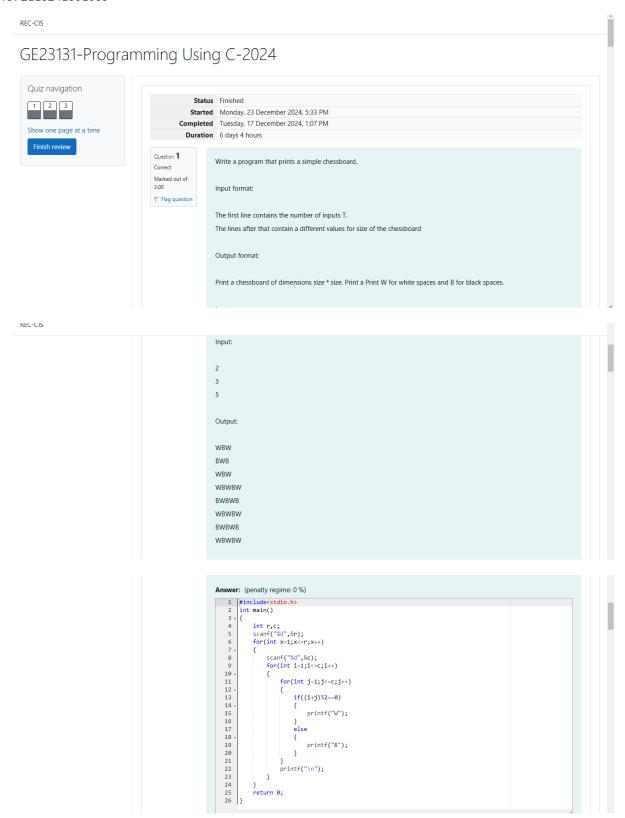
#### AIDS-A

#### ROLL NO: 2116241801035

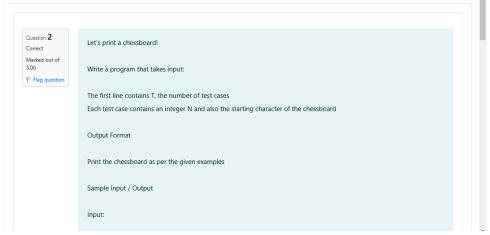




REC-CIS

## GE23131-Programming Using C-2024



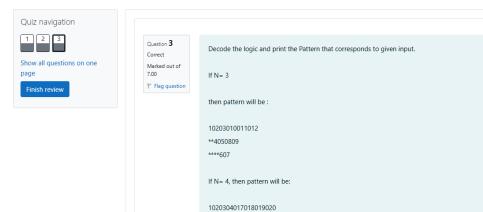


```
2
2 W
3 B
Output:
WB
BW
BWB
WBWB
WBWB
```

REC-CIS



# GE23131-Programming Using C-2024



\*\*50607014015016 \*\*\*\*809012013

\*\*\*\*\*\*10011

Constraints

2 <= N <= 100

Input Format

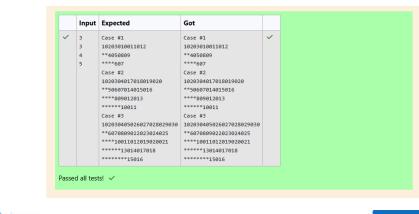
First line contains T, the number of test cases
Each test case contains a single integer N

Output

First line print Case #i where i is the test case number
In the subsequent line, print the pattern

Test Case 1

```
4
5
Output
Case #1
10203010011012
**4050809
****607
Case #2
1020304017018019020
**50607014015016
****809012013
*****10011
Case #3
102030405026027028029030
**6070809022023024025
****10011012019020021
*****13014017018
******15016
```



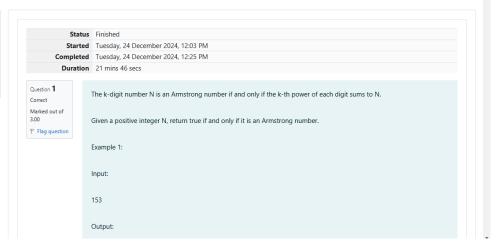
→ Previous page

Finish review

Ţ

## GE23131-Programming Using C-2024





```
true

Explanation:

153 is a 3-digit number, and 153 = 1^3 + 5^3 + 3^3.

Example 2:

Input:

123

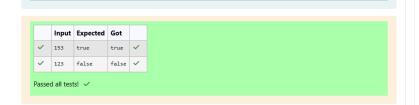
Output:

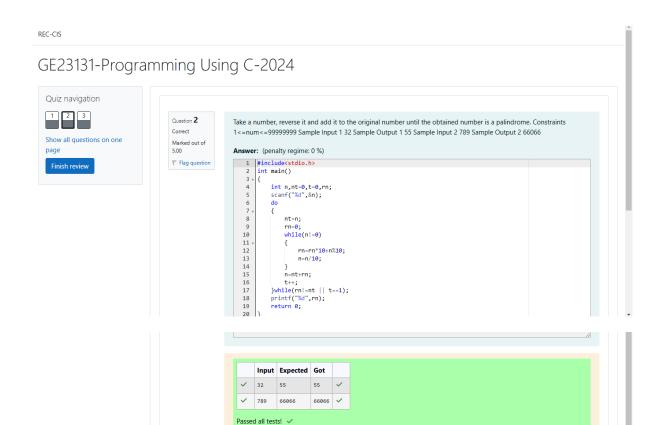
false

Explanation:

123 is a 3-digit number, and 123 != 1^3 + 2^3 + 3^3 = 36.
```

```
6
7
7
8
8 b=num;
9
10*
11 n=n/10;
d+;
12 d+;
13
14 while(b!=0)
{
    for(int i=1;i<=d;i++)
{
        s=b%10;
        b=b/10;
        a=a+(pow(s,d));
    }
21
22 if(num==a)
{
    printf("true");
    }
25 else
27
28    printf("false");
}
    return 0;
}</pre>
```

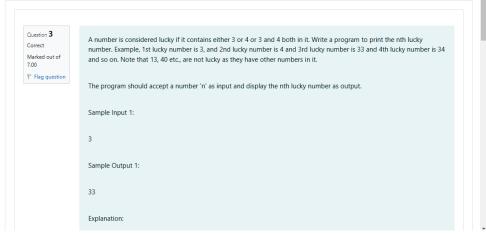




REC-CIS

# GE23131-Programming Using C-2024





REC-CIS