***Week 05 01 coding***

***Program 1***

*Write a program that prints a simple chessboard.*

*Input format:*

*The first line contains the number of inputs T.*

*The lines after that contain a different values for size of the chessboard*

*Output format:*

*Print a chessboard of dimensions size \* size. Print a Print W for white spaces and B for black spaces.*

*Input:*

*2*

*3*

*5*

*Output:*

*WBW*

*BWB*

*WBW*

*WBWBW*

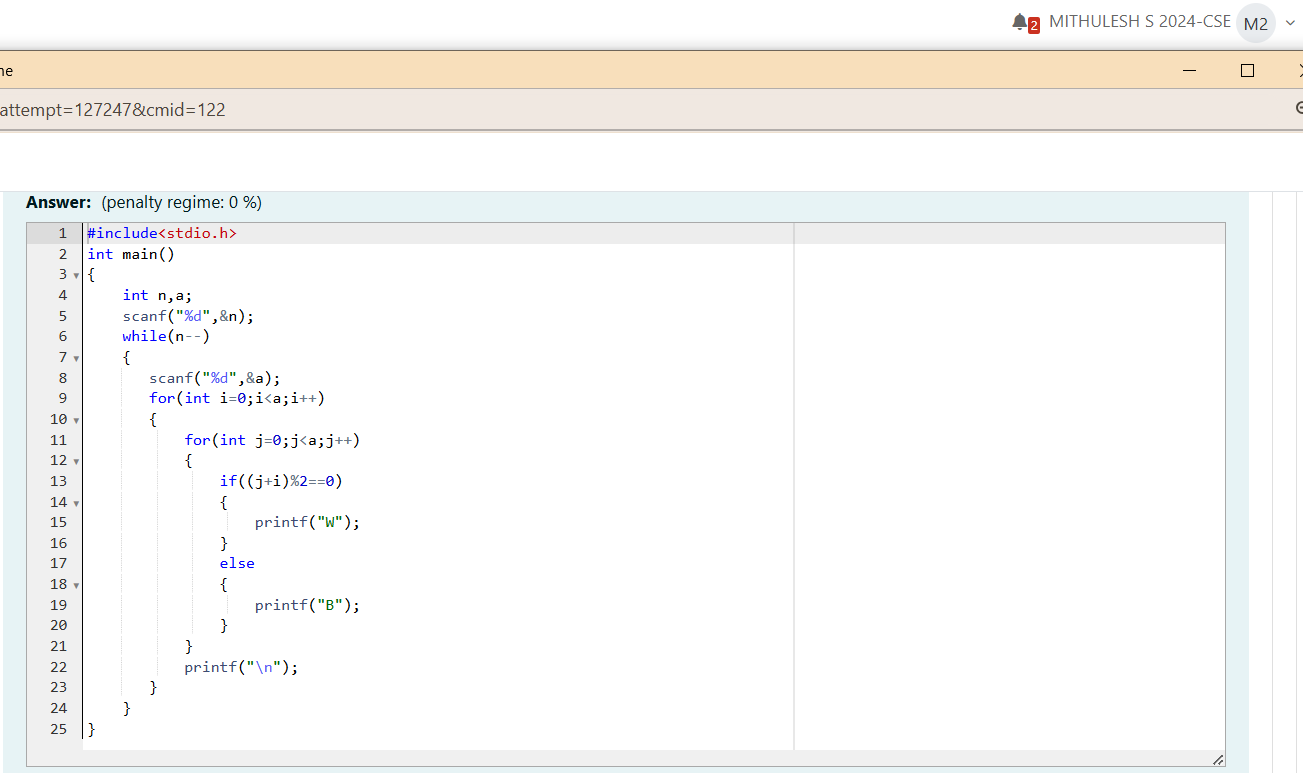
*BWBWB*

*WBWBW*

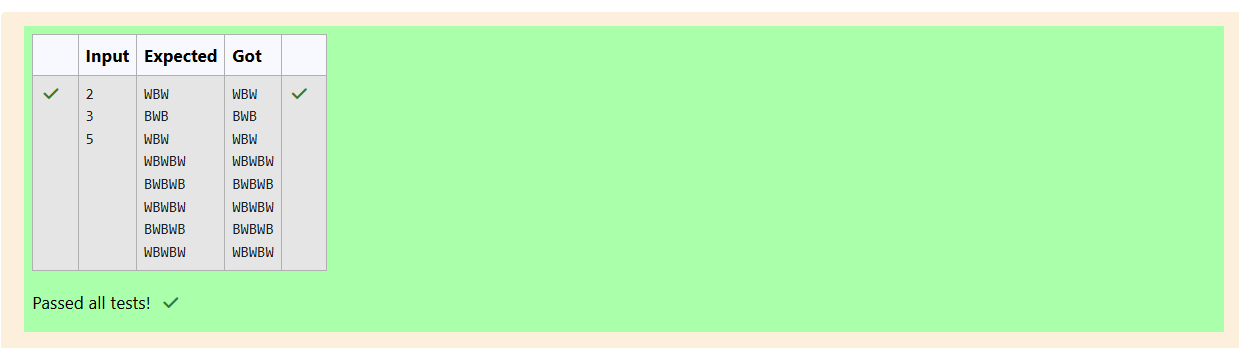
*BWBWB*

*WBWBW*

***Coding***

**

***Output***

**

***Program 2***

*Let's print a chessboard!*

*Write a program that takes input:*

*The first line contains T, the number of test cases*

*Each test case contains an integer N and also the starting character of the chessboard*

