size;
    private int[] heapArray;

//    Constructor
    public heap(double x, int capacity){
        this.size=0;
        heapArray=new int[capacity+1];
        this.x=x;
        Arrays.fill(heapArray,-1);
    }

    private int parent(int i){
        return (int) ((int)(i-1)/pow(2,x));

    }

    public boolean isFull(){
        return this.size == heapArray.length;
    }

    public void insert(int x){
        if(isFull()){
            throw new NoSuchElementException("Heap is full, no space to insert new
element.");
        }
        else{
            heapArray[size++]=x;
            heapifUp(size-1);
        }
    }

    private void heapifUp(int i){
        int tmp=heapArray[i];
        while(i>0 && tmp>heapArray[parent(i)]) {
            heapArray[i] = heapArray[parent(i)];
            i = parent(i);
        }
        heapArray[i]=tmp;
    }

    public int popMax(){
       int pop=heapArray[0];
       heapArray[0]=heapArray[size-1];
       heapArray[size-1]=-1;
       size--;
       int i=0;
       while(i<size-1){
           heapifUp(i);
           i++;
       }
       return pop;
    }
    public void print(){
        for(int i=0;i<size;i++){
            System.out.print(heapArray[i]);
            System.out.print(',');
        }
        System.out.println();
    }

}
```

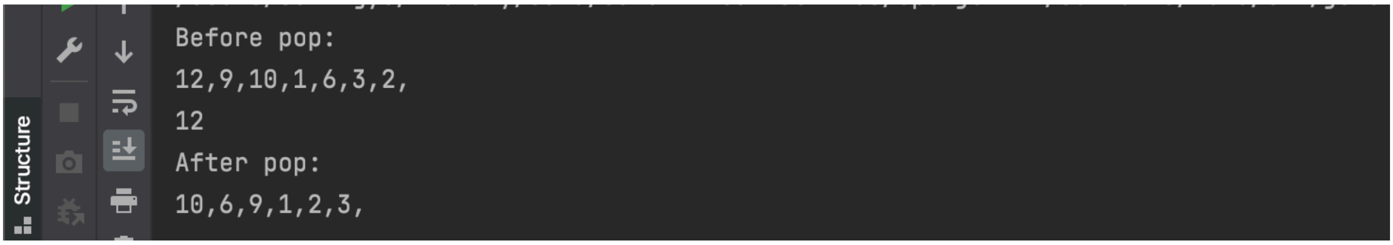
Code for Test:

```
package heap;

public class Tester {


    public static void main(String[] args){
        heap test1=new heap(1,10);
        test1.insert(3);
        test1.insert(1);
        test1.insert(10);
        test1.insert(9);
        test1.insert(6);
        test1.insert(12);
        test1.insert(2);
        System.out.println("Before pop:");
        test1.print();
        System.out.println(test1.popMax());
        System.out.println("After pop:");
        test1.print();
    }
}
```

The result:



```
Before pop:
12,9,10,1,6,3,2,
12
After pop:
10,6,9,1,2,3,
```