Week 02

**Question 1:**

Many people think about their height in feet and inches, even in some countries that primarily use the metric system. Write a program that reads a number of feet from the user, followed by a number of inches. Once these values are read, your program should compute and display the equivalent number of centimeters.

**Input Format:**

First line, read the number of feet. Second line, read the number of inches.

Output Format In one line print the height in centimeters.

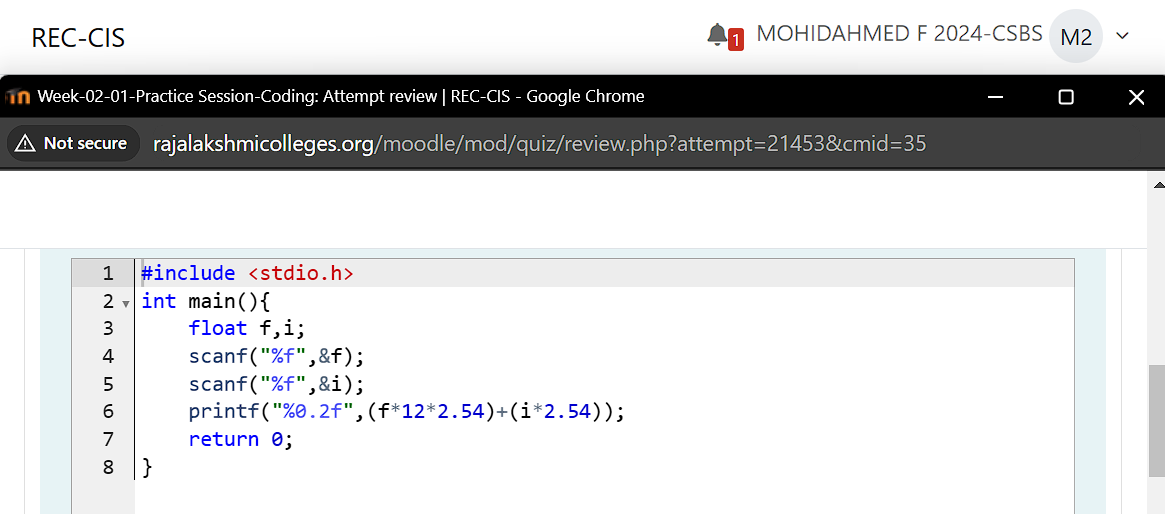
**Sample Input 1:**

56

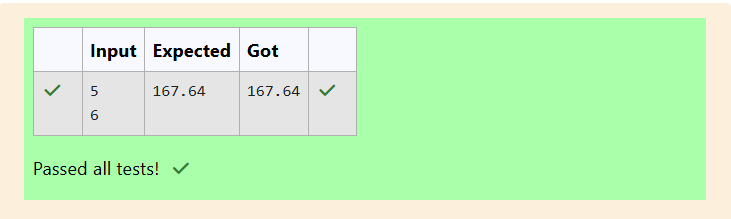
**Sample Output 1:**

167.64

**Program:**

****

**Output:**

****

**Question 2:**

Create a program that reads two integers, a and b, from the user. Your program should compute and display: The sum of a and b. The difference when b is subtracted from a. The product of a and b The quotient when a is divided by b. The remainder when a is divided by b

Input Format: First line, read the first number. Second line, read the second number.

Output Format: First line, print the sum of a and b, Second line, print the difference when b is subtracted from a. Third line, print the product of a and b. Fourth line, print the quotient when a is divided by b. Fifth line, print the remainder when a is divided by b,

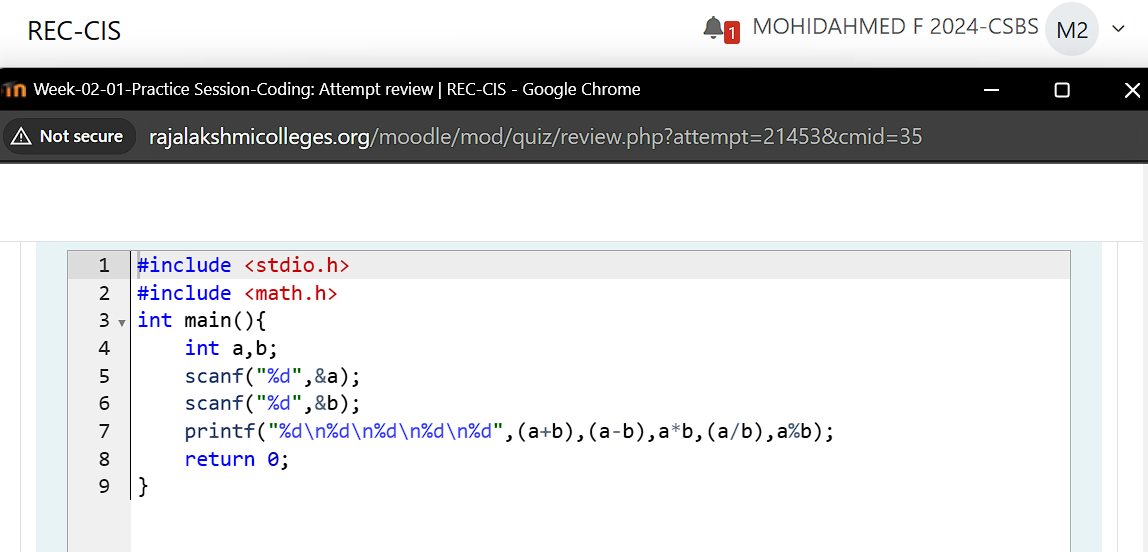
**Sample Input:**

1 100 6

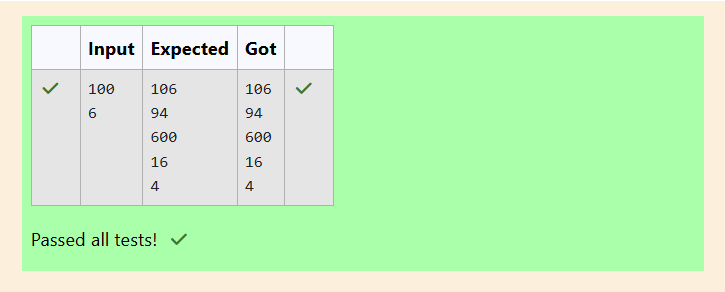
**Sample Output:**

106 94 600 16 4

**Program:**

****

**Output:**

****

**Question 3:**

**A bakery sells loaves of bread for $3.49 each. Day old bread is discounted by 60 percent. Write a program that begins by reading the number of loaves of day old bread being purchased from the user. Then your program should display the regular price for the bread, the discount because it is a day old, and the total price. Each of these amounts should be displayed on its own line with an appropriate label. All of the values should be displayed using two decimal places.**

**Input Format: Read the number of day old loaves.**

**Output Format: First line, print Regular price,Second line, print Discount,Third line, print Total.**

**Sample Input 1**

**10**

**Sample Output 1**

**Regular price: 34.90**

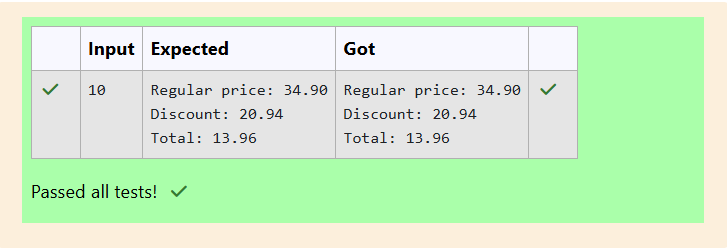
**Discount: 20.94**

**Total: 13.96**

**Program:**

****

**Output:**

****

**Question 4:**

**Goki recently had a breakup, so he wants to have some more friends in his life. Goki has N people who he can be friends with, so he decides to choose among them according to their skills set Yi(1<<i<=n). He wants atleast X skills in his friends. Help Goki find his friends.**

**INPUT**

**First line contains a single integer X - denoting the minimum skill required to be Goki's friend. Next line contains one integer Y - denoting the skill of the person**

**OUTPUT**

**Print if he can be friend with Goki. 'YES' (without quotes) if he can be friends with Goki else 'NO' (without quotes).**

**Sample Input 1:**

**100 110**

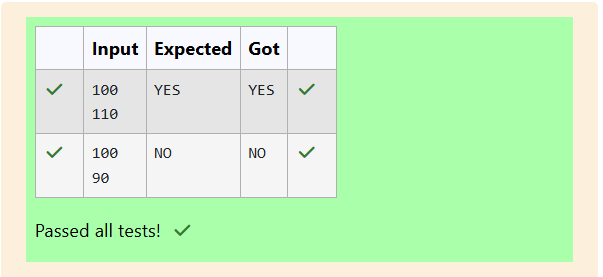
**Sample Output 1:**

**YES**

**Program:**

****

**Output:**

****

**Question 5:**

**Before the outbreak of corona virus to the world, a meeting happened in a room in Wuhan. A person who attended that meeting had COVID- 19 and no one in the room knew about it! So everyone started shaking hands with everyone else in the room as a gesture of respect and after meeting unfortunately everyone got infected! Given the fact that any two persons shake hand exactly once, Can you tell the total count of handshakes happened in that meeting? Say no to shakehands. Regularly wash your hands. Stay Safe.**

**Input Format**

**Read an integer N,the total number of people attended that meeting.**

**Output Format**

**Print the number of handshakes.**

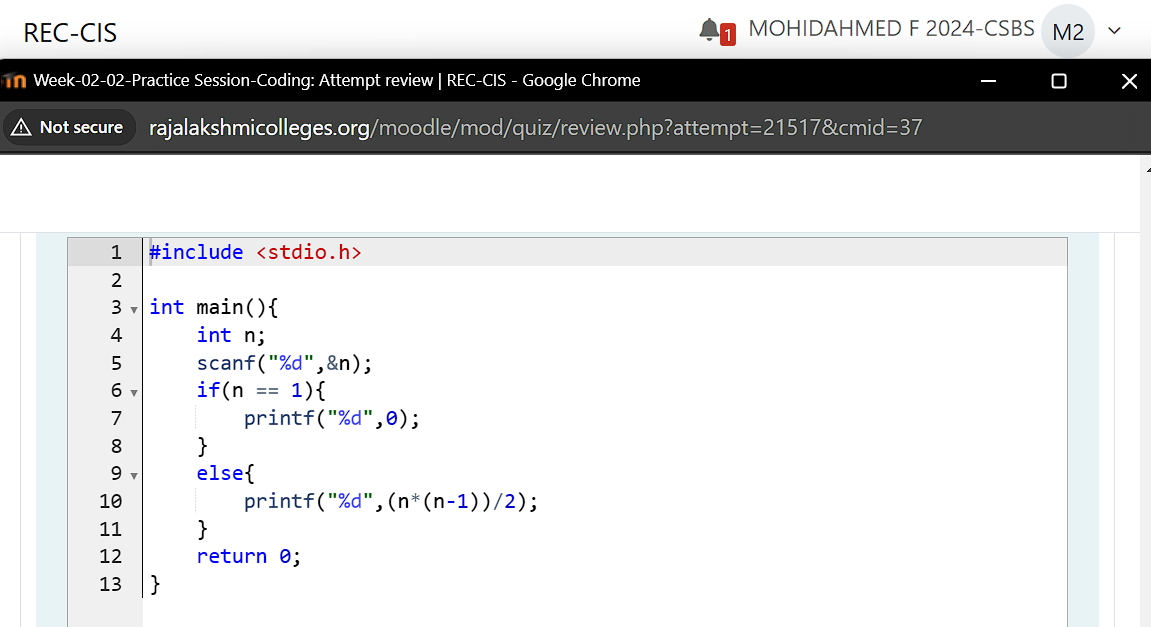
**Sample Input 1**

**1**

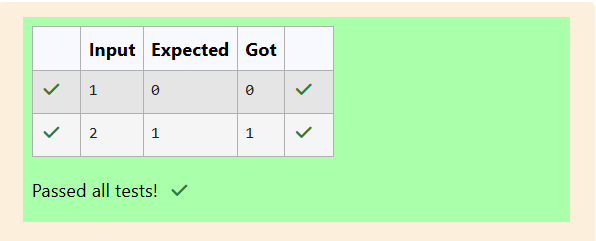
**Sample Output 1**

**0**

**Program:**

****

**Output:**

****

**Question 6:**

**In our school days, all of us have enjoyed the Games period. Raghav loves to play cricket and is Captain of his team. He always wanted to win all cricket matches. But only one last Games period is left in school now. After that he will pass out from school. So, this match is very important to him. He does not want to lose it. So he has done a lot of planning to make sure his teams wins. He is worried about only one opponent - Jatin, who is very good batsman. Raghav has figured out 3 types of bowling techniques, that could be most beneficial for dismissing Jatin. He has given points to each of the 3 techniques. You need to tell him which is the maximum point value, so that Raghav can select best technique. 3 numbers are given in input. Output the maximum of these numbers.**

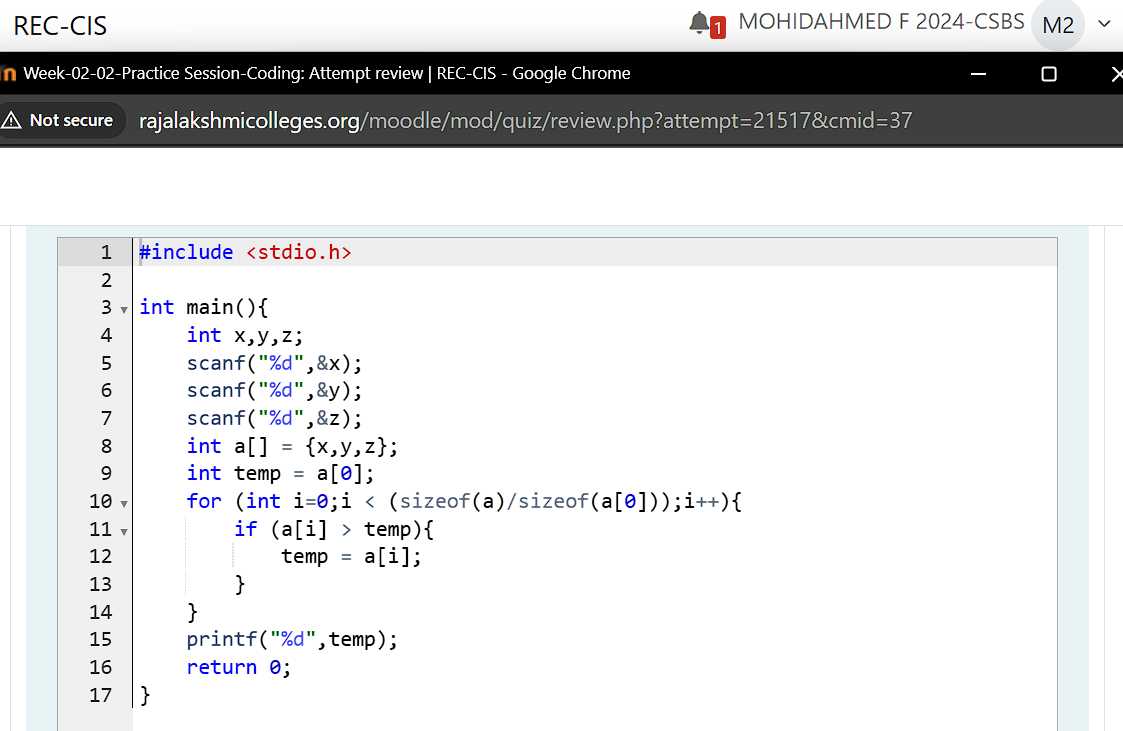
**Input format: Three space separated integers.**

