GE23131-Programming Using C-2024



Status Finished Started

Monday, 23 December 2024, 5:33 PM Completed Tuesday, 12 November 2024, 10:55 AM

REC-CIS

Correct Marked out of F Flag question

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.

Input Format

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

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.

Output Format

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

Constraints

1<=T<=1000

1<=N<=10000

Sample Input and Output

Input

6

No

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
       int main()
             int T,i=0,n,t;
scanf("%d",&T);
while(i<T)</pre>
                      scanf("%d",&n);
                      t=n/4;
                     if(t%2 == 0 && n%2 == 0)
printf("No\n");
else if(t%2=1 && n%2=1)
printf("No\n");
10
11
12
13
                     else
printf("Yes\n");
i++;
14
15
16
17
      return 0;
```



unique style applied to each of them. The styling is based on the number of closed paths or holes present in a given number. ₹ Flag question 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, and 9 = 1 hole. Given a number, you must determine the sum of the number of holes for all of its digits. For example, the number 819 has Complete the program, it must must return an integer denoting the total number of holes in num. Constraints 1 ≤ num ≤ 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 Add the holes count for each digit, 6, 3 and 0. Return 1 + 0 +

Sample Case 1

Sample Input

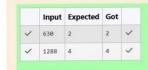
1288

Sample Output

Explanation

Add the holes count for each digit, 1, 2, 8, 8. Return 0 + 0 + 2

= 4.



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.

Print a single line denoting the minimum number of

1<=T<=100

1<=N<=5000

Refer the sample output for formatting

Sample Input 1:

Sample Output 1:

Sample Input 2:

5

Sample Output 2:

For test case 1, N=10.

Explanation:

According to Manish (\$1, \$2, \$3,... \$10) must be distributed.

But as per Manisha only (\$1, \$2, \$3, \$4) coins are enough to purchase any item ranging from \$1 to \$10. Hence minimum is 4. Likewise denominations could also be {\$1, \$2, \$3, \$5}. Hence answer is still 4. For test case 2, N=5.

According to Manish (\$1, \$2, \$3, \$4, \$5) must be distributed.

But as per Manisha only {\$1, \$2, \$3} coins are enough to purchase any item ranging from \$1 to \$5. Hence minimum is 3. Likewise, denominations could also be (\$1, \$2, \$4). Hence answer is still 3.

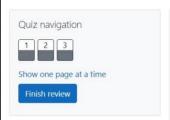
Answer: (penalty regime: 0 %)

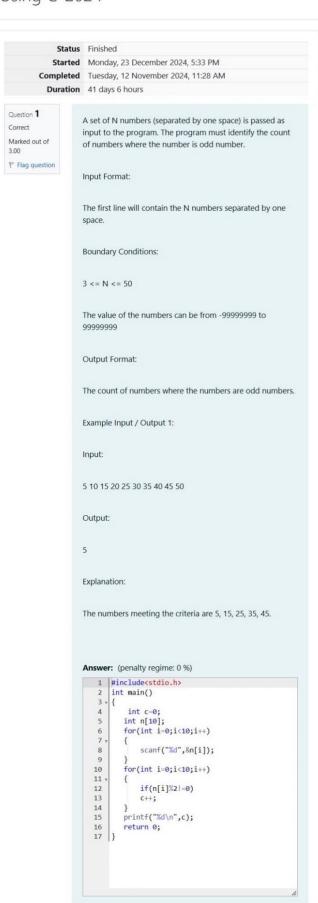
```
1 | minclude<stdio.h>
2 | int main()
3 | {
4 | int N;
5 | scanf("%d",8N)
6 | int denominati
7 | int current_su
                        int N;
scanf("%d",8N);
int denominations=0;
int current_sum=0;
while(current_sum<N) {
                                denominations++;
current_sum+=current_sum+1;
                         printf("%d\n",denominations);
return 0;
```

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

KEC-CIS

GE23131-Programming Using C-2024







Question 2

Correct

Marked out of

Given a number N, return true if and only if it is a confusing number, which satisfies the following condition:

Input	Expected	Got	
5 10 15 20 25 30 35 40 45 50	5	5	~

Question 2 Marked out of

5.00

F Flag question

Given a number N, return true if and only if it is a confusing number, which satisfies the following condition:

We can rotate digits by 180 degrees to form new digits. When 0, 1, 6, 8, 9 are rotated 180 degrees, they become 0, 1, 9, 8, 6 respectively. When 2, 3, 4, 5 and 7 are rotated 180 degrees, they become invalid. A confusing number is a number that when rotated 180 degrees becomes a different number with each digit valid.

Example 1:

6->9

Output: true

Explanation:

We get 9 after rotating 6, 9 is a valid number and 9!=6.

Example 2:

89 -> 68

Input: 89

Output: true

Explanation:

We get 68 after rotating 89, 86 is a valid number and 86!=89.

Example 3:

11 -> 11

Input: 11

Output: false

Explanation:

We get 11 after rotating 11, 11 is a valid number but the

Note: 1. 0 <= N <= 10^9

- After the rotation we can ignore leading zeros, for example if after rotation we have 0008 then this number is considered as just 8.

value remains the same, thus 11 is not a confusing number.

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
     int main()
           int n,x,y=1;
scanf("%d",&n);
while(n!=0 && y==1)
                x=n%10;
                n=n/10;
                if(x==2||x==3||x==4||x==7)
10
12
                if(y==1)
printf("true");
else
14
15
           printf("false");
return 0;
16
18 }
```

	Input	Expected	Got	
~	6	true	true	~
/	89	true	true	~
/	25	false	false	~

Question 3 Correct

A nutritionist is labeling all the best power foods in the market. Every food item arranged in a single line, will have a of food items.

The second line contains an integer, *k*, that denotes the unhealthy number.

Sample Input 0

2

Sample Output 0

Explanation 0

The following sequence of n = 2 food items:

- Item 1 has 1 macronutrients.
- 2. 1 + 2 = 3; observe that this is the max total, and having avoided having exactly k = 2 macronutrients.

Sample Input 1

2

Sample Output 1

.

Explanation 1

- 1. Cannot use item 1 because k = 1 and $sum \equiv k$ has to be avoided at any time.
 - Hence, max total is achieved by sum = 0 + 2 = 2.

Sample Case 2

Sample Input For Custom Testing

Sample Input 2

Sample Output 2

5

Explanation 2

2 + 3 = 5, is the best case for maximum nutrients.

Answer: (penalty regime: 0 %)

	Input	Expected	Got	
~	2 2	3	3	~
~	2	2	2	~
~	3	5	5	~