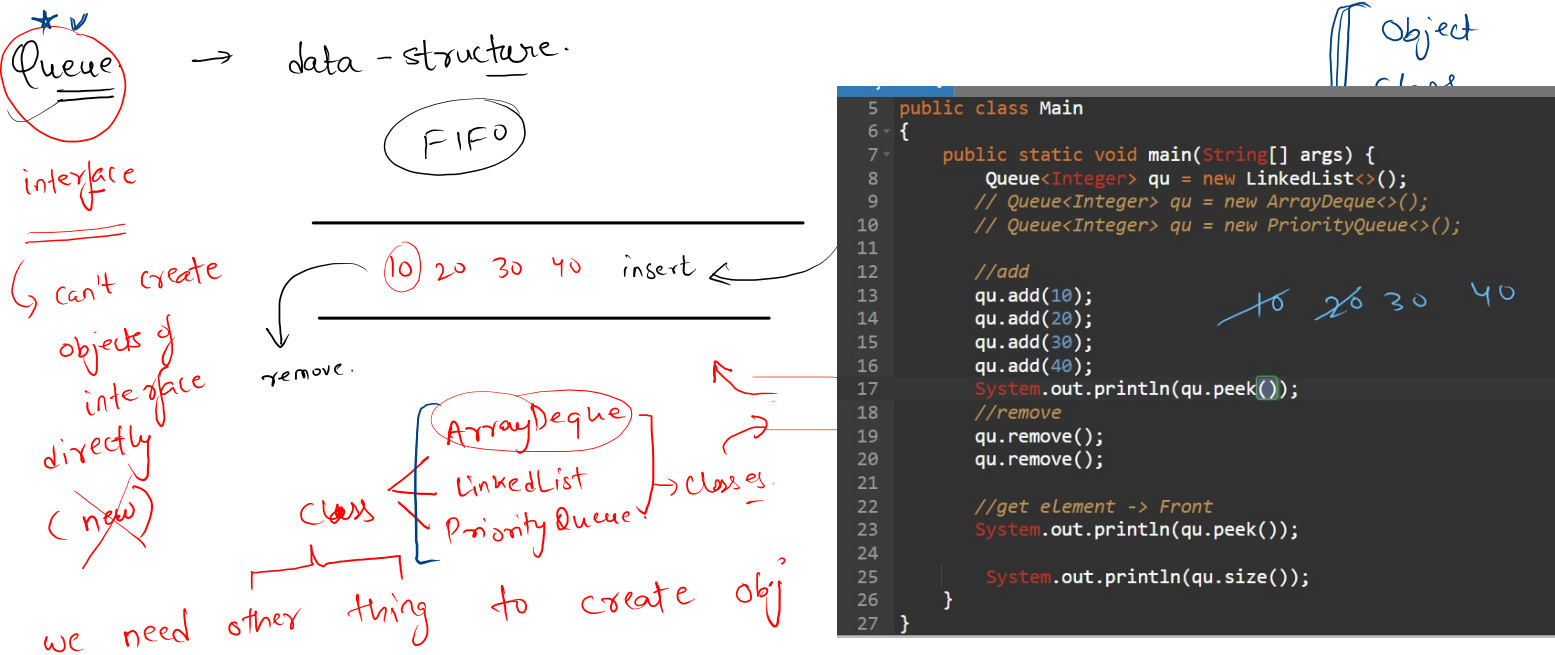



```java
public class Main
{
    public static void main(String[] args) {
        Queue<Integer> qu = new LinkedList<>();
        // Queue<Integer> qu = new ArrayDeque<>();
        // Queue<Integer> qu = new PriorityQueue<>();

        //add
        qu.add(10);
        qu.add(20);
        qu.add(30);
        qu.add(40);
        System.out.println(qu.peek());
        //remove
        qu.remove();
        qu.remove();

        //get element -> Front
        System.out.println(qu.peek());

        System.out.println(qu.size());
    }
}
```

Queue.

- init
- add
- remove
- get
- size
- traverse

```java
public class Main
{
    public static void main(String[] args) {
        Queue<Integer> qu = new LinkedList<>();
        // Queue<Integer> qu = new ArrayDeque<>();
        // Queue<Integer> qu = new PriorityQueue<>();

        //add
        qu.add(10);
        qu.add(20);
        qu.add(30);
        qu.add(40);
        System.out.println(qu.peek());
        //remove
        qu.remove();
        qu.remove();

        //get element -> Front
        System.out.println(qu.peek());

        System.out.println(qu.size());
    }
}
```

10 20 30 40

Queue Syntax Learning

1. Declare an Empty $queue\ s$.
2. Take Single Integer T as input.
3. For next T Lines format $(case, x(optional))$

- case 1. $Print$ the $size$ of the $queue$ in a separate line.
- case 2. $Remove$ an element from the queue. If the queue is empty then print -1 in a separate line.
- case 3. Add Integer x to the $queue\ s$.
- case 4. $Print$ an element at the $front$ of the $queue$. If queue is empty print -1 in a seperate line.