**Output format: Maximum integer value**

**Sample Input: 8 6 1**

**Sample Output: 8**

**Program:**

****

**Output:**

****

**Week 02 assessments**

**Question 1:**

**Each Sunday, a newspaper agency sells copies of a certain newspaper for Rs. per copy. The cost to the agency of each newspaper is Rs. . The agency pays a fixed cost for storage, delivery, and so on of Rs. 100 per Sunday. The newspaper agency wants to calculate the profit obtained on Sundays. Can you please help them out by writing a C program to compute the profit given , , and .**

**Input Format:**

**Input consists of 3 integers: , , and . is the number of copies sold, is the cost per copy, and is the cost the agency spends per copy.**

**Output Format:**

**Refer Sample Input and Output for exact formatting specifications.**

**Sample Input and Output:**

**Input**

**1000**

**2**

**1**

**Output**

**900**

**For example:**

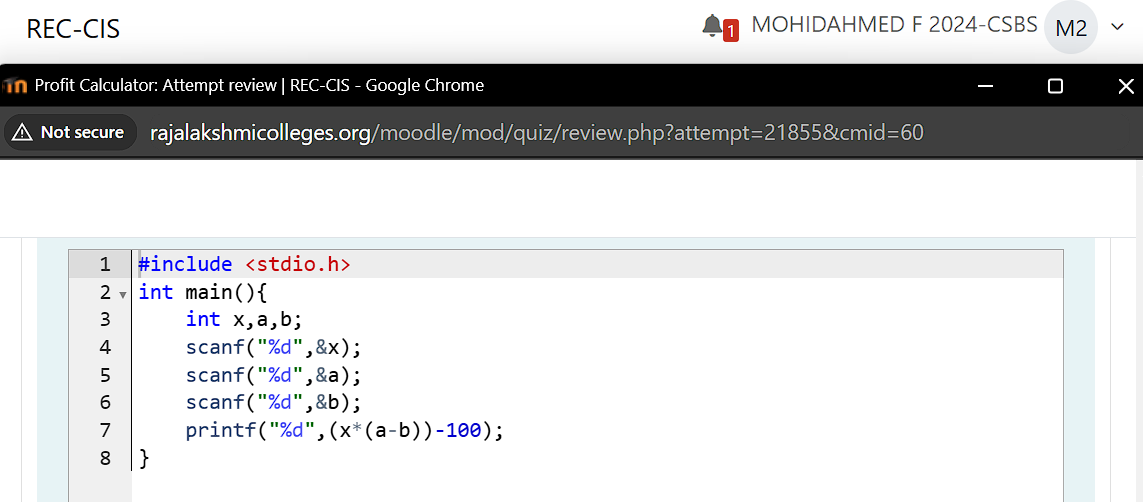
**Input | Result**

**1000 900**

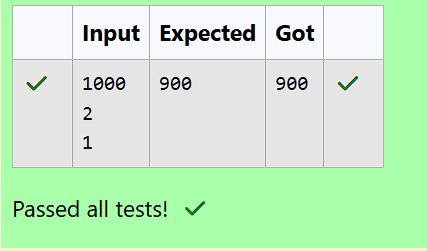
**2**

**1**

**Program:**

****

**Output:**

****

**Question 2:**

**Baba is very kind to beggars and every day Baba donates half of the amount he has whenever a beggar requests him. The money left in Baba's hand is passed as the input and the number of beggars who received the alms are passed as the input.**

**The program must print the money Baba had in the beginning of the day.**

**Input Format:**

**The first line denotes the value of .**

**The second line denotes the value of .**

**Output Format:**

**The first line denotes the value of money with Baba in the beginning of the day.**

**Example Input/Output:**

**Input:**

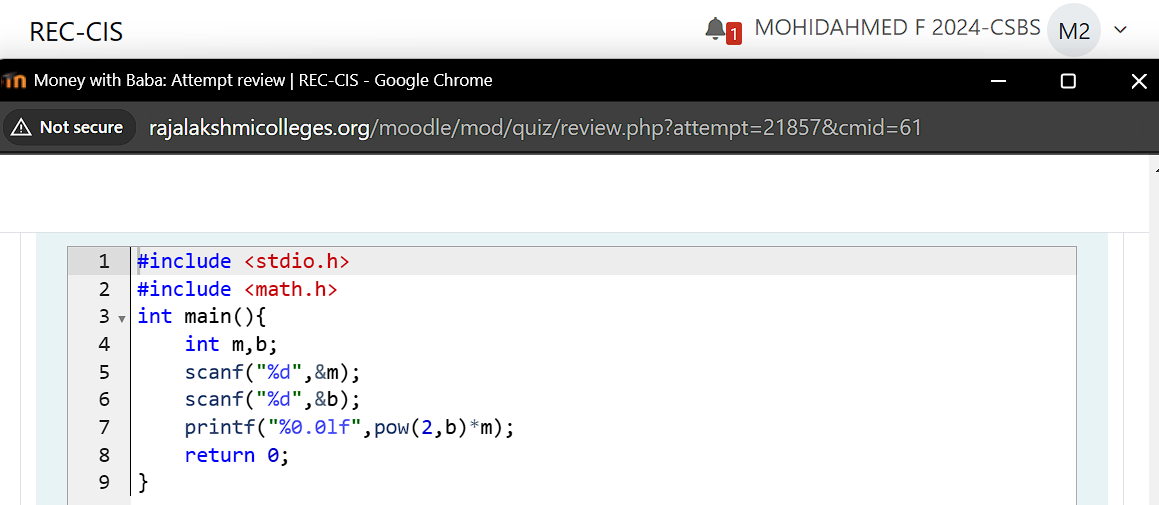
**100**

**2**

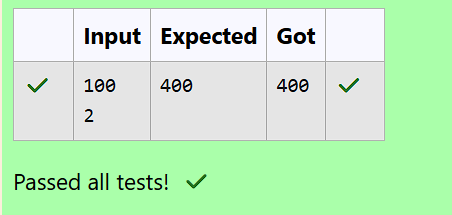
**Output:**

**400**

**Program:**

****

**Output:**

****

**Question 3:**

**The CEO of company ABC Inc wanted to encourage the employees coming on time to the office. So he announced that for every consecutive day an employee comes on time in a week (starting from Monday to Saturday), he will be awarded Rs. 200 more than the previous day as "Punctuality Incentive."**

**The incentive for the starting day (i.e., Monday) is passed as the input to the program.**

**The number of days an employee came on time consecutively starting from Monday is also passed as the input.**

**The program must calculate and print the "Punctuality Incentive" of the employee.**

**Input Format:**

**The first line denotes the value of .**

**The second line denotes the value of .**

**Output Format:**

**The first line denotes the value of .**

**Example Input/Output:**

**Input:**

**500**

**3**

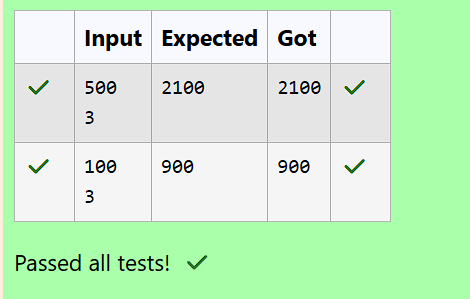
**Output:**

**2100**

**Program:**

****

**Output:**

****

**Question 4:**

**Bajan Lal distributes C chocolates to school N students every Friday. The C chocolates are distributed among N students and the remaining chocolates R are given back to Bajan Lal.**

**As an example if C=100 and N=40, each student receives 2 chocolates and the balance 100-40\*2 = 20 is given back.**

**If C=205 and N=20, then each student receives 10 chocolates and the balance 205-20\*10 = 5 is given back.**

**Help the school to calculate the chocolates to be given back when C and N are passed as input.**

**Input Format:**

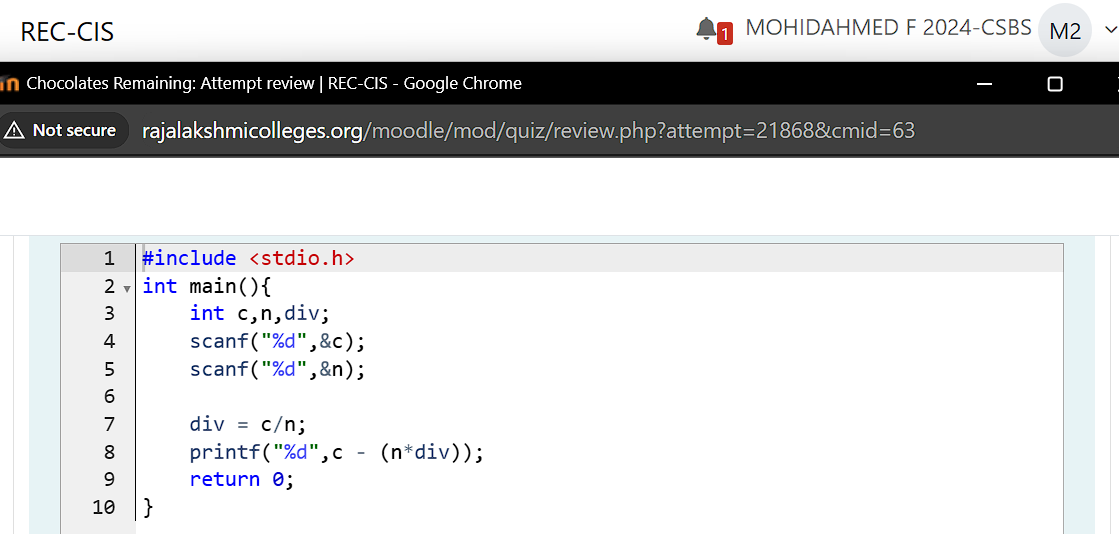
**The first line denotes C**

**The second line denotes N**

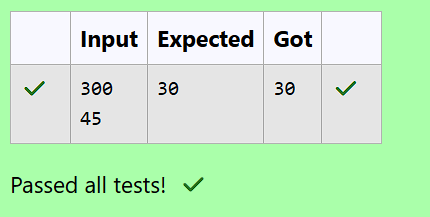
**Output Format:**

**The first line denotes R - the number of chocolates to be given back.**

**Program:**

****

**Output:**

****

**Question 5:**

**The general format of if statement is**

**if (condition) {**

**statement-1;**

**statement-2;**

**.**

**.**

**.**

**statement-n;**

**}**

**The if construct is a selective statement, the statements within the block are executed only once when the condition evaluates to true, otherwise the control goes to the first statement after the if construct.**

**If only one statement is presented in the if construct then there is no need to specify the braces { }, i.e., if braces are not specified for the if construct, by default the next immediate statement is the only statement considered for the if construct.**

**Below code prints the number only when it is divisible by 3:**

**#include <stdio.h>**

**int main()**

**{**

**int num;**

**printf("Enter a number: ");**

**scanf("%d", &num);**

**if (num % 3 == 0)**

**{**

**printf("Given number %d is divisible by 3", num);**

**}**

**return 0;**

**}**

**In the above code, num % 3 == 0 is the condition, which verifies whether the number is divisible by 3. Only the condition returns 1 (true) then the control enters into the if-block and executes the statement.**

**Fill in the missing code in the below program to check whether the given number is divisible by 3 or not.**

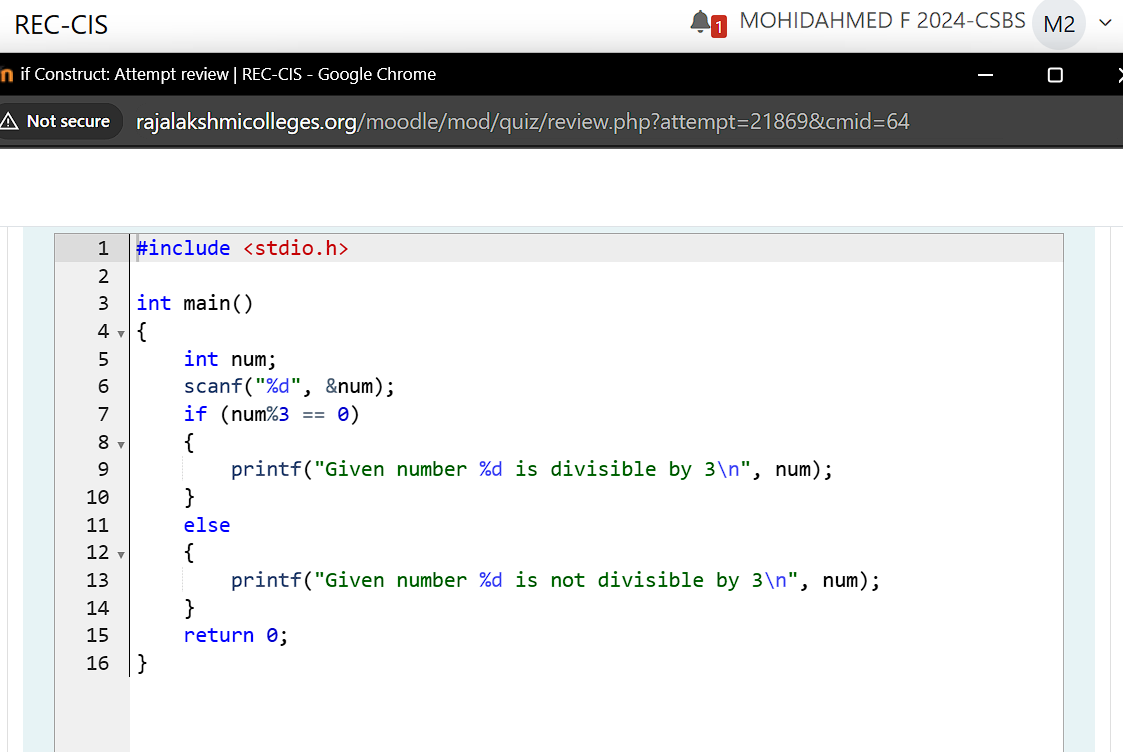
**For example:**

**| Input | Result |**

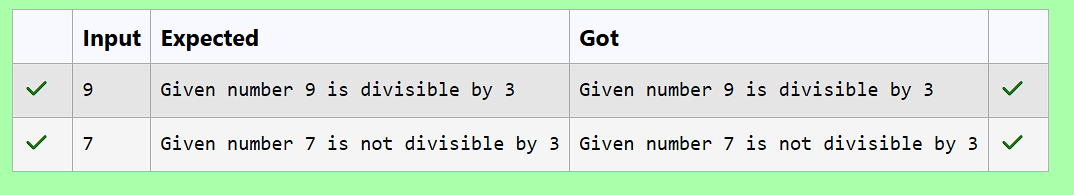
**| 9 | Given number 9 is divisible by 3 |**