*Output Format*

*Print the chessboard as per the given examples*

*Sample Input / Output*

*Input:*

*2*

*3B*

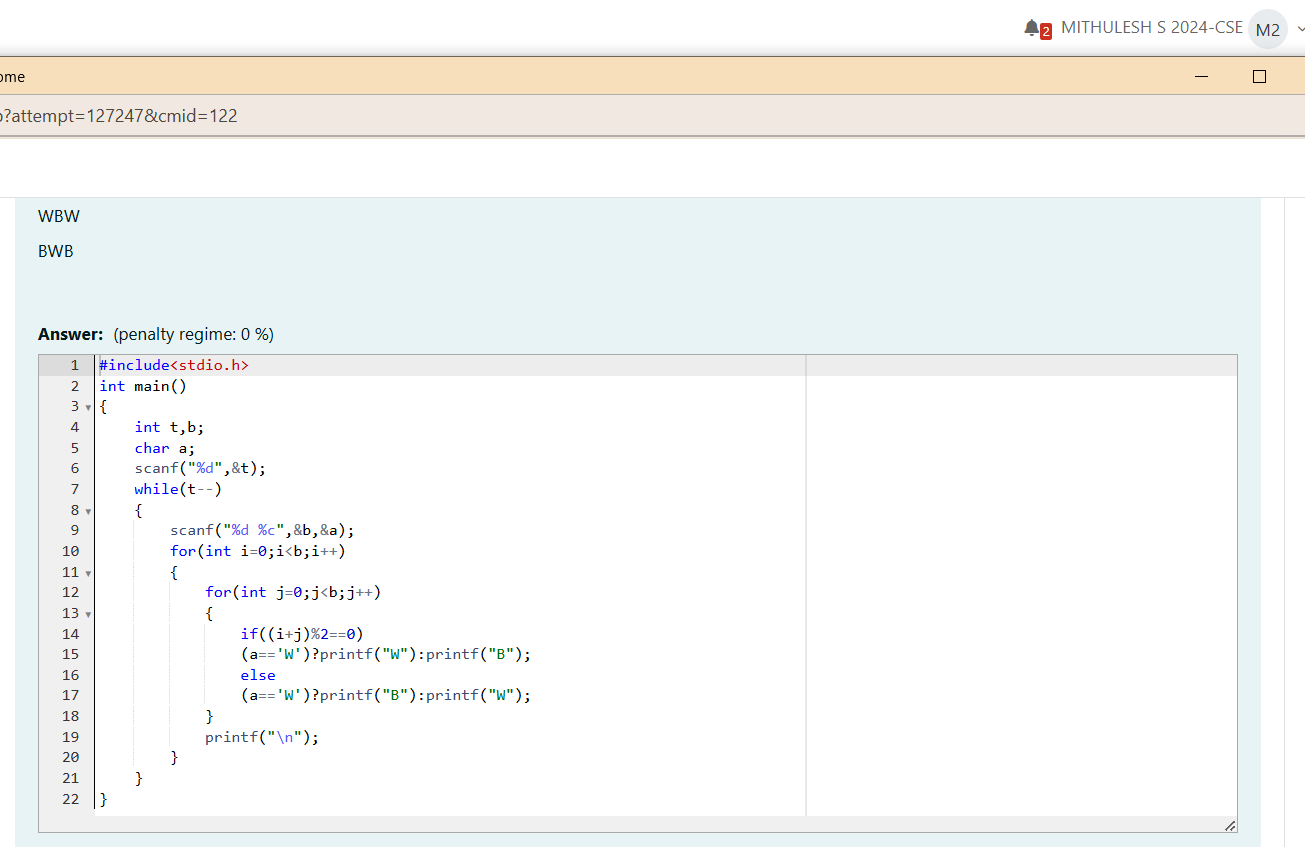
*Output:*

*BWB*

*WBW*

*BWB*

***Coding***

**

***Output***

**

***Program 3***

*Decode the logic and print the Pattern that corresponds to given input.*

*then pattern will be :*

*10203010011012*

*If N: 4, then pattern will be:*

*1020304017018019020*

*\*\*\*\*809012013*

*Constraints*

*-N 100*

*Constraints*

*= 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 I*