Sample Input 0

```
5
1
2
3 9
4
1
```

```
9
```

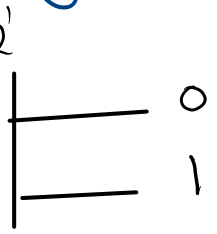
Sample Output 0

```
0
-1
9
1
```

```java
import java.util.*;
public class Solution{
    public static void main(String [] args){
        Scanner scn = new Scanner(System.in);
        int t = scn.nextInt();
        Queue<Integer> qu = new LinkedList<>();

        for(int i = 0; i < t; i++){
            int caseNu = scn.nextInt();
            if(caseNu == 1){
                System.out.println(qu.size());
            }else if(caseNu == 2){
                if(qu.size() == 0){
                    System.out.println(-1);
                }else{
                    qu.remove();
                }
            }else if(caseNu == 3){
                int x = scn.nextInt();
                qu.add(x);
            }else if(caseNu == 4){
                if(qu.size() == 0){
                    System.out.println(-1);
                }else{
                    System.out.println(qu.peek());
                }
            }
        }
    }
}
```

Binary Number System



Decimal		
0	10	20
1	11	21
2	12	
3	13	
4		
5		
6		
7		
8		
9	19	29

0 ... 0

1 ... 1

2 ... 10

3 ... 11

100

101

110

111

1000

1001

Ternary:

0

1

2

10

11

12

20

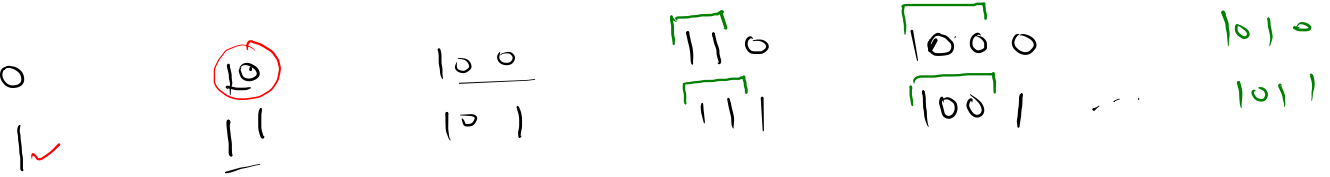
21

22

100

101

102



Print Binary

Problem | Submissions | Leaderboard | Discussions

Given a number **N**. The task is to generate and print all **binary numbers** with decimal values from **1 to N.** (Note : Use the **queue** for implementation.)

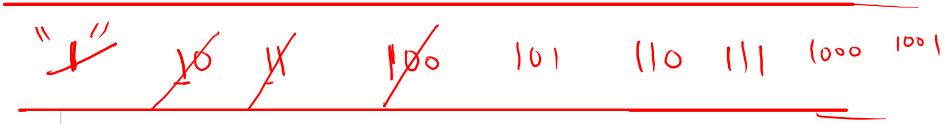
NOTE :- After answering the question, attempt the related question in the linked resource to improve your understanding of this question .Click here

Sample Input 0

```
4
```

Sample Output 0

```
1 10 11 100
```



```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();
        Queue<String> qu = new LinkedList<>();
        qu.add("1");
        //rem print add2
        for(int i = 0; i < n; i++){
            String rem = qu.remove();
            System.out.print(rem + " ");
            qu.add(rem + "0");
            qu.add(rem + "1");
        }

    }
}
```

```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();
        Queue<String> qu = new LinkedList<>();
        qu.add("1");
        //rem print add2
        for(int i = 0; i < n; i++){
            String rem = qu.remove();
            System.out.print(rem + " ");
            qu.add(rem + "0");
            qu.add(rem + "1");
        }

    }
}
```

FIFO

rem = 1

10

11

1 (1)

2

3

4

0
1
2 ✓
3
4
5
6
7
8
9

10
11
12
13
14
15
16
17
18
19

20
21
22
23
24
25
26
27
28
29

0 10 100 110
1 11 101 111

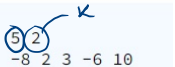
---

0 10 20 100
1 11 21 101
2 12 22 102

First Negative Integer 2

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**.

Sample Input 0



Sample Output 0

n-k+1 ⇒ total window

k=2

-8	2	3	-6	10
0	1	2	3	4

i

1. first k : -ve → add
2. rest :
   - ↳ previous ans.
   - ↳ remove unnecessary
     peek() < i-k+1
   - ↳ add -ve

idx.

3

$k=3$

$-8 \quad 2 \quad -2 \quad -3 \quad 4 \quad 5 \quad 6 \quad -7 \quad 8$