**| 7 | Given number 7 is not divisible by 3 |**

**Program:**

****

**Output:**

****

**Question 6:**

**The if statement tells a program to execute a certain section of code only if a particular test evaluates to true. if (expression) {statement}.**

**Below is a sample code which uses if statement:**

**int distinction\_marks = 75;**

**if (marks > distinction\_marks)**

**{**

**printf("User secured distinction.\n");**

**}**

**An if statement will execute its block only when condition evaluates to 1 (true).**

**We can also conditionally execute another block when the condition evaluates to 0 (false) using the else construct. The else construct must be attached to an if, hence together they are referred to as if-else construct.**

**The if-else statement provides two different paths of execution depending on the result of the condition.**

**Below is the general syntax for the if-else statement:**

**if (expression)**

**{**

**statement-1;**

**}**

**else**

**{**

**statement-2;**

**}**

**Below is an example with code:**

**int distinction\_marks = 75;**

**if (marks > distinction\_marks)**

**{**

**printf("User secured distinction.\n");**

**}**

**else**

**{**

**printf("User did not secure distinction.\n");**

**}**

**Fill in the missing code in the below program to check whether the user secured distinction or not.**

**For example:**

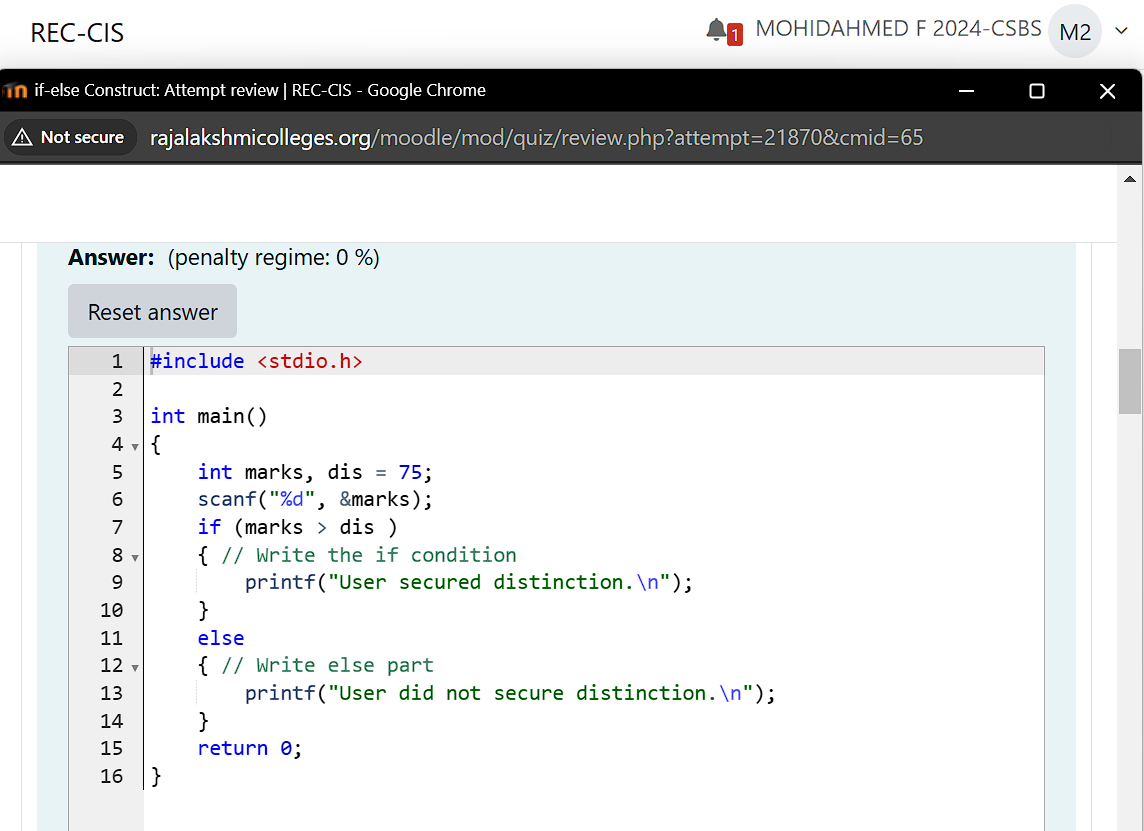
**| Input | Result |**

**|---|---|**

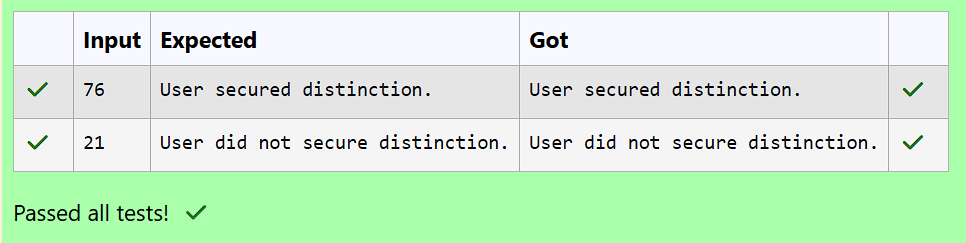
**| 76 | User secured distinction. |**

**| 21 | User did not secure distinction. |**

**Program:**

****

**Output:**

****

**Question 7:**

**Write code which uses an if-else statement to check whether a given account balance is greater or lesser than the minimum balance.**

**Use the if-else statement and print "Balance is low" if the balance is less than 1000, otherwise print "Sufficient balance".**

**For example, if the user gives the input as 1500:**

**1500**

**then the program should print the result as:**

**Sufficient balance**

**Similarly, if the input is given as 700 then print**

**Balance is low**

**[Hint: Make sure to read the input as a float value.]**

**For example:**

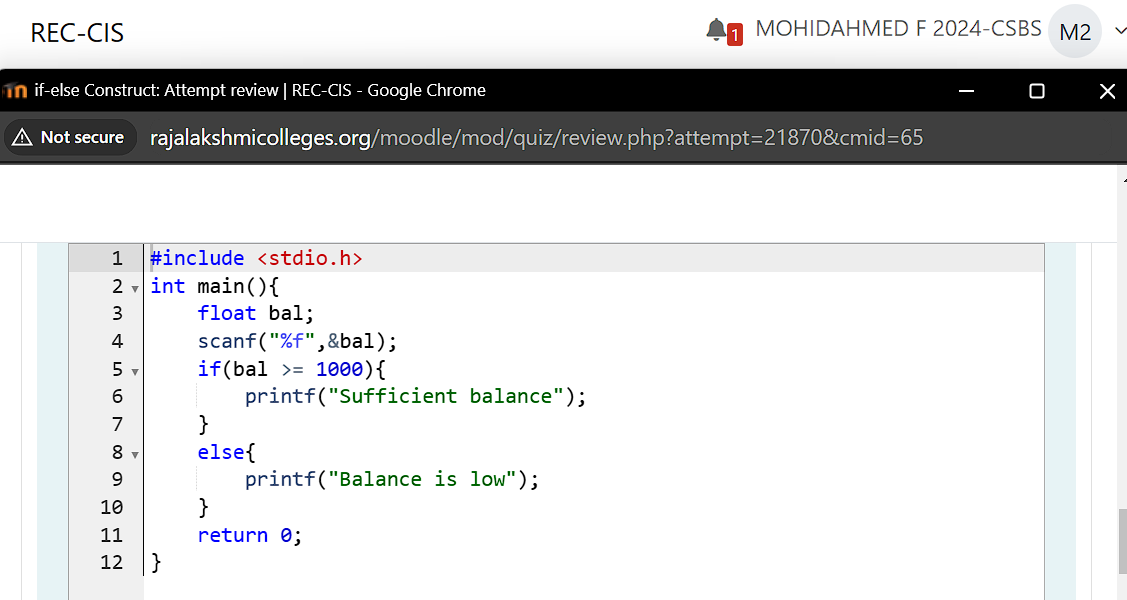
**| Input | Result |**

**|---|---|**

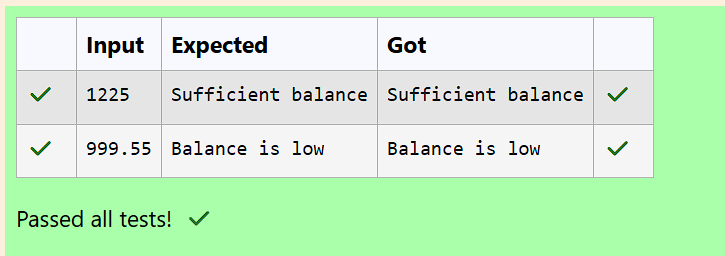
**| 1225 | Sufficient balance |**

**| 999.55 | Balance is low |**

**Program:**

****

**Output:**

****

**Question 8:**

**Fill in the missing code in the below program to check whether the student secured first class or not.**

**Note-1: Read 6 subjects marks, find total and percentage, then print the student secured first class or not.**

**Note-2: If percentage is greater than or equal to 60 then print student secured first class and the percentage.**

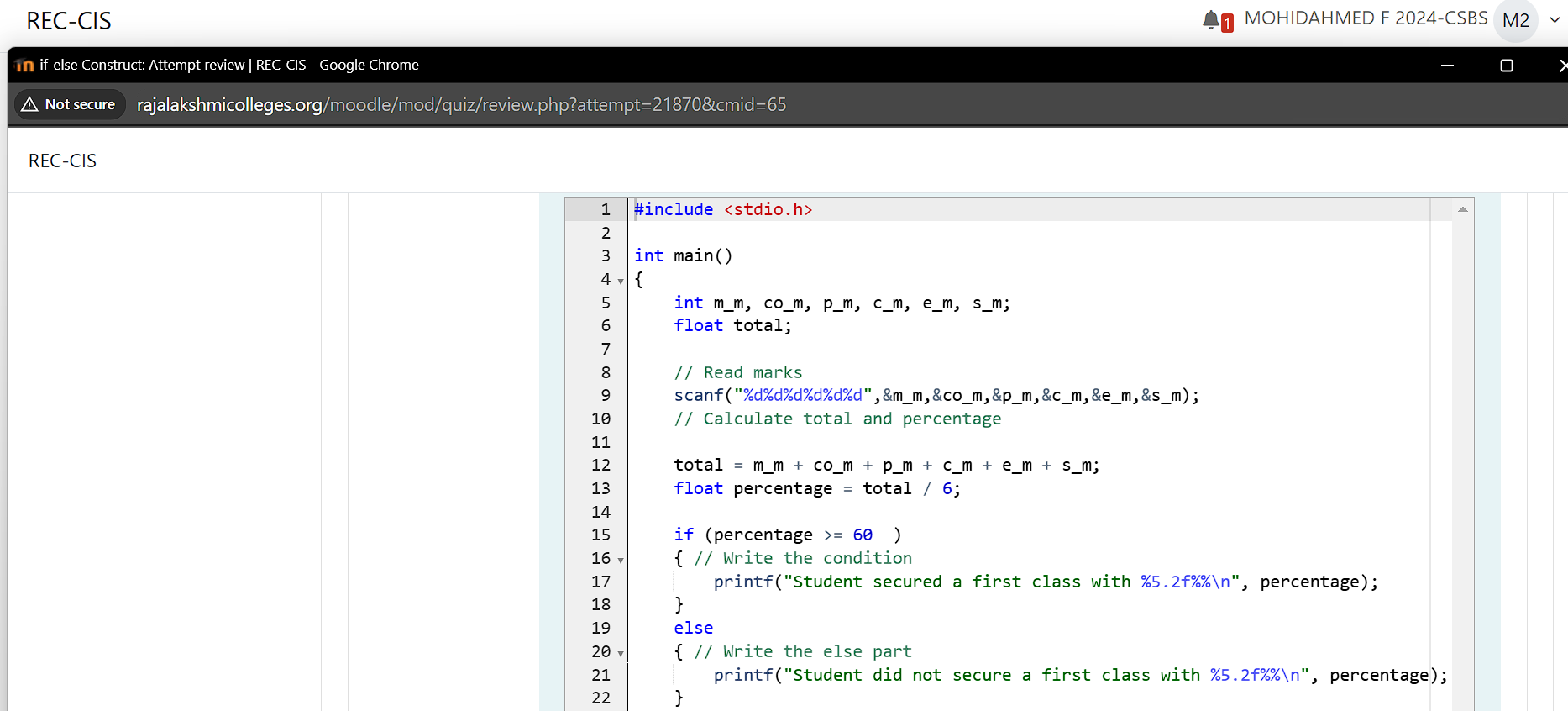
**For example:**

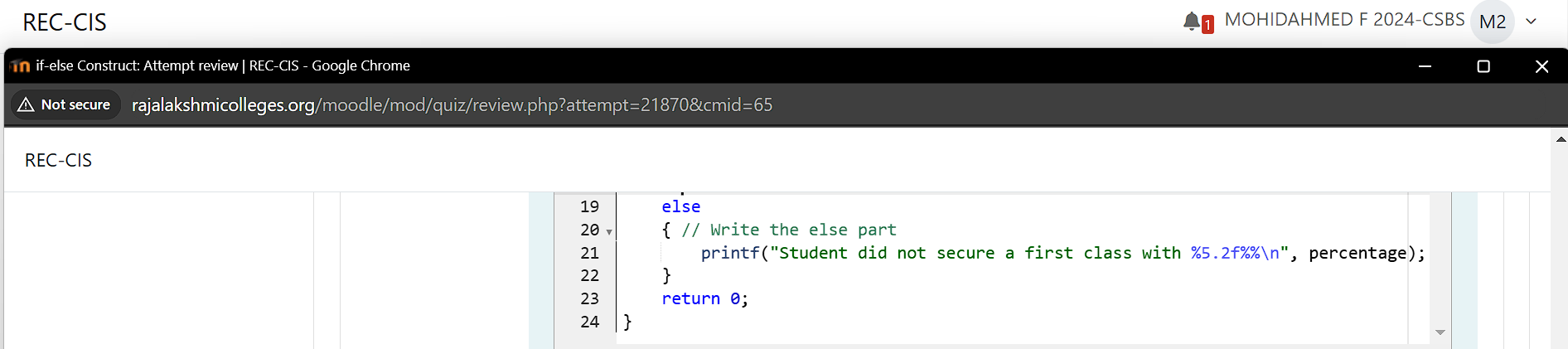
**| Input | Result |**

**| 45 67 34 57 68 81 | Student did not secure a first class with 58.67% |**

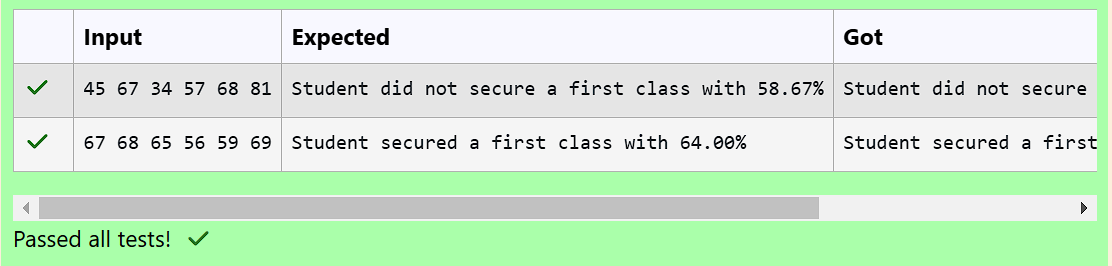
**| 67 68 65 56 59 69 | Student secured a first class with 64.00% |**

