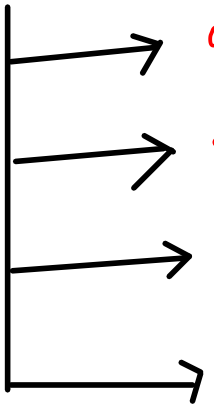
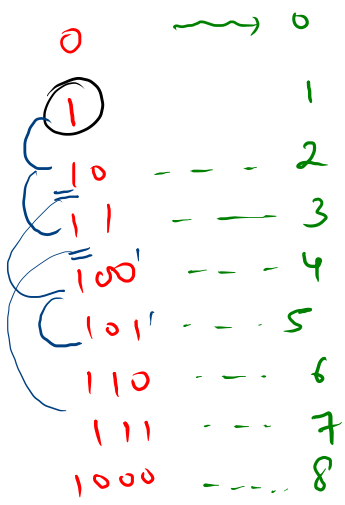
Queue.



→ get front element.

Decimal: 0 1 2 3 4 5 6 7 8 9

Binary



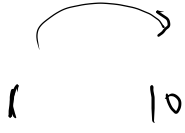
$(78)_{10} = 7 \times 10^1 + 8 \times 10^0 = 78$

$(110)_2$

$2^2\ 2^1\ 2^0$

$2^2 \times 1 + 2^1 \times 1 + 2^0 \times 0$

$4 + 2 + 0 = 6$

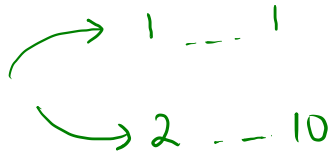


1 ~~10~~ 11 100 101

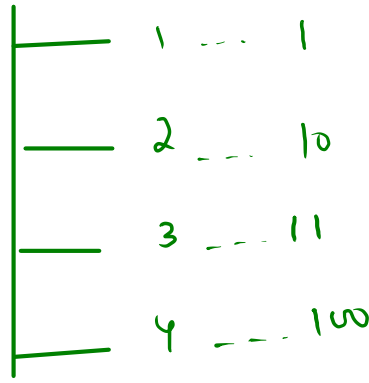
Generate Binary Numbers

eg1.

N = 2



eg2. N = 4



generate Binary Number.

N=5

1 ---- N

"1" "111" "1000" "1001" "1010" "1011"

1
10
11
100
101

rem = "101" ---- 1010

1011

print N numbers
N times.

loop (N times)

{

}

N = 5

110 111 1000 1001 1010 1011

loop (N times)
{
rem
print
add 2 more
}

100 → 0 → 1000
100 → 1 → 1001

101 → 0 → 1010
101 → 1 → 1011

1
10
11
100
101

```java
    //function to generate binary numbers from 1 to N using
    static ArrayList<String> generate(int N)
    {
        ArrayList<String> ans = new ArrayList<>();
        Queue<String> qu = new LinkedList<>();
        qu.add("1");
        for(int i = 1; i <= N; i++){
            String rm = qu.remove();
            ans.add(rm);
            qu.add(rm+"0");
            qu.add(rm+"1");
        }



        return ans;
    }

}
```

ans <"1", "10" "11" >

rm = "11"

N=3

"100" "101" "110" "111"

First Negative Integer 2

N - K + 1

5 - 2 + 1 = 4

3

Problem | Submissions | Leaderboard | Discussions

Given an array **A[]** of size **N** and a positive integer **K**, find the first negative integer for each and every window(**contiguous subarray**) of size **K**.

