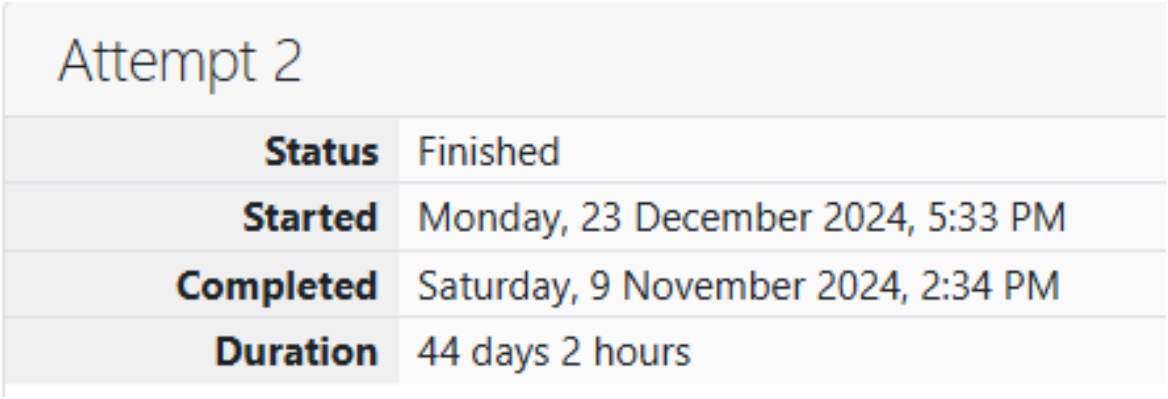
Week 4 – 1

Decision making and Looping

Roll no: 240801137

Name : Jhanani.M



Problem 1:

Alice and Bob are playing a game called "Stone Game". Stone game is a two-player game. Let N be the total number of stones. In each turn, a player can remove either one stone or four stones. The player who picks the last stone, wins. They follow the "Ladies First" norm. Hence Alice is always the one to make the first move. Your task is to find out whether Alice can win, if both play the game optimally.

Input Format

First line starts with T, which is the number of test cases. Each test case will contain N number of stones.

Output Format

Print "Yes" in the case Alice wins, else print "No".

Constraints

1<=T<=1000 1<=N<=10000

Sample Input

3

1

6

7

Sample Output

Yes

Yes

No

Code:

```c
#include <stdio.h>
int main()
{
    int T,N;
    scanf("%d",&T);
    while(T--){
        scanf("%d",&N);
        if(N % 8 == 2 || N % 8 == 5 || N % 8 == 7 || N % 8 == 0){
            printf("No\n");
        }
        else{
            printf("Yes\n");
        }

    }
    return 0;
}
```

OUTPUT:

	Input	Expected	Got	
✓	3 1 6 7	Yes Yes No	Yes Yes No	✓

Passed all tests! ✓

Problem 2:

You are designing a poster which prints out numbers with a unique style applied to each

of them. The styling is based on the number of closed paths or holes present in a given

number. The number of holes that each of the digits from 0 to 9 have are equal to the number of

closed paths in the digit. Their values are:

1, 2, 3, 5, 7 = 0 holes.

0, 4, 6, 9 = 1 hole.

8 = 2 holes.

Given a number, you must determine the sum of the number of holes for all of its digits.

For example, the number 819 has 3 holes.

Complete the program, it must return an integer denoting the total number of holes in

num.

Constraints

$1 \leq num \leq 109$

Input Format

For Custom Testing There is one line of text containing a single integer num, the value to process.

Sample Input

630

Sample Output

2

Code

```c
#include <stdio.h>
int main()
{
    int N,sum;
    sum = 0;
    scanf("%d",&N);
    while(N>0)
    {
        int r = N % 10;
        if(r == 0 || r == 4 || r== 6 || r == 9)
        {
            sum = sum + 1;
        }
        else if(r == 8){
            sum = sum + 2;
        }
        else {
            sum = sum + 0;
        }
        N = N/10;
    }
    printf("%d",sum);
    return 0;
}
```

OUTPUT:

	Input	Expected	Got	
✓	630	2	2	✓
✓	1288	4	4	✓

Passed all tests! ✓

Problem 3:

The problem solvers have found a new Island for coding and named it as Philaland. These smart people were given a task to make a purchase of items at the Island easier by distributing various coins with different values. Manish has come up with a solution that if we make coins category starting from $1 till the maximum price of the item present on Island, then we can purchase any item easily. He added the following example to prove his point.

Let's suppose the maximum price of an item is 5$ then we can make coins of {$1, $2, $3, $4, $5}to purchase any item ranging from $1 till $5.

Now Manisha, being a keen observer suggested that we could actually minimize the number of coins required and gave following distribution {$1, $2, $3}. According to him any item can be purchased one time ranging from $1 to $5. Everyone was impressed with both of them. Your task is to help Manisha come up with a minimum number of denominations for any arbitrary max price in Philaland.

Input Format

Contains an integer N denoting the maximum price of the item present on Philaland.

Output Format

Print a single line denoting the minimum number of denominations of coins required.

Constraints

1<=T<=100 1<=N<=5000

Sample Input 1:

10

Sample Output 1:

4

Code:

```c
#include <stdio.h>
int main()
{
    int a,c;
    scanf("%d",&a);
    int d= 1;
    while(d<=a)
    {
        c++;
        d = d * 2;
    }
    printf("%d",c);
    return 0;

}
```

OUTPUT:

	Input	Expected	Got	
✓	10	4	4	✓
✓	5	3	3	✓
✓	20	5	5	✓
✓	500	9	9	✓
✓	1000	10	10	✓

Passed all tests! ✓