**Program:**

****

****

**Output:**

****

**Question 9:**

**Write a program which uses an if-else statement to verify and print if the given number is an odd or an even.**

**For example, if the user gives the input as 10:**

**10**

**then the program should print the result as:**

**The given number 10 is an even number**

**If the input is given as 35, then the program should print the result as :**

**The given number 35 is an odd number**

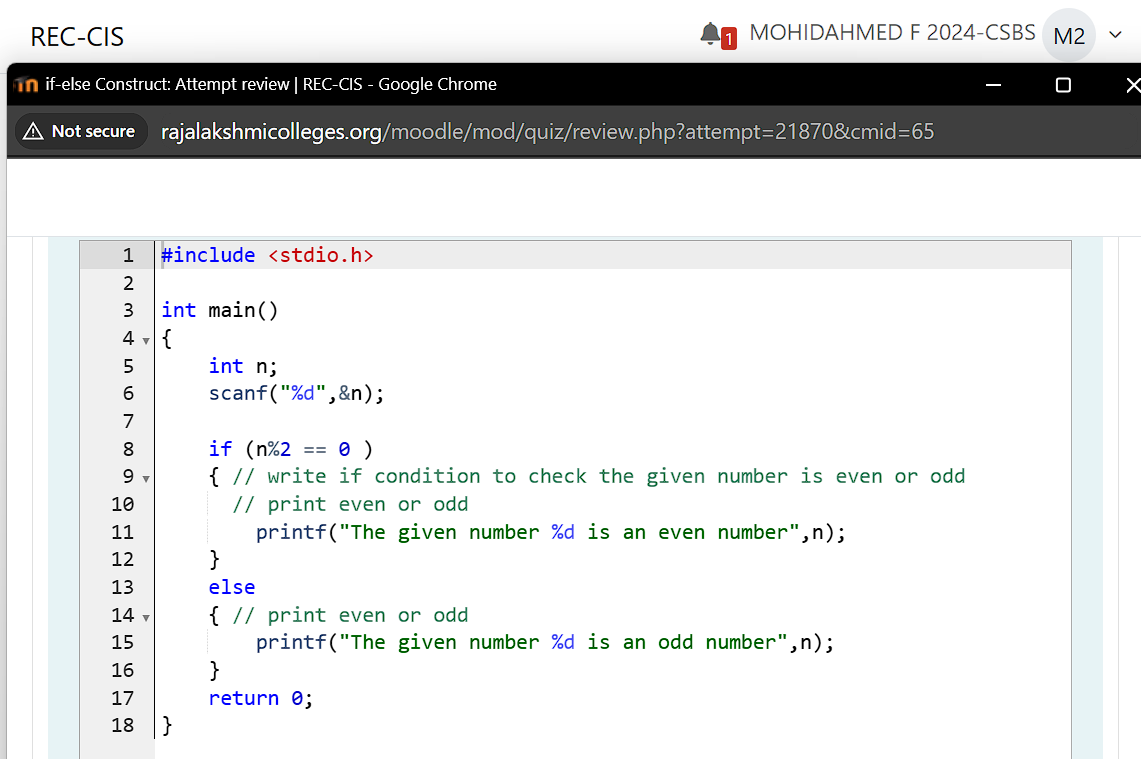
**For example:**

**| Input | Result |**

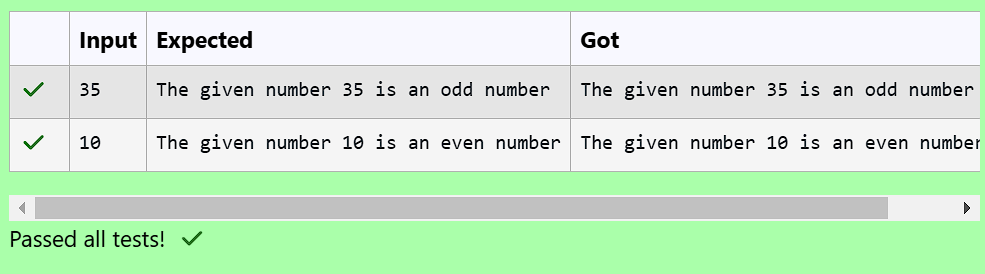
**| 35 | The given number 35 is an odd number |**

**| 10 | The given number 10 is an even number |**

**Program:**

****

**Output:**

****

**Question 10:**

**Write a program which uses an if-else statement to verify if the given character is an alphabet or not.**

**For example, if the user gives the input as W:**

**W**

**then the program should print the result as:**

**Given character W is an alphabet**

**If the input us given as 7, then print the result as:**

**Given character 7 is not an alphabet**

**[Hint: The ASCII values of alphabets 'A' to 'Z' are 65 to 90 and 'a' to 'z' are 97 to 122.]**

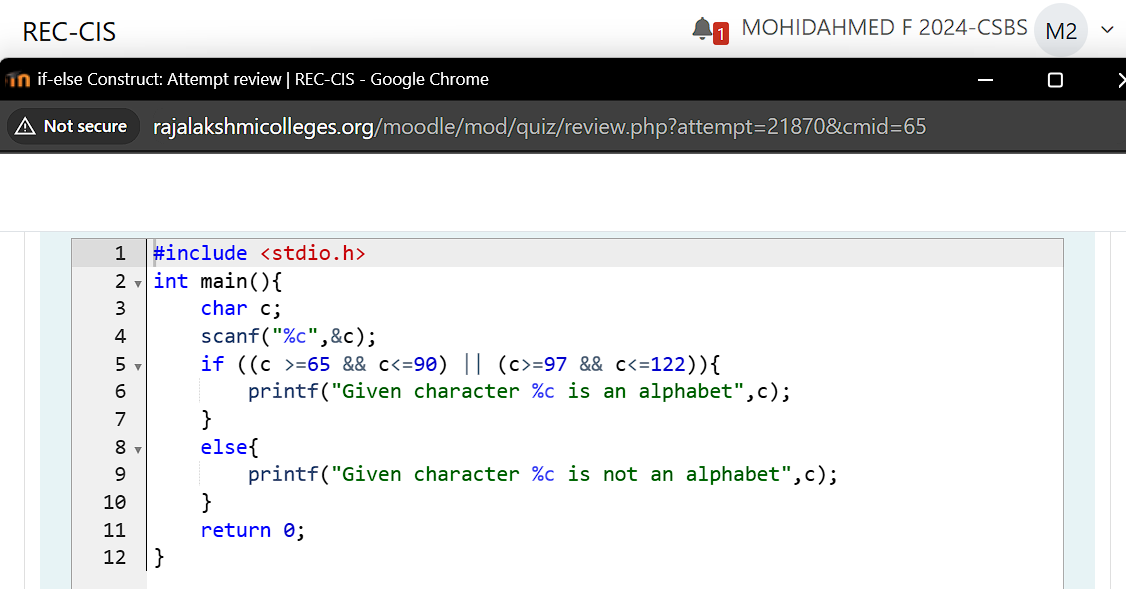
**For example:**

**| Input | Result |**

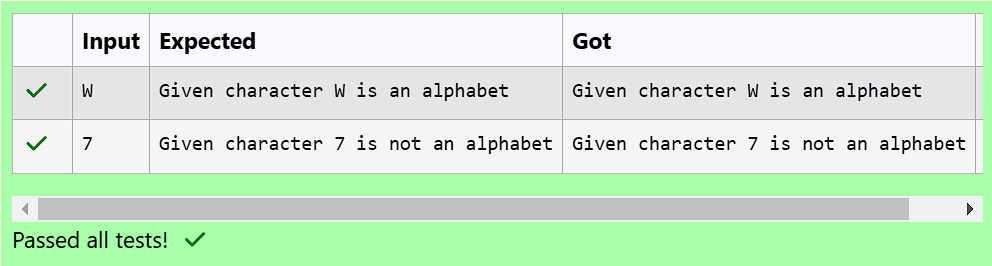
**| W | Given character W is an alphabet |**

**| 7 | Given character 7 is not an alphabet |**

**Program:**

****

**Output:**

****

**Question 11:**

**When an if-else construct appears as a statement within another if-block or a else-block, it is referred to as nesting of if-else construct.**

**Below is an example of a nested if-else construct:**

**if (expression\_1)**

**{**

**if (expression\_2)**

**{**

**if (expression\_3)**

**{**

**statement\_1;**

**}**

**else**

**{**

**statement\_2;**

**}**

**}**

**else**

**{**

**statement\_3;**

**}**

**}**

**In the above syntax, the statement\_2 will be executed only when the conditions in expression\_1, expression\_2 and expression\_3 evaluates to 1 (true).**

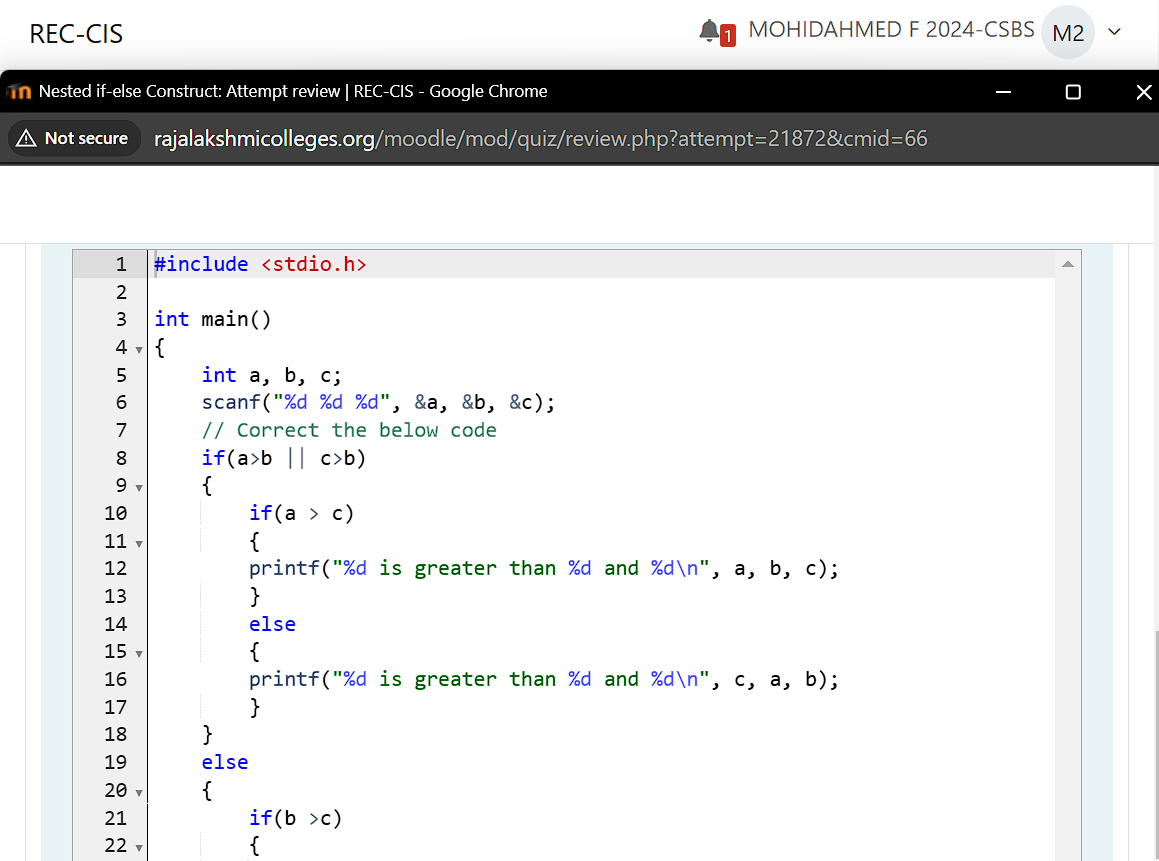
**Fill in the missing code in the below program to find the largest of three numbers using nested if-else.**

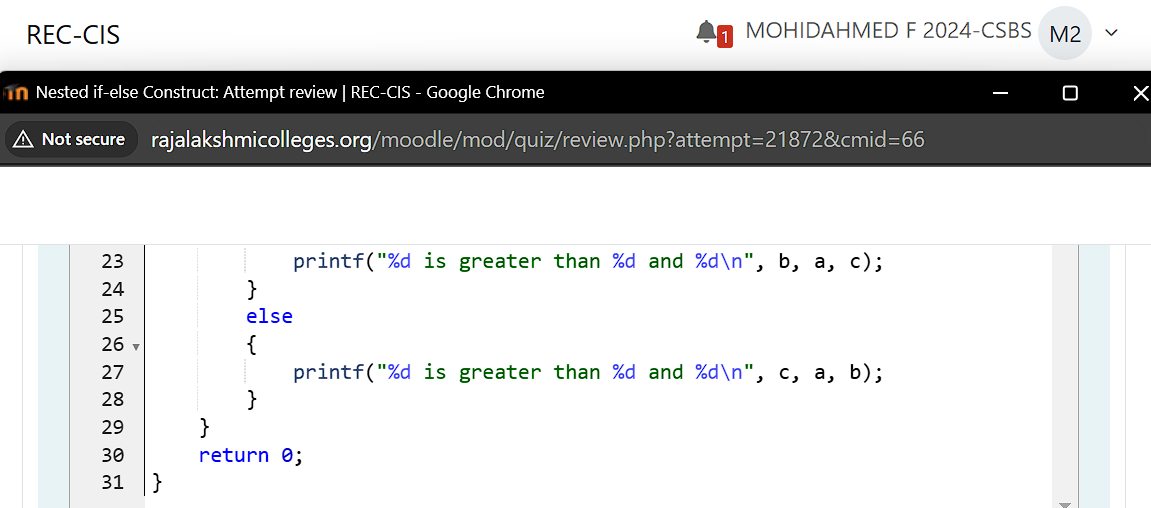
**For example:**

**| Input | Result |**

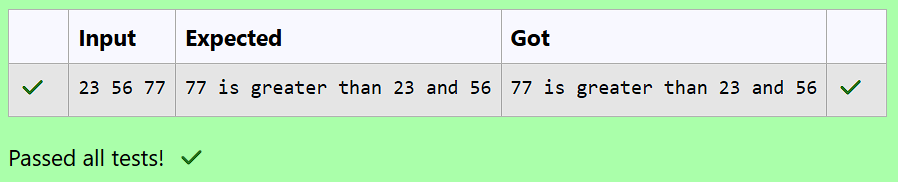
**| 23 56 77 | 77 is greater than 23 and 56 |**

