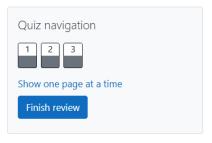
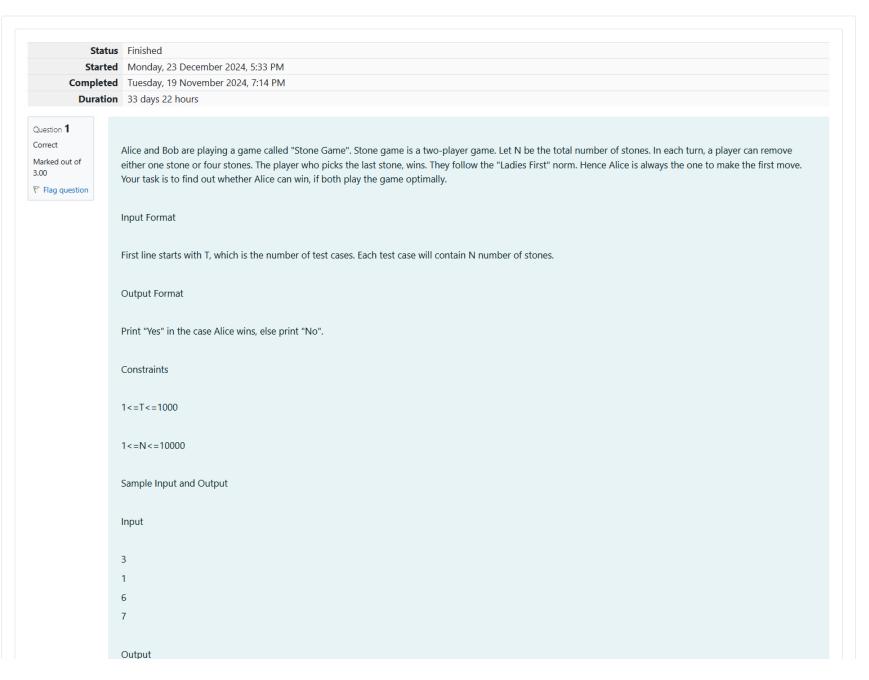
GE23131-Programming Using C-2024

MUTHURAM 240901062





MUTHURAM 240901062

Yes

Yes

No

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main()
3 ₹ {
4
      int T,i=0,n,t;
5
      scanf("%d",&T);
6
      while(i<T)
7 🔻
          scanf("%d",&n);
8
9
          t=n/4;
          if(t%2==0 && n%2==0)
10
11 v
12
          printf("No\n");
13
14
          else if(t%2==1 && n%2==1)
15 ,
          printf("No\n");
16
17
18
          else
19 •
20
          printf("Yes\n");
21
22
          i++;
23
24
       return 0;
25
          }
26
```

Question **2**Correct
Marked out of 5.00

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:

₱ Flag question

, 2, 3, 5, and 7 = 0 holes.	MUTHURAN
), 4, 6, and 9 = 1 hole.	We the total
s = 2 holes.	
Siven a number, you must determine the sum of the number of holes for all of its digits. For example, the number 819 has 3 h	oles.
Complete the program, it must must return an integer denoting the total number of holes in num.	
Constraints	
≤ num ≤ 109	
nput Format For Custom Testing	
here is one line of text containing a single integer num, the value to process.	
iample Input	
330	
iample Output	
xplanation	
add the holes count for each digit, 6, 3 and 0. Return 1 + 0 + 1 = 2.	
iample Case 1	
ample Input	
288	
ample Output	

Add the holes count for each digit, 1, 2, 8, 8. Return 0 + 0 + 2 + 2 = 4.

Answer: (penalty regime: 0 %)

Explanation

```
1 #include<stdio.h>
2
    int main()
3 ₹ {
4
       int a,b,n=0;
       scanf("%d",&a);
5
6
       while(a>0)
7 .
 8
           b=a\%10;
9
           if(b==0 || b==6 || b==9 || b==4)
10
11
               n=n+1;
12
13
           else if(b==8)
14
15
               n=n+2;
16
17
           a=a/10;
18
19
20
           printf("%d",n);
21
           return 0;
22
23
```

	Input	Expected	Got	
~	630	2	2	~
~	1288	4	4	~

Passed all tests! <

Question **3**Correct
Marked out of 7.00

Flag question

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

Explanation:

For test case 1, N=10.

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 #include<stdio.h>
2
    int main()
3 ₹ {
4
       int n,r=0;
5
       scanf("%d",&n);
 6
       while(n!=0)
7 🔻
 8
          n=n/2;
9
          r=r+1;
10
11
       printf("%d",r);
12
13
       return 0;
14 }
```

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

Passed all tests! <