```java
import java.util.Arrays;
import java.util.HashSet;
import java.util.Set;

public class main {
   public static void main(String[] args){
       main m=new main();
       System.out.println("-----Problem 1 testing-----");
       System.out.println(m.binarySqrt(28));
       System.out.println(m.binarySqrt(16));
       System.out.println(m.binarySqrt(71));
       System.out.println(m.binarySqrt(144)+"\n");
       System.out.println("-----Problem 2 testing-----");
       int[] a={0,1,3,6,8,9};
       System.out.println(m.smallestMissing(a,10));
       int[] b={2,5,7,11};
       System.out.println(m.smallestMissing(b,15));
       int[] c={0,1,2,3,4};
       System.out.println(m.smallestMissing(c,8));
       int[] d={12};
       System.out.println(m.smallestMissing(d,13));
       Set<Integer> set=new HashSet<>();
       for(int i=0;i<10;i++){
           set.add((int)(Math.random()*10));
       }
       int[] arr=set.stream().mapToInt(Integer::intValue).toArray();
       Arrays.sort(arr);
       System.out.println("Random Generated Array: "+ Arrays.toString(arr));
       System.out.println(m.smallestMissing(arr,10));

   }

   //Problem#1
   public int binarySqrt(int n){
       int mid=n;
       int start=0;
       int end=n;
       if(n<=1){
           return n;
       }

       while(mid*mid>n){

           mid=Math.round((float)(start+end)/2);

           if(mid*mid<n & (mid+1)*(mid+1)>n){
               return mid+1;
           }
```

```
        if((mid+1)*(mid+1)<n){
            return mid+1;
        }

        if((mid-1)*(mid-1)<n & (mid*mid)>n){
            return mid;
        }

        if(mid*mid>n){
            end=mid;
        }

        else{
            start=mid;
        }

    }

    return mid;
}

//Problem#2
public int smallestMissing(int[] a, int m){
    int start=0;
    int end=a.length-1;
    int mid=0;
    while(start<end+1){
        mid=(start+end)/2;

        if(a[mid]>mid){
            //System.out.println("ok");
            end=mid-1;
        }
        else if(mid>=a[mid]){
            start=mid+1;
        }

        if(mid<end){
            if(a[mid]<a[mid+1]-1){
                break;
            }
        }

    }

    if(mid==0&a[mid]!=0){
        return 0;
    }
    else if(a[mid]<m){
```

```
            return a[mid]+1;
        }
        else{
            return -1;
        }

    }
}
```