**Program:**

****

****

**Output:**

****

**Question 12:**

**The if-else-if construct extends the if-else construct by allowing to chain multiple if constructs as shown below:**

**if (expression\_1)**

**{**

**statement\_1;**

**}**

**else if (expression\_2)**

**{**

**statement\_2;**

**}**

**else if (expression\_3)**

**{**

**statement\_3;**

**}**

**else if (expression\_4)**

**{**

**statement\_4;**

**}**

**else**

**{**

**statement\_5;**

**}**

**As shown in the above syntax, multiple if constructs can be chained to any length. The else construct which appears at the end is optional, and if it is to be included it has to be only at the end.**

**The if-else-if construct is used whenever we have multiple mutually exclusive if conditions which work on the same input.**

**In a if-else-if construct the conditions are evaluated from top to bottom. Whenever a condition evaluates to true (1), the control enters into that if-block and after that the control comes out of the complete if-else-if construct ignoring all the remaining if and else constructs that may exist below the currently satisfied if-block.**

**For example, if the condition in the expression\_2 is the first condition to evaluate to true after executing statement\_2, the control comes out of the complete if-else-if construct.**

**The below program reads a character from the console and should print if the given character is an alphabet or a digit. Do not remove the existing code, add the missing lines of code which employs the if-else-if statement to produce appropriate output.**

**For example:**

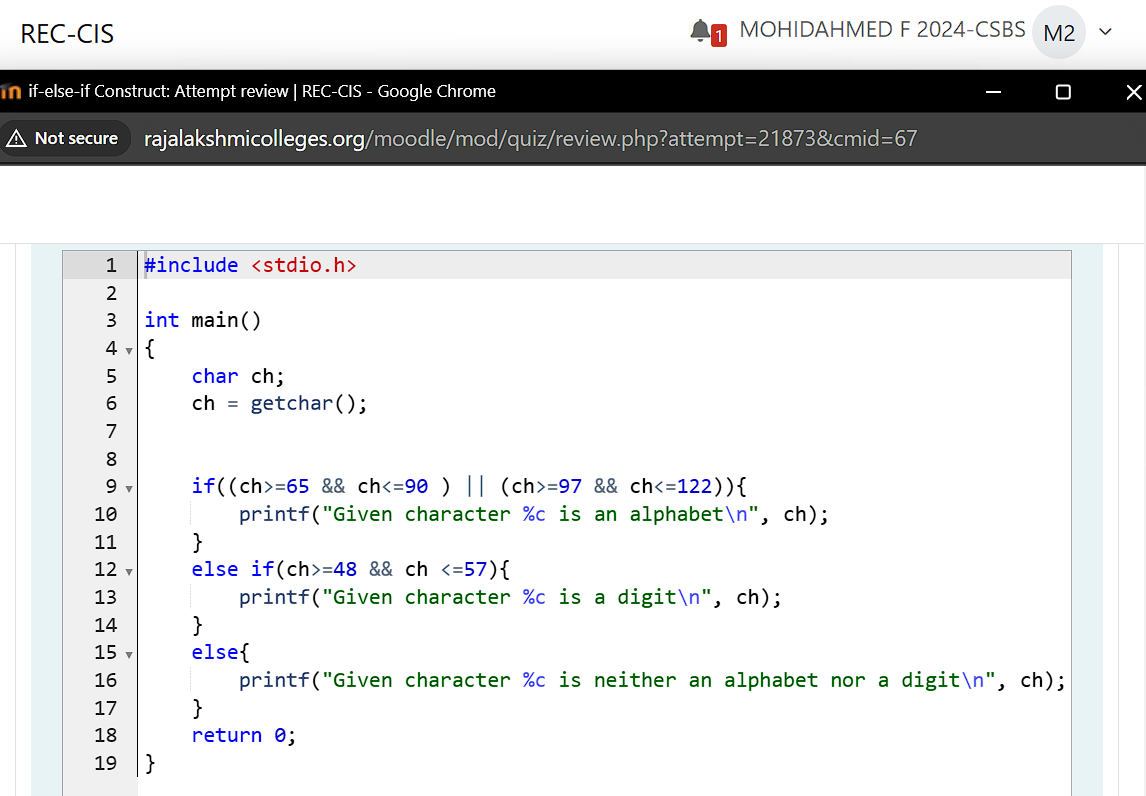
**| Input | Result |**

**| A | Given character A is an alphabet |**

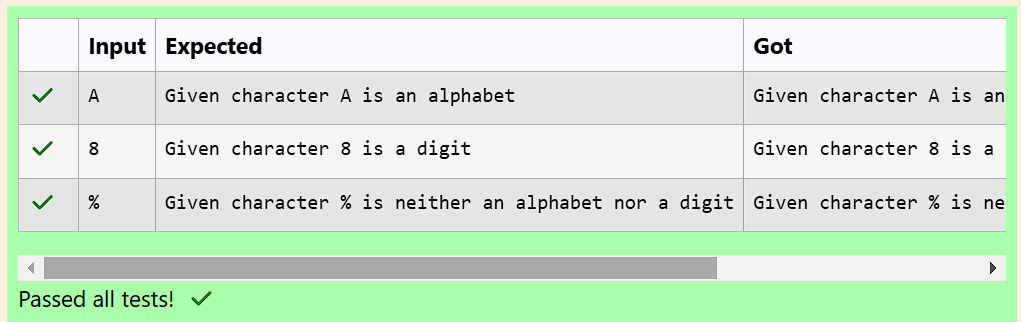
**| 8 | Given character 8 is a digit |**

**| % | Given character is neither an alphabet nor a digit |**

**Program:**

****

**Output:**

****

**Question 13:**

**The following code uses if-else statement to check whether the given integer number is a valid leap year or not.**

**Use if-else statement and print “\_ is a leap year”:**

**\* if a year is divisible by 4 and should not be divisible by 100.**

**\* If a year is divisible by 400.**

**Otherwise, print “\_ is not a leap year”.**

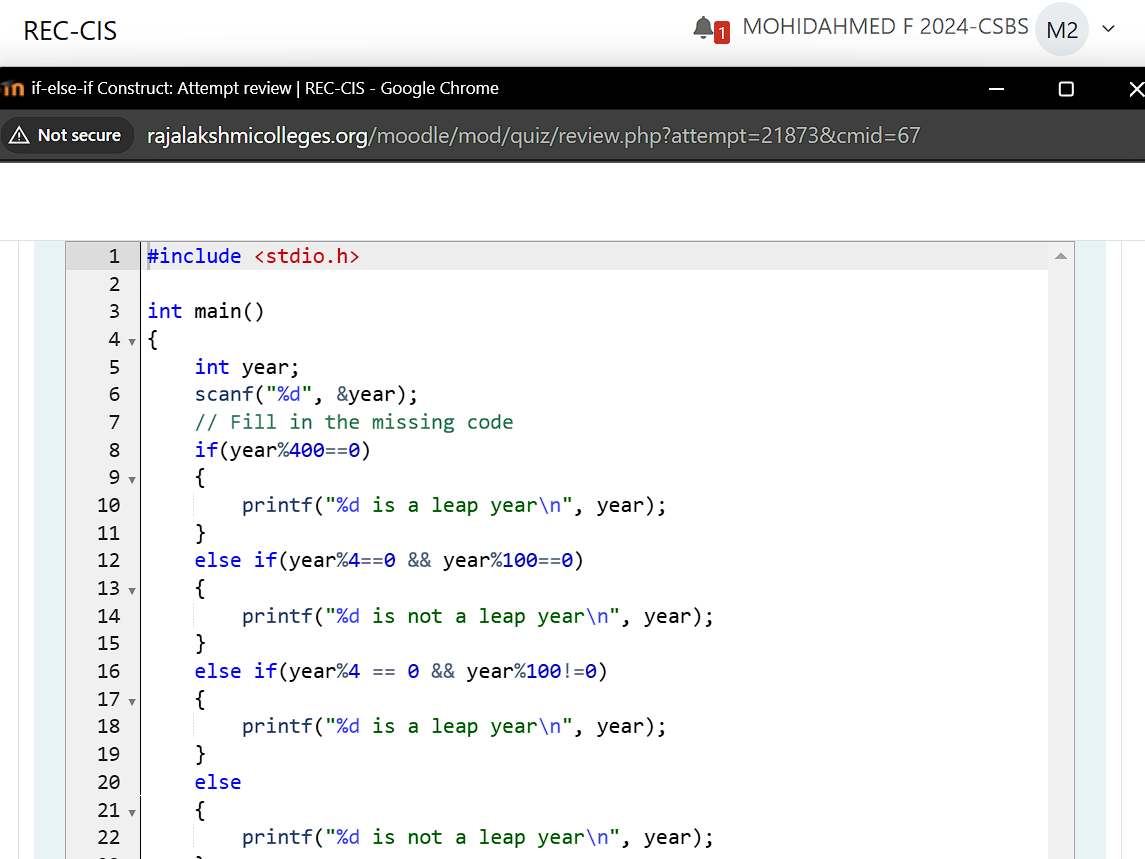
**Fill in the missing code in the below program to check whether the given year is a leap year or not..**

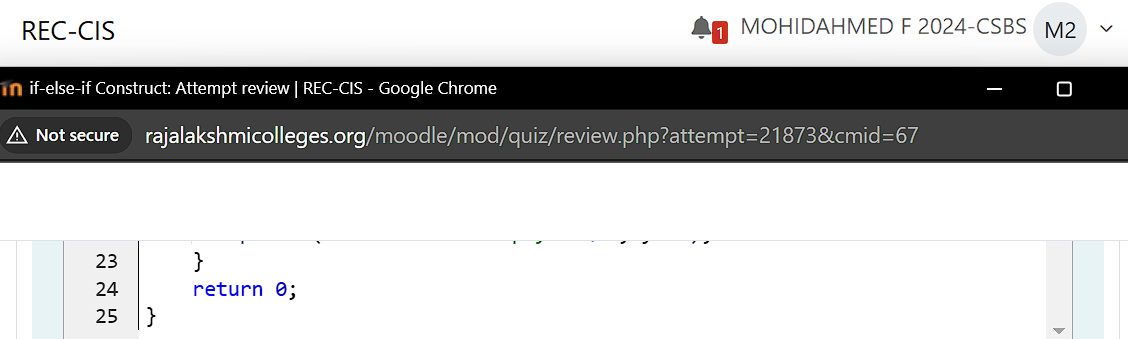
**For example:**

**| Input | Result |**

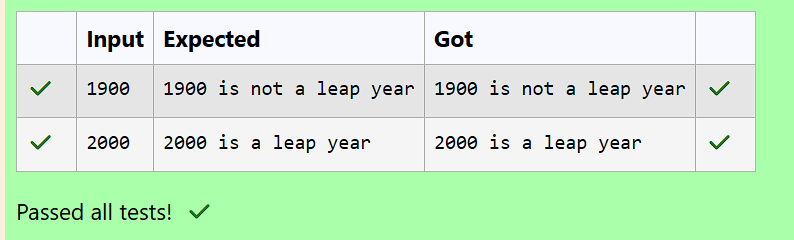
**| 1900 | 1900 is not a leap year |**

**Program:**

****

****

**Output:**

****

**Question 14:**

**Fill in the missing code in the below program to read an integer value for a variable age and use if-else statements to determine the age and print appropriate ticket price.**

**If age is less than or equal to infant\_age (3 years) or greater than or equal to centenarian\_age (100 years) then print Ticket Price: 0.**

**Otherwise, If age is less than or equal to child\_age (13 years) or greater than or equal to senior\_citizen\_age (60 years) then print Ticket Price: 5.**

**Otherwise, print Ticket Price: 10.**

**For example:**

**| Input | Result |**

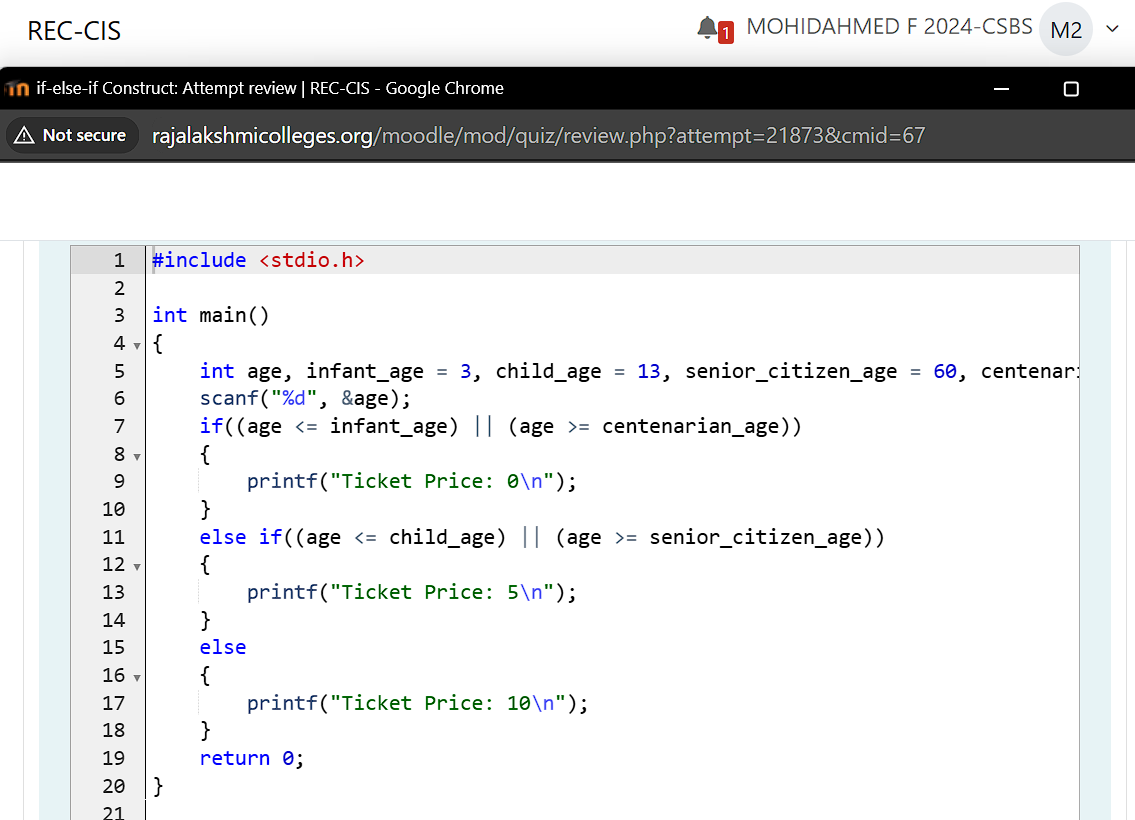
**| 34 | Ticket Price: 10 |**

**| 2 | Ticket Price: 0 |**

**| 101 | Ticket Price: 0 |**

**| 72 | Ticket Price: 5 |**

**Program:**

****

**Output:**

****

**Question 15:**

**See the below code which uses a if-else-if statement for calculating AM or PM for a given hour.**

**In the main() function read an integer value between 0 and 23 for the variable hour and use if-else-if statement to display AM or PM.**