*3*

*3*

*4*

*5*

*Output*

*Case*

*10203010011012*

*Case #2*

*1020304017018019020*

*"50607014015016*

*10011*

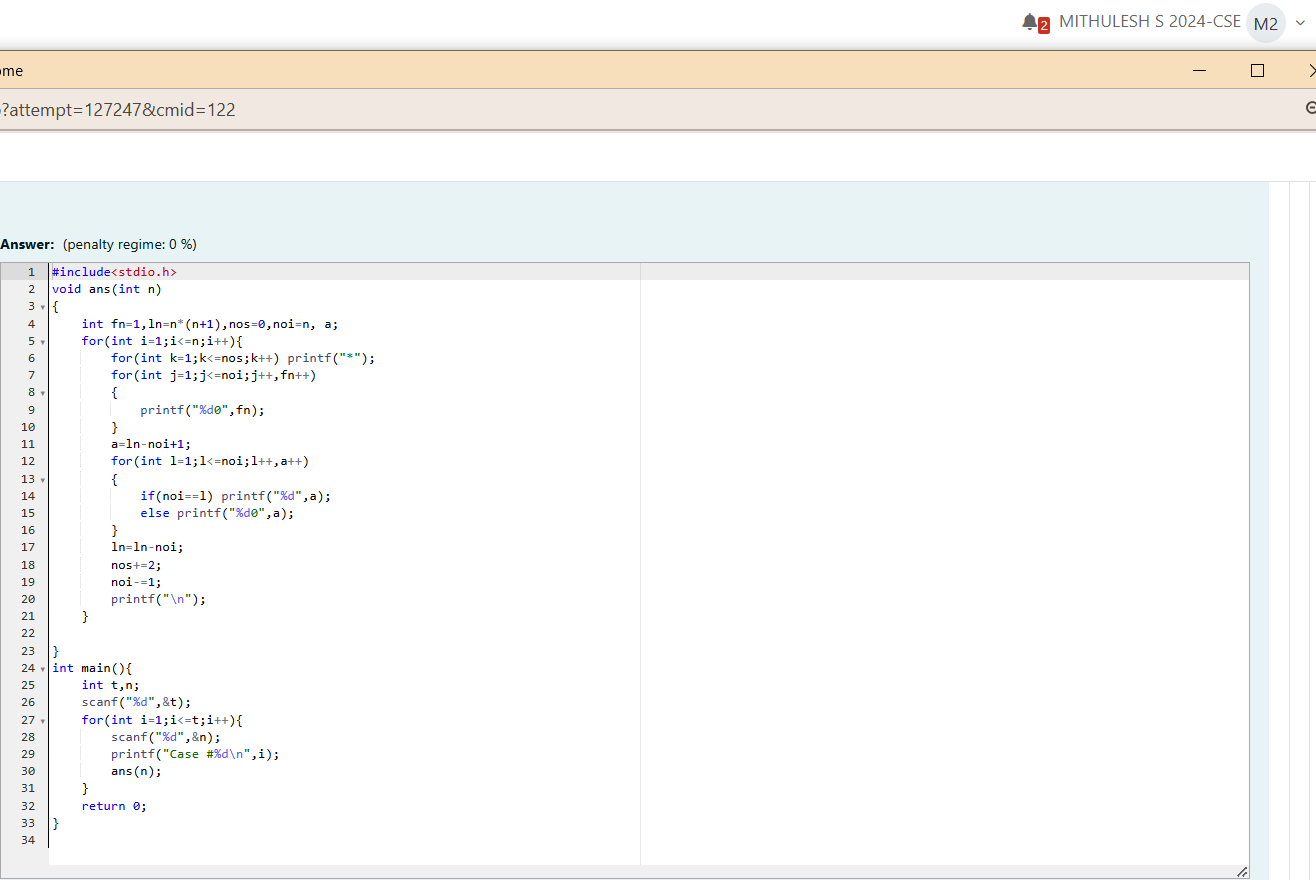
*Case #3*

*102030405026027028029030*

*13014017018*

*15016*

***Coding***

**

***Output***

**

***Week 05 02 Coding***

***Program 1***

*The k-digit number N is an Armstrong number if and only if the k-th power of each digit sums to N.*

*Given a positive integer N, return true if and only if it is an Armstrong number.*