N = 5

K = 2

-8 2 3 -6 10

-8 0 -6 -6

Sample Input 0

```
5 2
-8 2 3 -6 10
```

Sample Output 0

```
-8 0 -6 -6
```

N = 9

K = 3

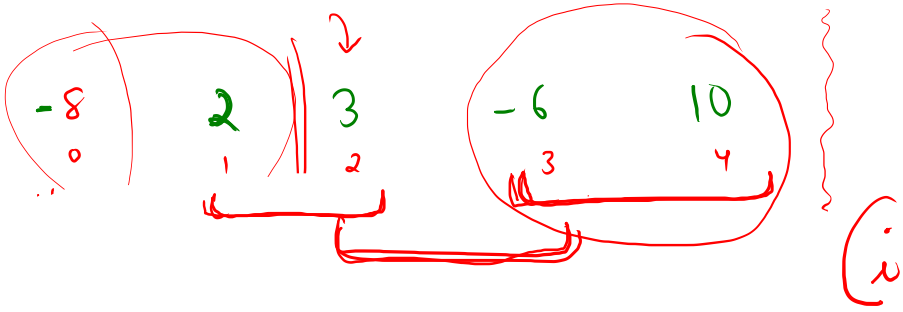
-8 2 -2 -3 4 5 6 -7 8

-8 -2 -2 -3 0 -7 -7

9 - 3 + 1

7

```
5 2
-8 2 3 -6 10
```



K = 2

Algo

- ✓ find ans prev
- ✓ remove unnecessary
- ✓ add -ve

idx of -ve ele

3 (-6)

↳ qu.peek < i - k + 1

3 < 4 - 2 + 1

3 < 3 ✗

A[qu.peek]
-8 ✓
0
-6
-6

N = 9

K = 3

-8	2	-2	-3	4	5	6	-7	8
0	1	2	3	4	5	6	7	8

i

i

find p ans ✓

rm unnecessary ✓

add -ve

idx of -ve

7

peek < i-k+1

A[peek]

-8

-2

-2

-3

0

-7

-7

2 -1 3

```java
import java.io.*;
import java.util.*;
public class Solution {
    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();
        int k = scn.nextInt();
        int [] A = new int[n];
        for(int i = 0; i < n; i++){
            A[i] = scn.nextInt();
        }
        //first k item
        Queue<Integer> qu = new LinkedList();
        int i = 0;
        while(i < k){
            if(A[i]<0){
                qu.add(i);
            }
            i++;
        }
        //rest item 1. find p ans 2. rm unnece.. 3.add -ve
        while(i < n){
            //find p ans
            if(qu.size() == 0){
                System.out.print(0 + " ");
            }else{
                System.out.print(A[qu.peek()] + " ");
            }
            //remove unnece..
            if(qu.size() != 0 && qu.peek() < i-k+1){
                qu.remove();
            }
            //add -ve
            if(A[i] < 0){
                qu.add(i);
            }
            i++;
        }
        if(qu.size() == 0){
            System.out.print(0 + " ");
        }else{
            System.out.print(A[qu.peek()] + " ");
        }
    }
}
```

Priority Queue.