**Fill in the if condition to check if the given hour is between 0 and 11 (both inclusive) for AM. Fill in the else if condition to check if the given hour is between 12 and 23 (both inclusive) for PM.**

**For example:**

**| Input | Result |**

**| 9 | AM |**

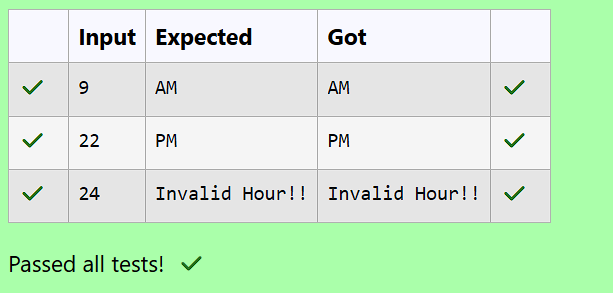
**| 22 | PM |**

**| 24 | Invalid Hour!! |**

**Program:**

****

**Output:**

****

**Question 16:**

**A switch statement is used to change the control flow of a program execution through multiple paths depending on an expression's value.**

**The below code demonstrates how to use a switch-case construct to print the corresponding English words for the digits (1 to 9) read from the standard input.**

**One way is to write a long nested if-else-if for the 10 numbers or the other way is to use a switch-case statement.**

**See and retype the below code which demonstrates the usage of switch statement to print the English word of the given number between 1 to 9.**

**#include <stdio.h>**

**int main()**

**{**

**int value;**

**scanf("%d", &value);**

**switch (value)**

**{**

**case 1:**

**printf("One");**

**break;**

**case 2:**

**printf("Two");**

**break;**

**case 3:**

**printf("Three");**

**break;**

**case 4:**

**printf("Four");**

**break;**

**case 5:**

**printf("Five");**

**break;**

**case 6:**

**printf("Six");**

**break;**

**case 7:**

**printf("Seven");**

**break;**

**case 8:**

**printf("Eight");**

**break;**

**case 9:**

**printf("Nine");**

**break;**

**case 10:**

**printf("Ten");**

**break;**

**default:**

**printf("Number %d is not in the range 1 to 10", value);**

**}**

**return 0;**

**}**

**For example:**

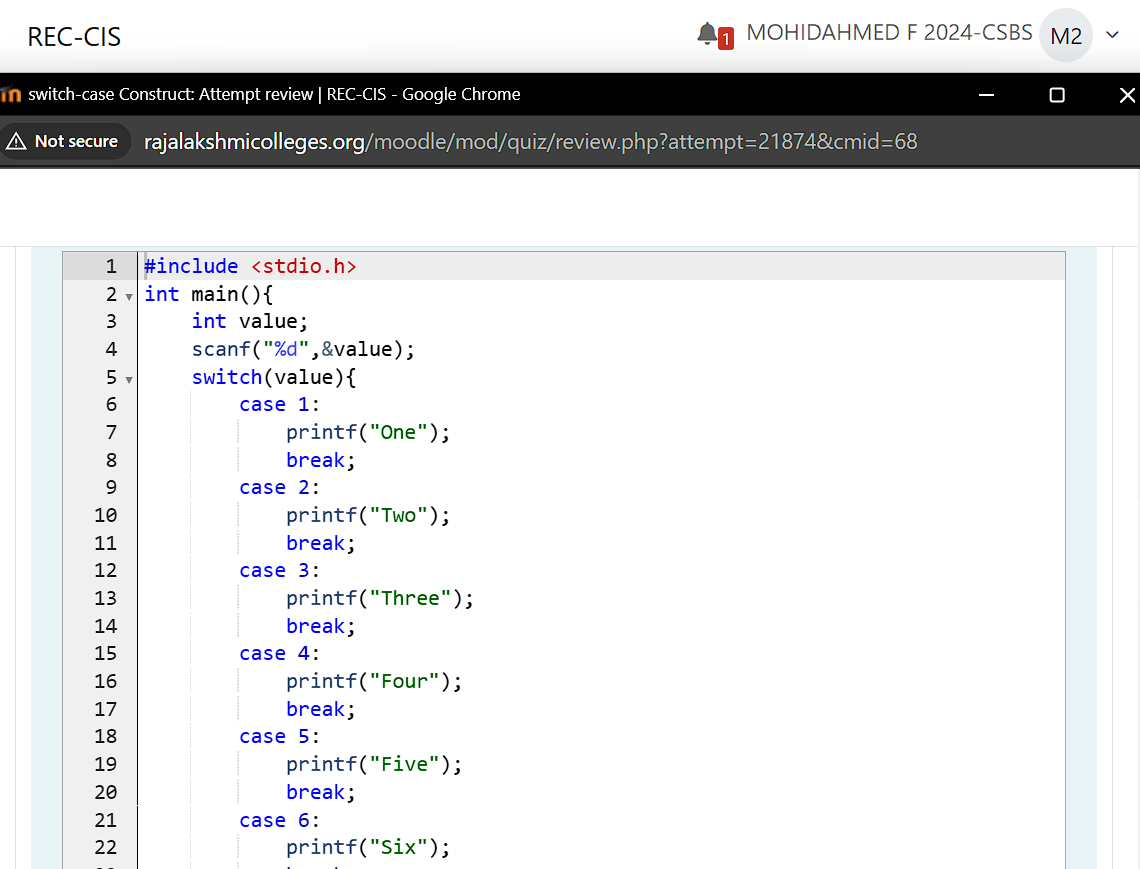
**| Input | Result |**

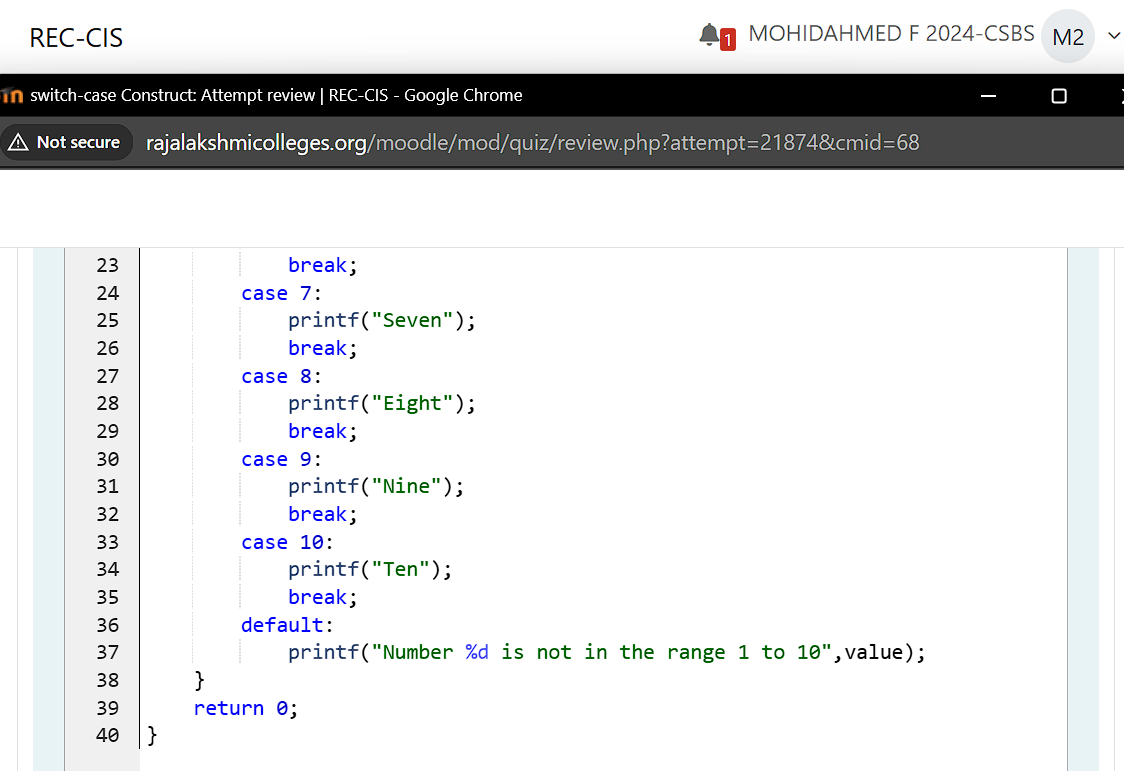
**| 2 | Two |**

**| 9 | Nine |**

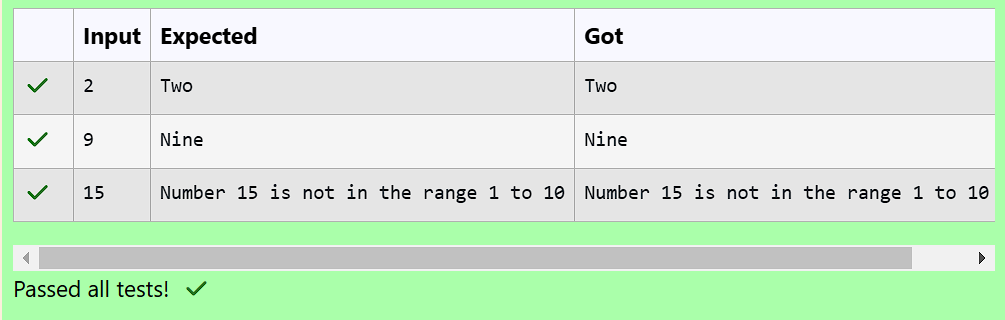
**| 15 | Number 15 is not in the range 1 to 10 |**

**Program:**

****

****

**Output:**

****

**Question 18:**

**Assume that the weekdays are provided with the below numbers:**

**\* Sunday = 0**

**\* Monday = 1**

**\* Tuesday = 2**

**\* Wednesday = 3**

**\* Thursday = 4**

**\* Friday = 5**

**\* Saturday = 6**

**Write a program to read the weekday number from the standard input and print the weekday name using switch-case.**

**For example, if the user gives the input as 1:**

**1**

**then the program should print the result as:**

**Monday**

**Note: If the given input number is not in the range i.e., other than 0 to 6, the output should be as given below:**

**Invalid weekday number**

**For example:**

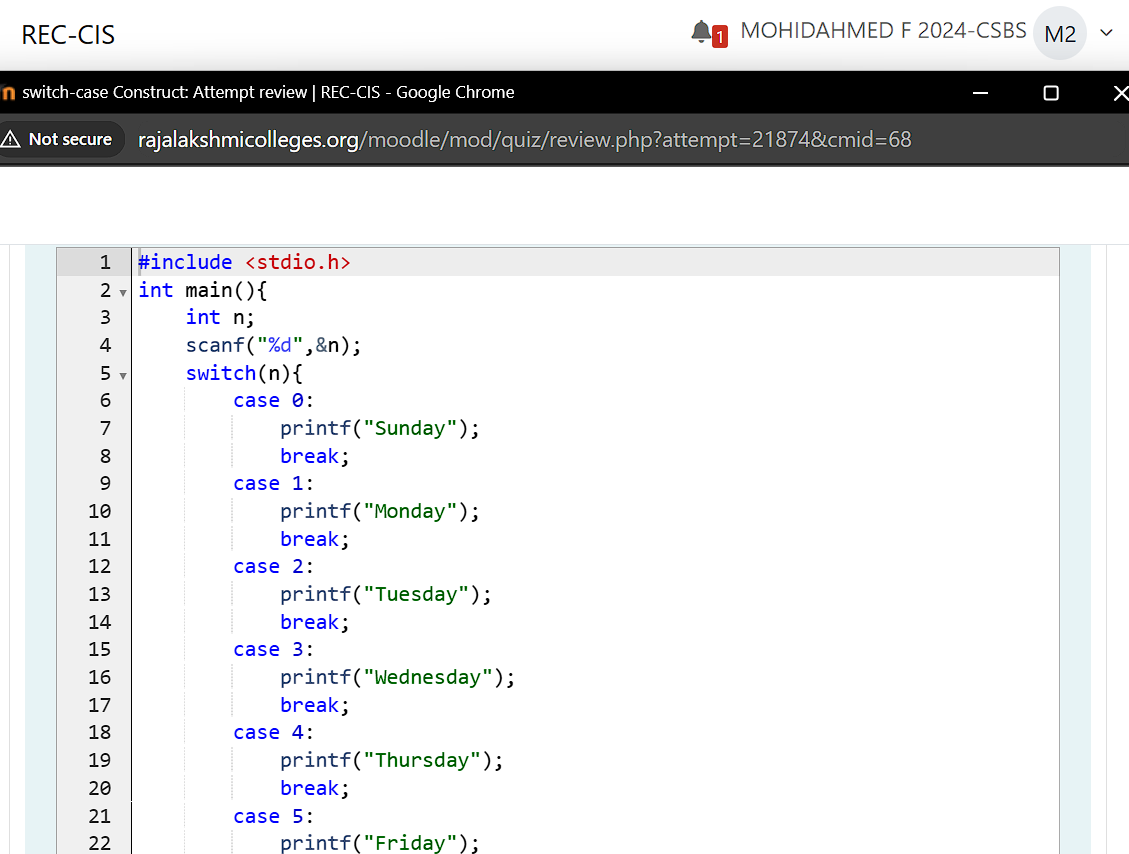
**| Input | Result |**

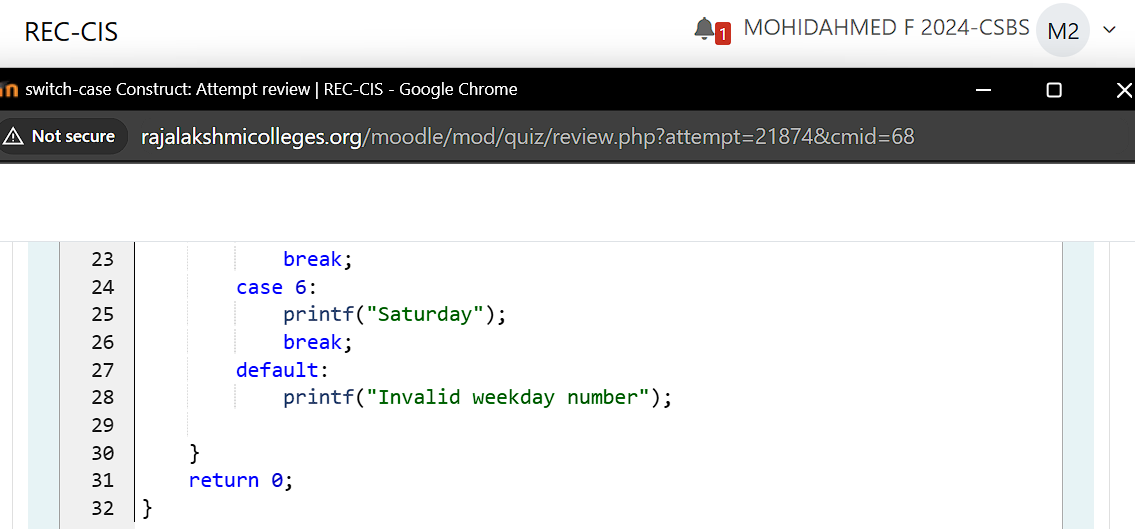
**| 6 | Saturday |**

**| 0 | Sunday |**

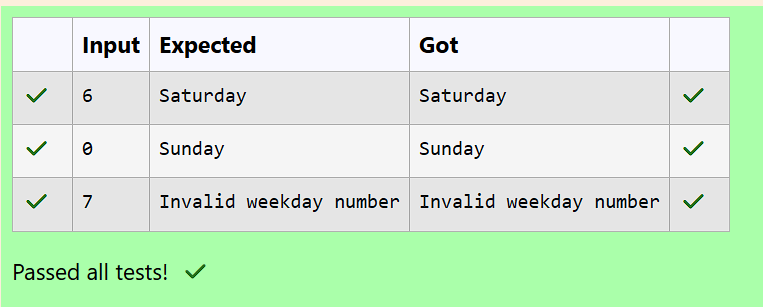
**| 7 | Invalid weekday number |**

**Program:**

****

****

**Output:**

****

**Question 19:**

**Most of the programming languages provide a special construct/statement using which we can repeatedly execute one or more statements as long as a condition is true. In C, we have while, do-while and for as the three main looping constructs or statements.**