*Example I:*

*Input:*

*153*

*Output:*

*true*

*Explanation:*

*153 is a 3-digit number, and 153*

*Example 2:*

*Input:*

*123*

*Output:*

*false*

*Explanation:*

*123 is a 3-di it number and 123*

*= IA3 + +3 + 3A3.*

*- + 2A3 + 30 = 36.*

*Example 3:*

*Input:*

*1634*

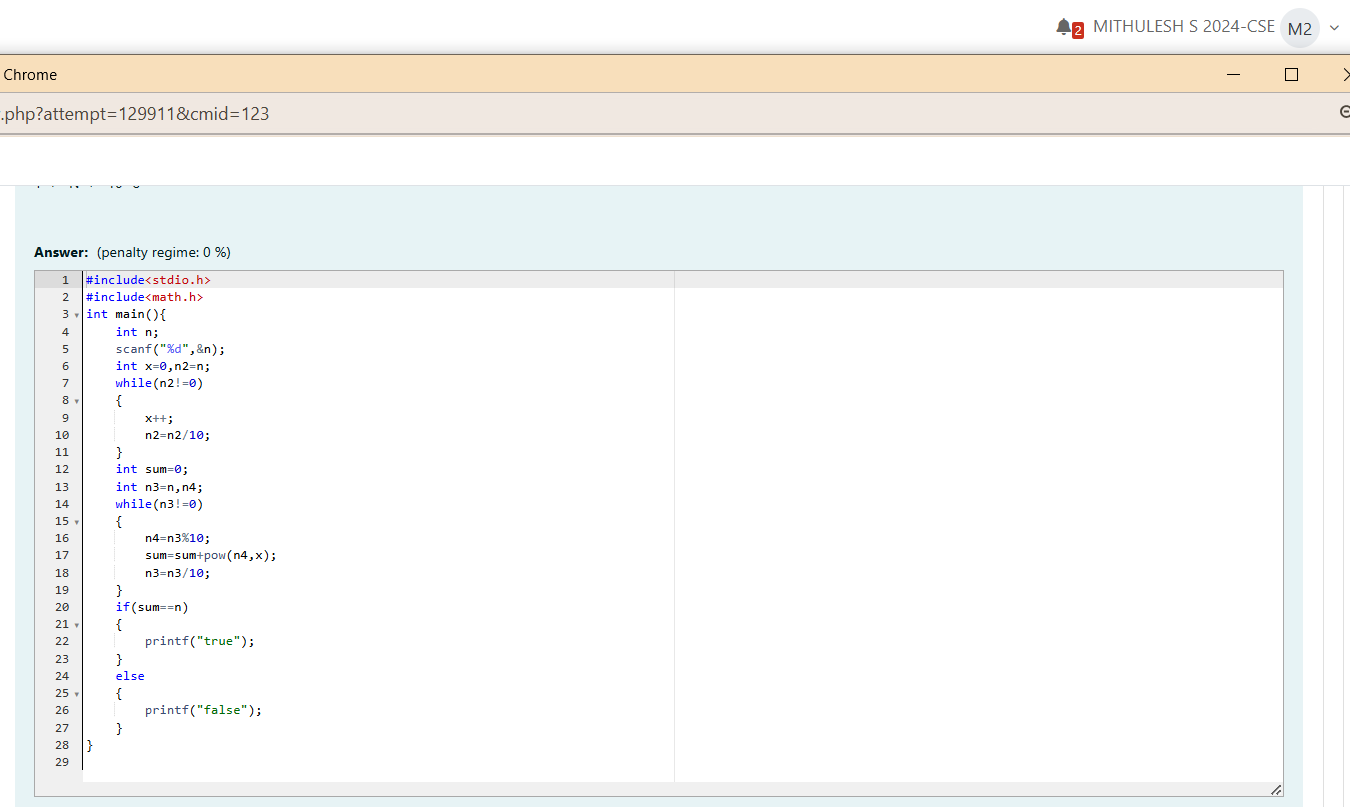
*Output:*

*true*

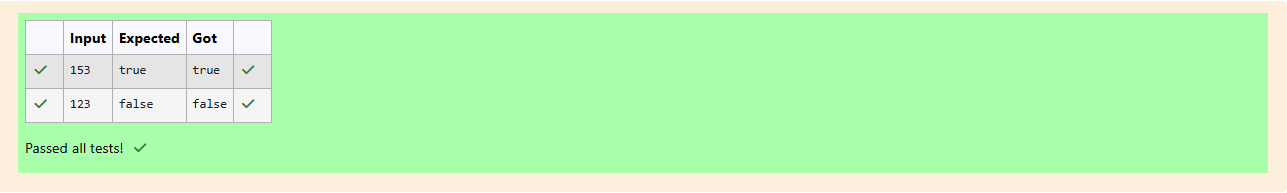
*Note:*

*= N IOA8*

