# **GE23131-Programming Using C-2024**

**Status** Finished

**Started** Monday, 23 December 2024, 5:33 PM **Completed** Thursday, 7 November 2024, 9:42 PM

**Duration** 45 days 19 hours

### **Question 1**

Correct
Marked out of 3.00

Flag question

#### **Question text**

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.



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 and Output

Input

3

1

U

7

Output

Yes

Yes

No

Answer:(penalty regime: 0 %)

```
1 #include <stdio.h>
   int N,T,a;
3 * int main(){
        scanf("%d",&T);
4
 5 -
        for(int i=1;i<=T;i++){</pre>
            scanf("%d",&N);
 6
 7
            a=N/4;
 8 -
            if (N\%4==0\&\&a\%2!=0) | N\%3==0 | N==1
9
                 printf("Yes\n");
10
11 -
            else{
            printf("No\n");
12
13
14
            }
15
16
   }
```

#### **Feedback**

#### **Input Expected Got**

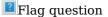
```
3 Yes Yes
1 Yes Yes
6 No No
```

Passed all tests!

#### **Question 2**

Correct

Marked out of 5.00



#### **Question text**

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,  and 7 = 0  holes.
```

$$0, 4, 6, \text{ and } 9 = 1 \text{ hole.}$$

8 = 2 holes.



```
scanf("%d",&n);
 5 🌞
        while(n>0){
 6
             int d=n%10;
             if(d==0||d==4||d==6||d==9){
 7 =
 8
                 h+=1:
 9
10 -
             else if(d==8){
11
                 h+=2;
12
13
        n/=10;
14
        printf("%d",h);
15
16
        return 0;
17
   }
```

#### **Feedback**

#### **Input Expected Got**

630 2 2 1288 4 4

Passed all tests!

#### **Question 3**

Correct
Marked out of 7.00

Flag question

#### **Question text**

The problem solvers have found a new Island for <u>coding</u> 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.

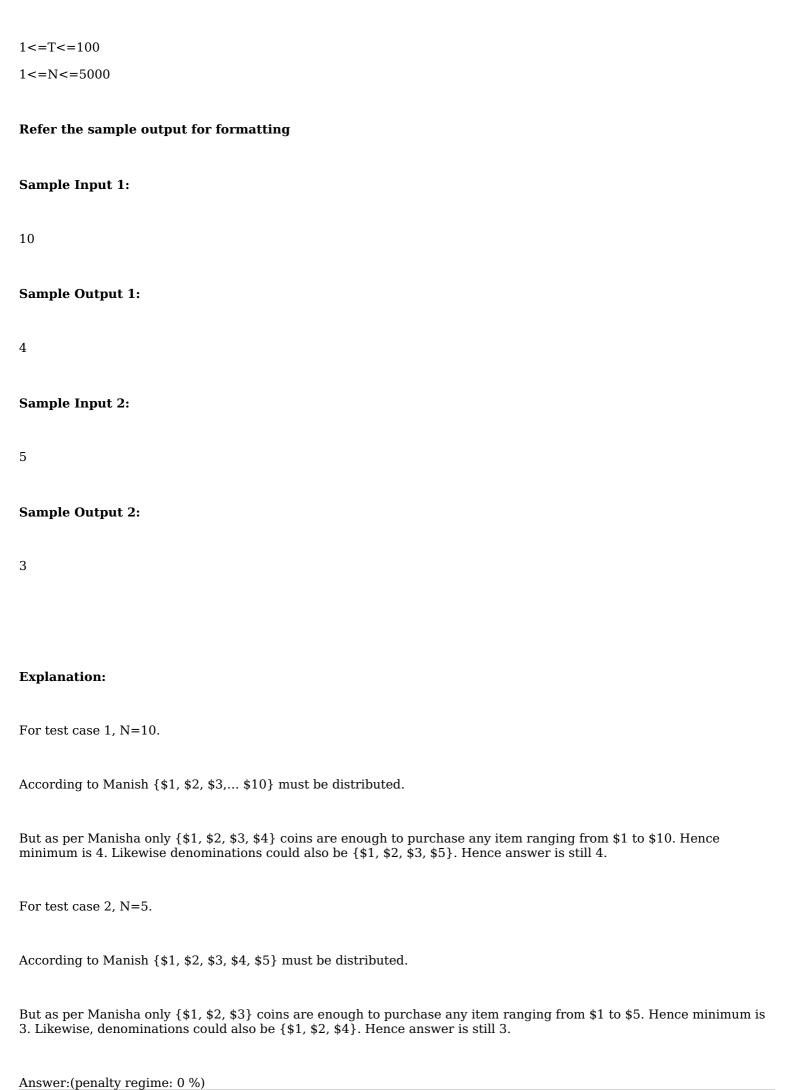
#### **Input Format**

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**



```
2 int main() {
    int N,1, coins=0;
    scanf("%d", &N);
    for(i=1;i<=N;i*=2) {
        coins++;
    }
    printf("%d\n", coins);
    return 0;
}</pre>
```

#### Feedback

### **Input Expected Got**

10	4	4
5	3	3
20	5	5
500	9	9
1000	10	10

Passed all tests!

Finish review

# **Blocks**

Skip Quiz navigation

## **Quiz navigation**

<u>Question 1 This page Question 2 This page Question 3 This page</u>

<u>Show one page at a time</u> Finish review

## **Blocks**