**Below is a general syntax for using a while statement:**

**while (condition)**

**{**

**statement\_1;**

**statement\_2;**

**...**

**}**

**The block of code inside the opening and closing brace which follows the while-statement is called the while-loop body.**

**A while statement is used to execute some code repeatedly as long as a condition evaluates to true.**

**The condition is an expression which should always evaluate to either true or false.**

**\* If it evaluates to true, the body containing one or more code statements is executed.**

**\* If the expression evaluates to false, the control skips executing the while-loop body.**

**The while-loop construct is also referred to as an entry controlled loop. Meaning, first the condition evaluates to true, the body of the loop is executed. After executing the body, the control is transferred back to the condition and the process continues until the condition evaluates to false.**

**See and retype the below code which uses a while-loop to read multiple numbers from standard input and prints their sum when the sum exceeds 100.**

**#include <stdio.h>**

**int main()**

**{**

**int total = 0;**

**while (total <= 100)**

**{**

**int num;**

**scanf("%d", &num);**

**total += num;**

**}**

**printf("The total of given numbers is: %d", total);**

**return 0;**

**}**

**For example:**

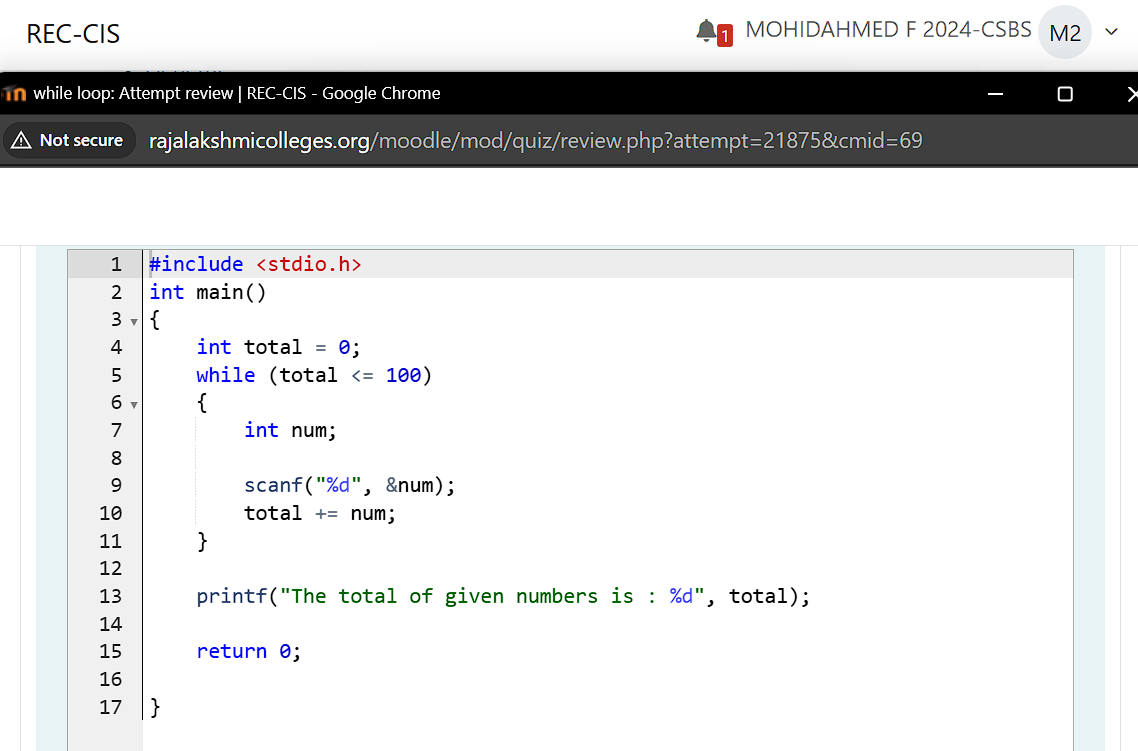
**| Input | Result |**

**| 34 | The total of given numbers is: 120 |**

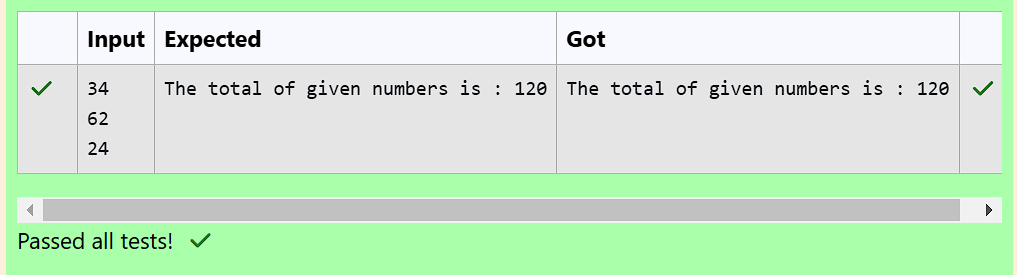
**| 62 | |**

**| 24 | |**

**Program:**

****

**Output:**

****

**Question 20:**

**The below sample code should print Ganga by number of times, where as the input is read by the programmer using scanf.**

**Fill in the missing code so that it produces the desired output.**

**For example:**

**| Input | Result |**

**| 3 | Ganga**

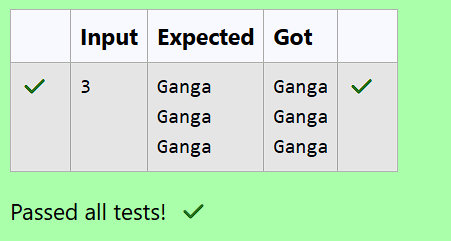
**Ganga**

**Ganga**

**Program:**

****

**Output:**

****

**Question 21:**

**Write a C program to print first n natural numbers.**

**For example, if the user gives the input as:**

**3**

**then the program should print the result as:**

**The natural numbers from 1 - 3: 1 2 3**

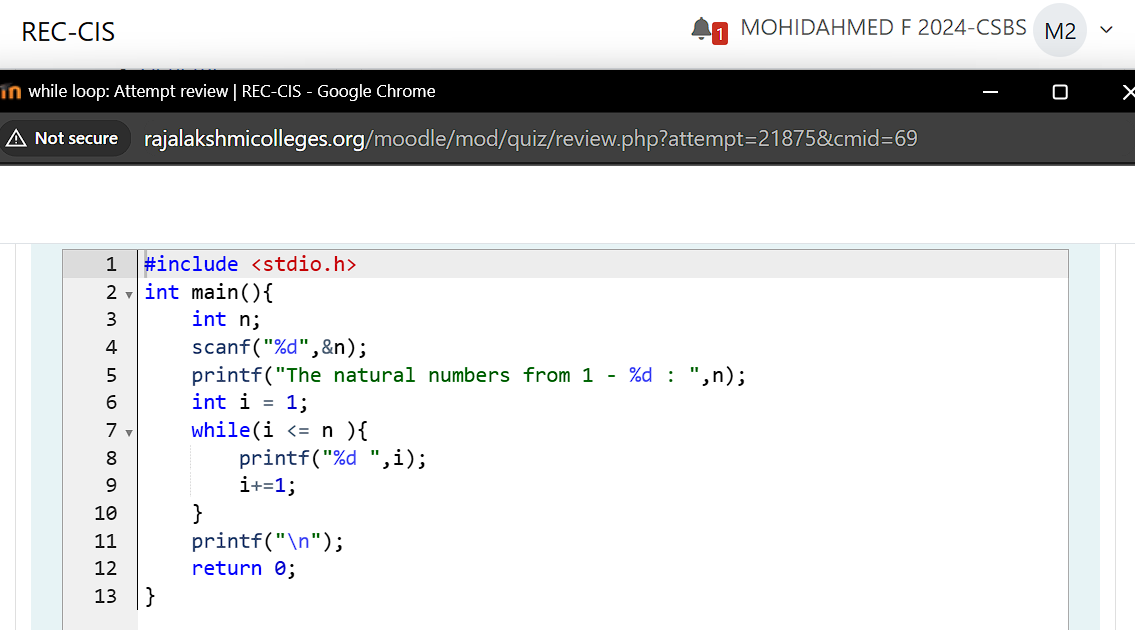
**For example:**

**| Input | Result |**

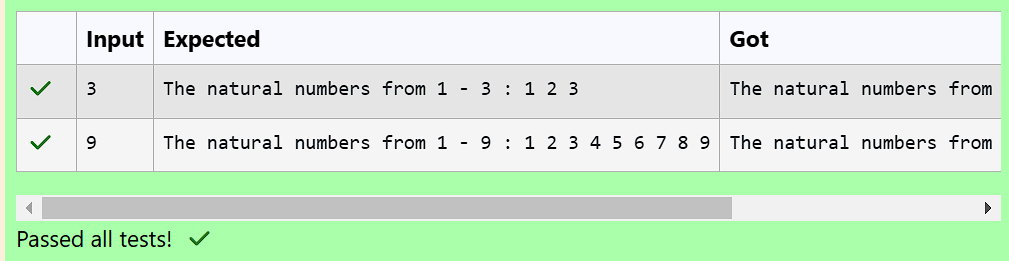
**| 3 | The natural numbers from 1 - 3: 1 2 3 |**

**| 9 | The natural numbers from 1 - 9: 1 2 3 4 5 6 7 8 9 |**

**Program:**

****

**Output:**

****

**Question 22:**

**The below sample code should find the sum of even numbers between any two numbers.**

**[Hint: The numbers should be read by using scanf()].**

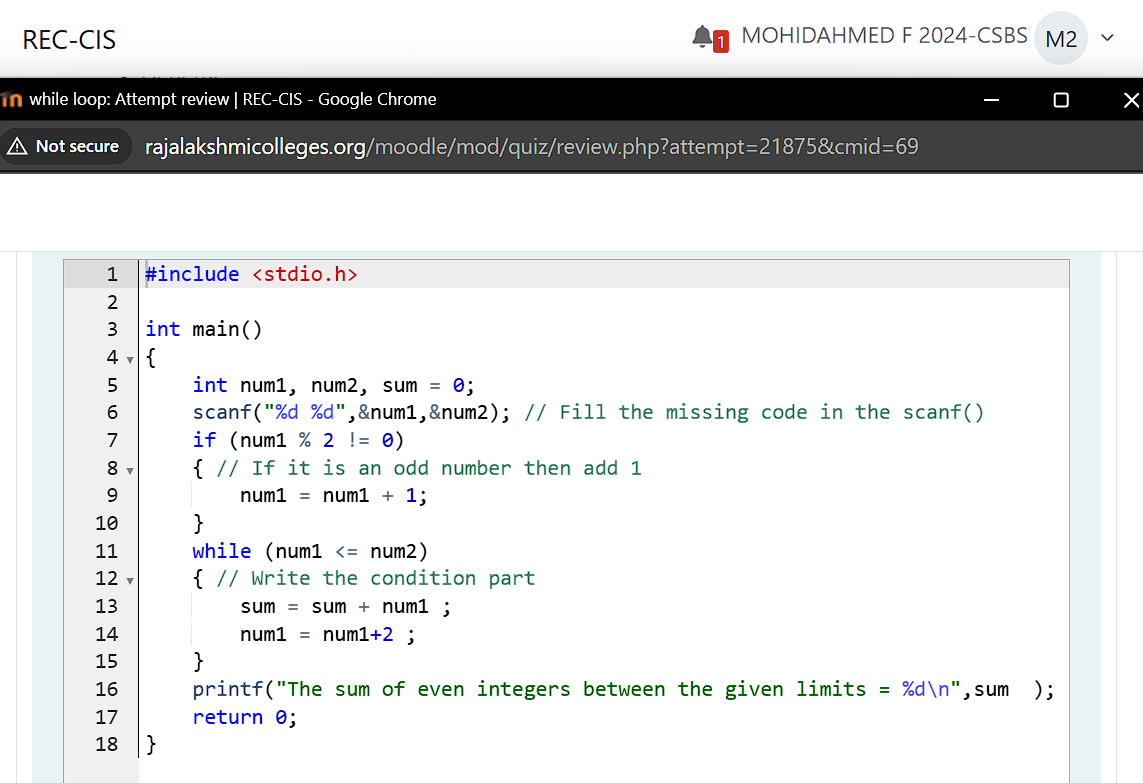
**Fill in the missing code so that it produces the desired output.**

**For example:**

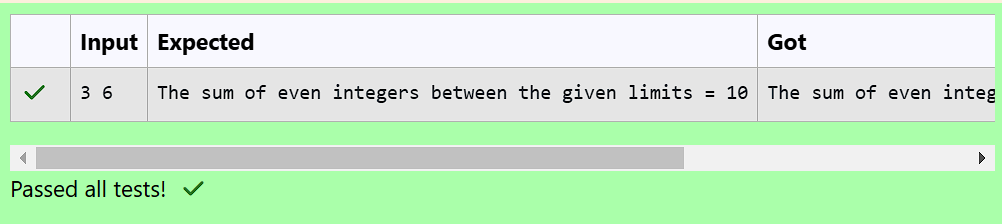
**| Input | Result |**

**| 3 6 | The sum of even integers between the given limits = 10 |**

**Program:**

****

**Output:**

****

**Question 23:**

**Fill in the missing code in the below program to read an integer number and find the reverse of the given number.**

**For example: if the input is 1234, then the output will be 4321.**

**Hints**

**The logic of reversing of any number is pretty simple if you know how to find last digit of any number. Initially the variable reverse contains zero(0), the process of reversing involves four basic steps:**