***Coding***

**

***Output***

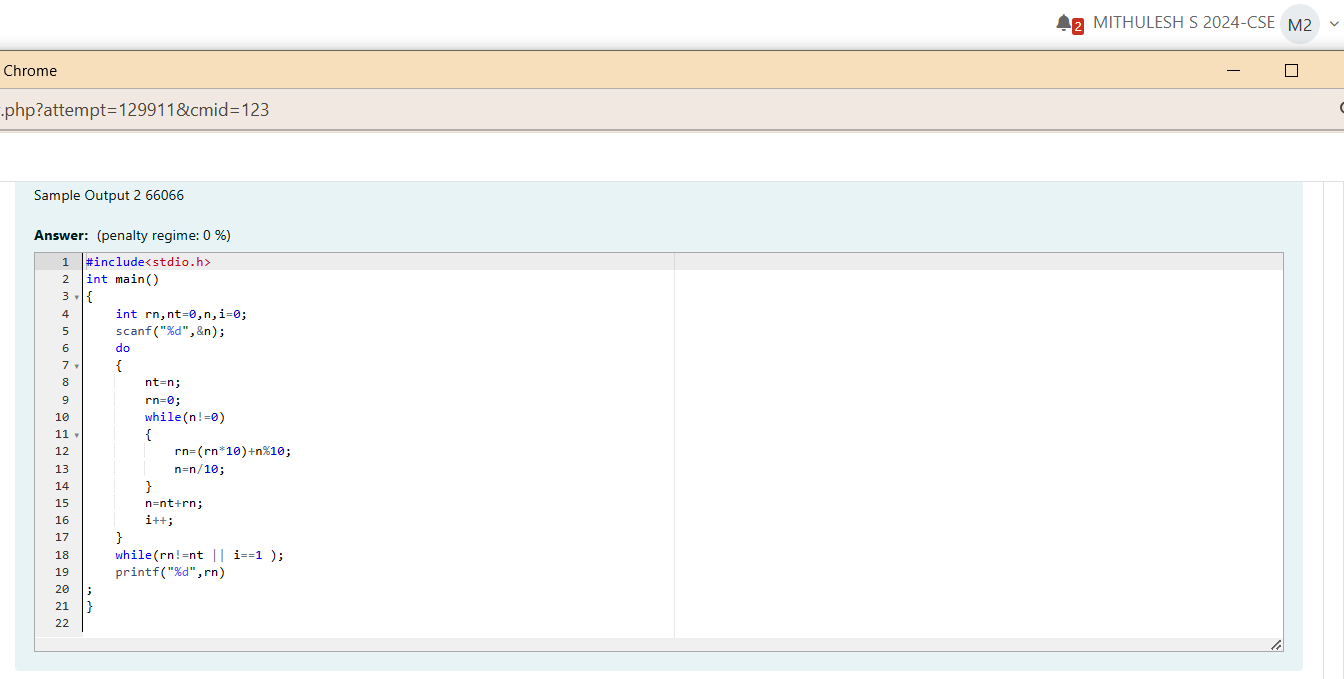
**

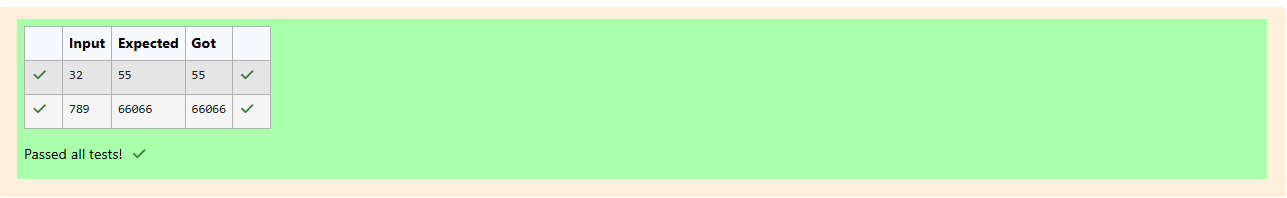
***Program 2***

*Take a number, reverse it and add it to the original number until the obtained number is a palindrome. Constraints 1 < =num<=99999999 Sample Input 1 32 Sample Output 1 55 Sample Input 2 789*

*Sample Output 2 66066*

***Coding***

**

***Output*** **

***Program 3***

*A number is considered lucky if it contains either 3 or 4 or 3 and 4 both in it. Write a program to print the nth lucky number. Example, 1st lucky number is 3, and 2nd lucky number is 4 and 3rd lucky*

*number is 33 and 4th lucky number is 34 and so on. Note that 13, 40 etc„ are not lucky as they have other numbers in it.*

*The program should accept a number •n' as input and display the nth lucky number as output.*