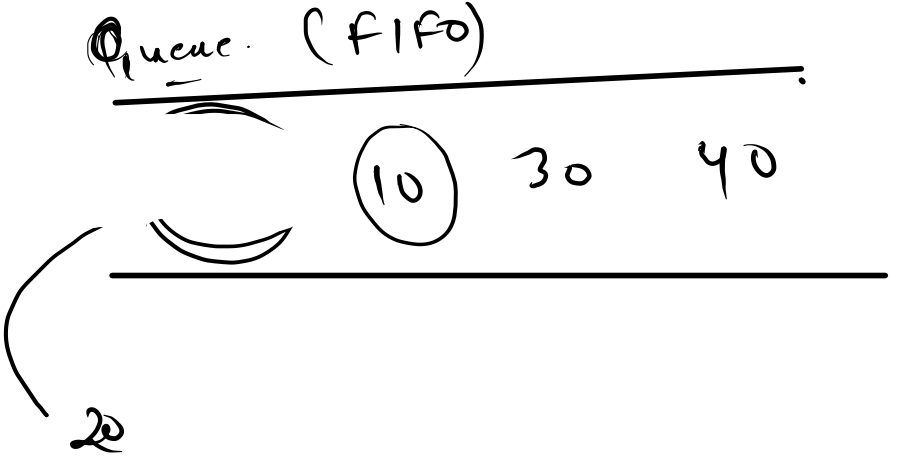
init

add

remove

size

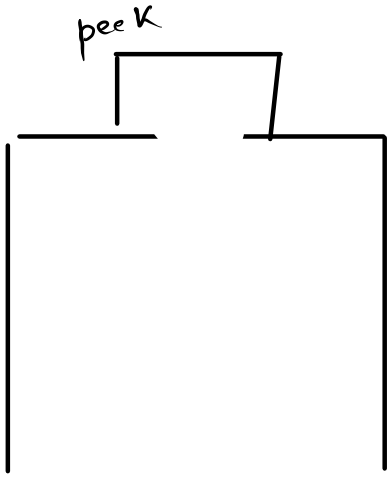
get



Priority Queue.

default : min.

20
10
30
40

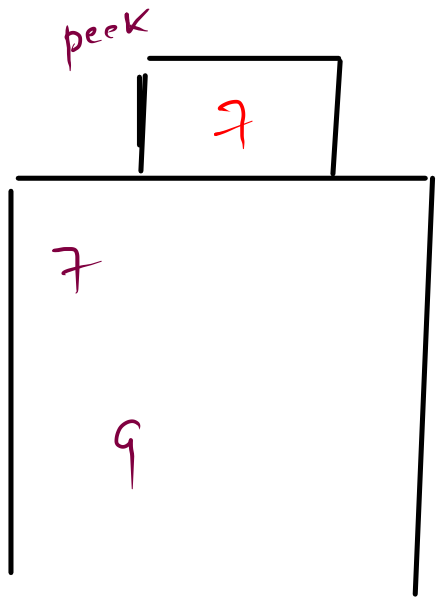


min. (asc.)

[10 20 30 40]

min.

A → 7 6 6 2 5 9 1 3



1 2 3 5 6 6

```java
import java.util.PriorityQueue;
import java.util.*;
public class Main
{
    public static void main(String[] args) {
        PriorityQueue<Integer> pq = new PriorityQueue<>();  //min
        pq.add(7);
        pq.add(2);
        pq.add(1);
        pq.add(3);
        pq.add(9);

        System.out.println(pq.remove());
        System.out.println(pq.peek());

        System.out.println(pq.size());

    }
}

```

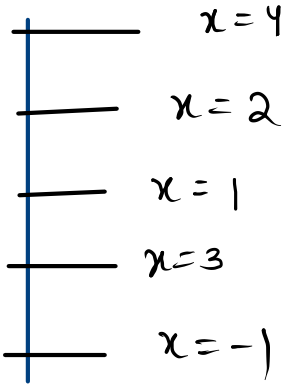
Priority Queue Basics

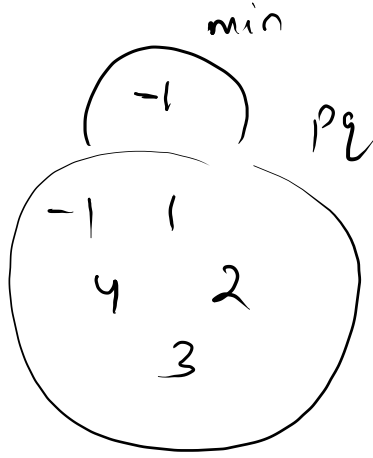
Take **T** as an integer input. Then take **t** integer elements as input. Each time you take an input. Print the **smallest element** so far, each time a new element is taken as an input.

T=5

Sample Input 0

```
5
4
2
1
3
-1
```





O/P

4
2
1
1
-1

```java
import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int t = scn.nextInt();
        PriorityQueue<Integer> pq = new PriorityQueue();     //min

        for(int i = 1; i <= t; i++){
            int x = scn.nextInt();
            pq.add(x);
            System.out.println(pq.peek());
        }
    }
}
```