*Sample Input I:*

*3*

*Sample Output 1:*

*33*

*Explanation:*

*Here the lucky numbers are 3, 4, 33, 34., and the 3rd lucky number is 33.*

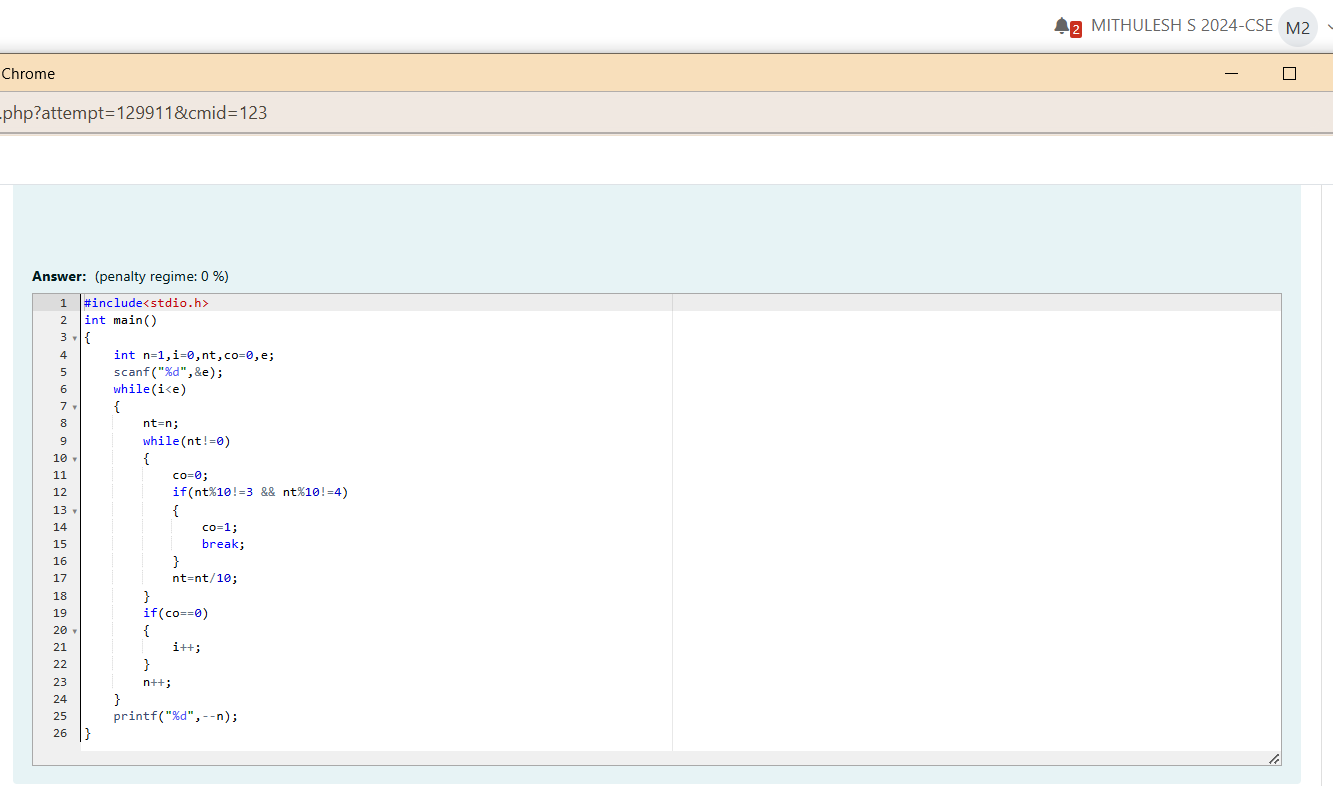
*Sample Input 2:*

*34*

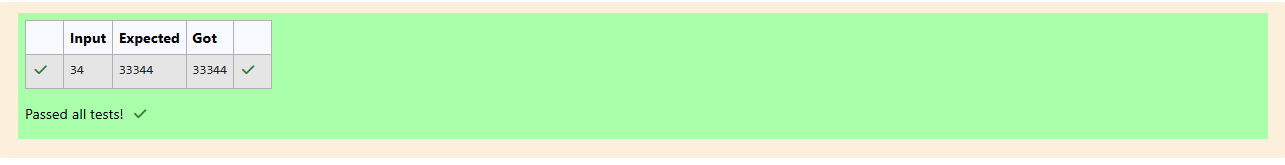
*Sample Output 2:*

*33344*

***Coding***

**

***Output***

**