**\* Multiply the reverse variable by 10.**

**\* Find the last digit of the given number by applying % 10.**

**\* Add the last digit just found to reverse.**

**\* Divide the original number by 10 to eliminate the last digit, which is not needed anymore.**

**Repeat the above four steps till the original number becomes 0 and finally we will be left with the reversed number variable.**

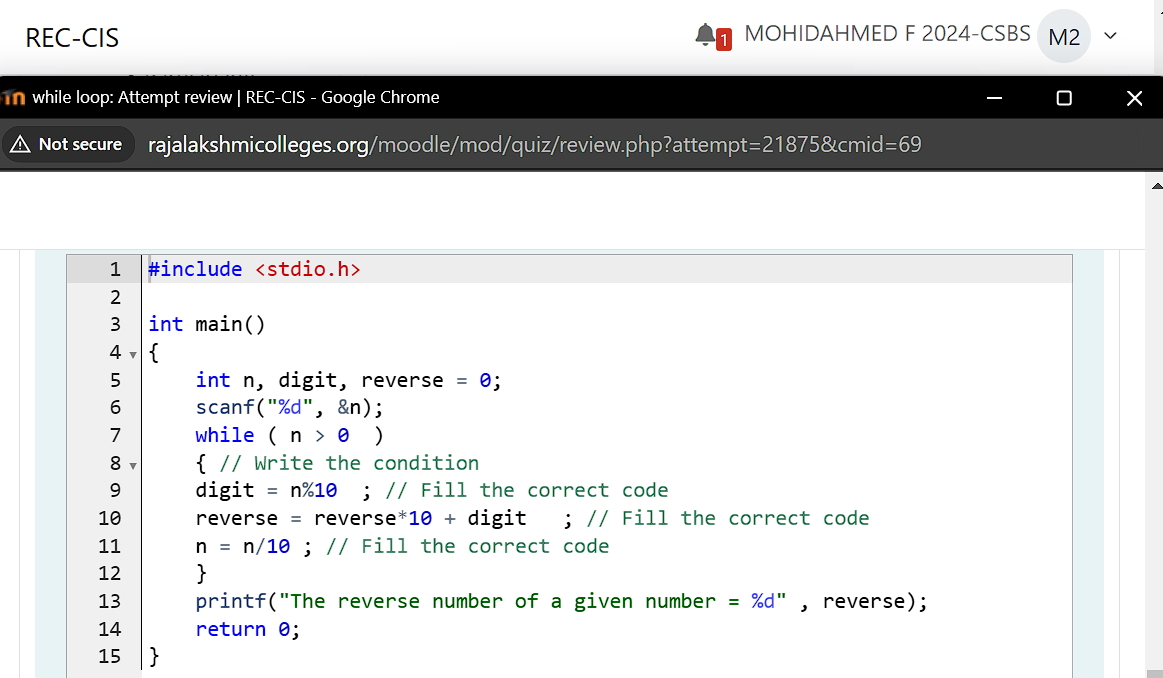
**For example:**

**| Input | Result |**

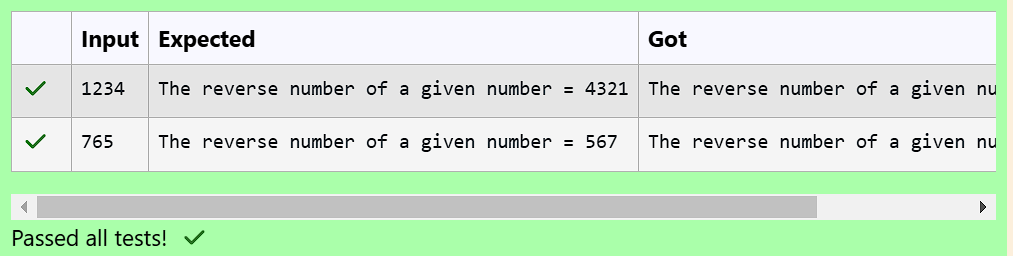
**| 1234 | The reverse number of a given number = 4321 |**

**| 765 | The reverse number of a given number = 567 |**

**Program:**

****

**Output:**

****

**Question 24:**

**Fill in the missing code in the below sample program which finds the factorial of a given number.**

**Factorial of a non-negative integer n, denoted by n!, is the product of all positive integers less than or equal to n. For example, 5! = 5 \* 4 \* 3 \* 2 \* 1 = 120.**

**The below sample code computes the factorial of a given non-zero integer.**

**The main() function declares an integer variable factorial and initializes it to 1, which it will use to store the computed value.**

**It uses a while-loop to iterate from 2 to n multiplying the loop counter in each iteration with the factorial and storing it again in factorial.**

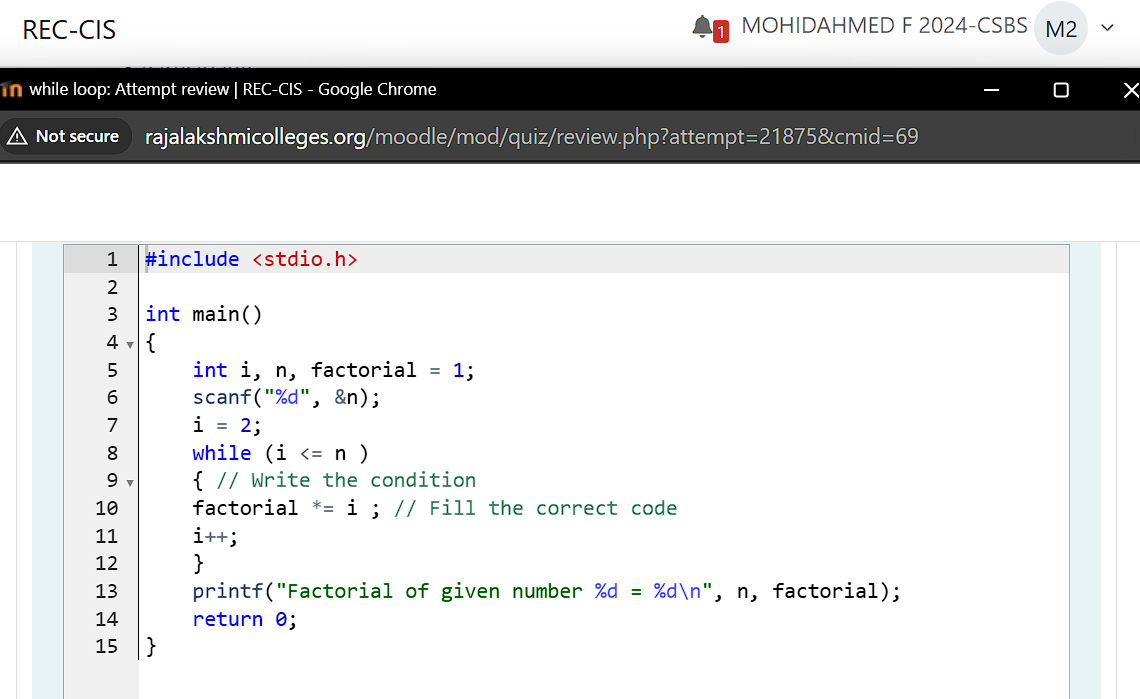
**For example:**

**| Input | Result |**

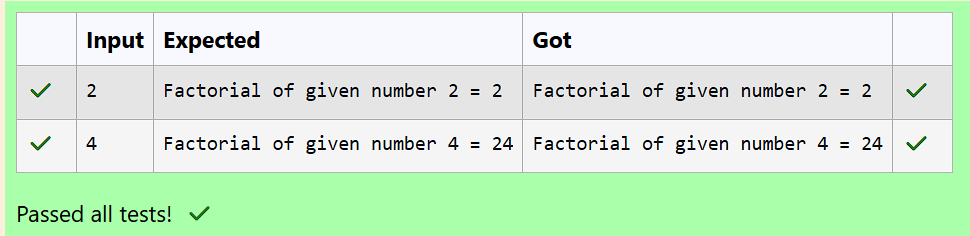
**| 2 | Factorial of given number 2 = 2 |**

**| 4 | Factorial of given number 4 = 24 |**

**Program:**

****

**Output:**

****

**Question 25:**

**Below partial code is to verify if the given number is a prime number or not.**

**A prime number is a positive integer greater than 1, which is not divisible by any other number other than 1 and itself. A few prime numbers are 2, 3, 5, 7, 11, 13, 17, 19, etc.**

**Fill in the missing code so that it produces the desired output.**

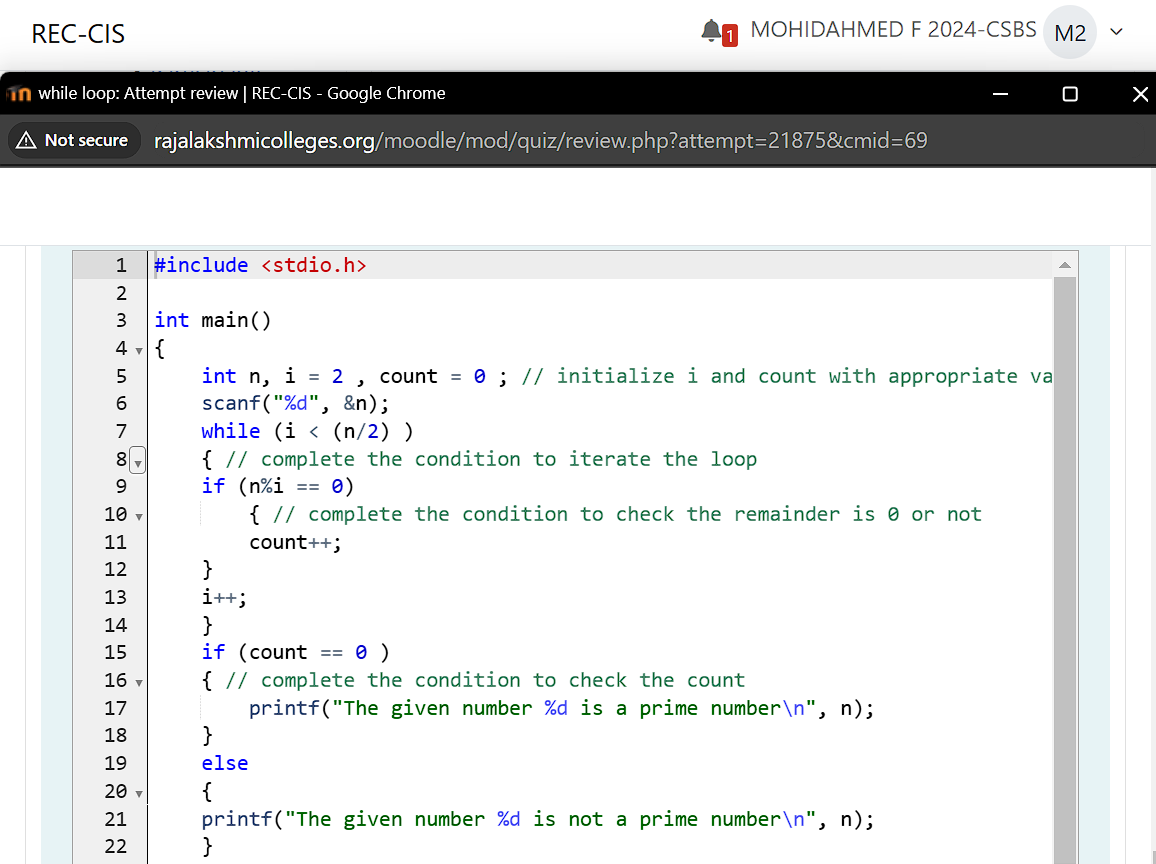
**For example:**

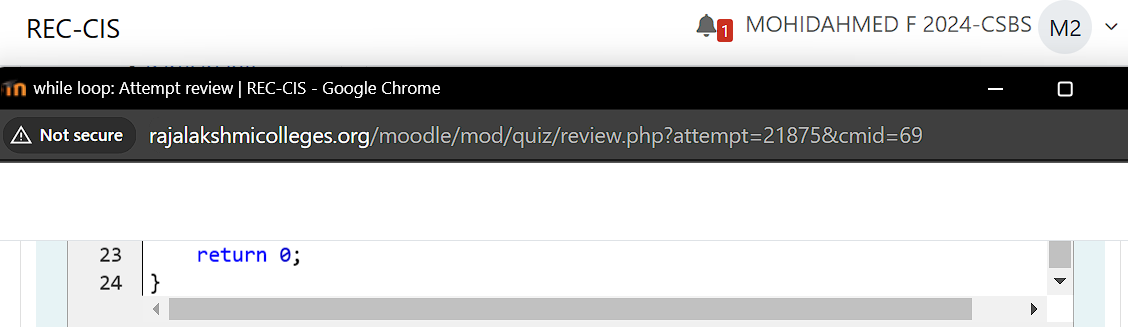
**| Input | Result |**

**| 7 | The given number 7 is a prime number |**

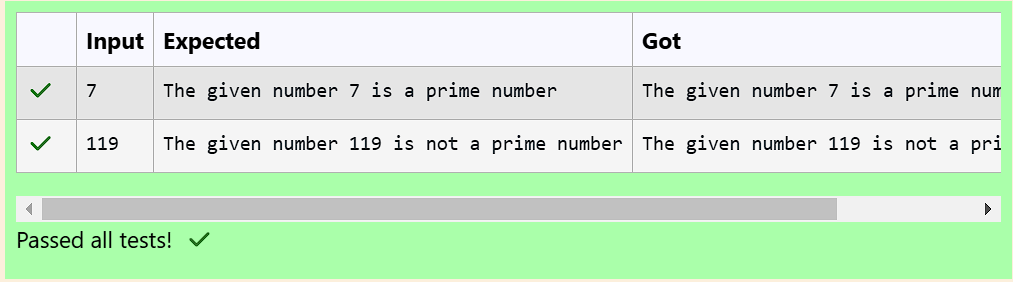
**| 119 | The given number 119 is not a prime number |**

**Program:**

****

****

**Output:**

****

**Question 26:**

**Below partial code is to verify if the given number is an armstrong number or not.**

**An armstrong number is a number that is the sum of its own digits raised to the power of number of digits in the original number.**

**For example, if the given number is 153, the total number of digits are 3, and the sum of cubes of each digit (1³ + 5³ + 3³) is equal to the same number 153. Such a number is known as an armstrong number.**

**Let us take another example, if the given number is 9474, the total number of digits are 4, and the sum of (9⁴ + 4⁴ + 7⁴ + 4⁴) is equal to the same number 9474. Such a number is known as an armstrong number.**

**Similarly,**

**9 = 9¹ = 9**

**371 = 3³ + 7³ + 1³ = 27 + 343 + 1 = 371**

**8208 = 8⁴ + 2⁴ + 0⁴ + 8⁴ = 4096 + 16 + 0 + 4096 = 8208**

**Fill in the missing code so that it produces the desired output.**

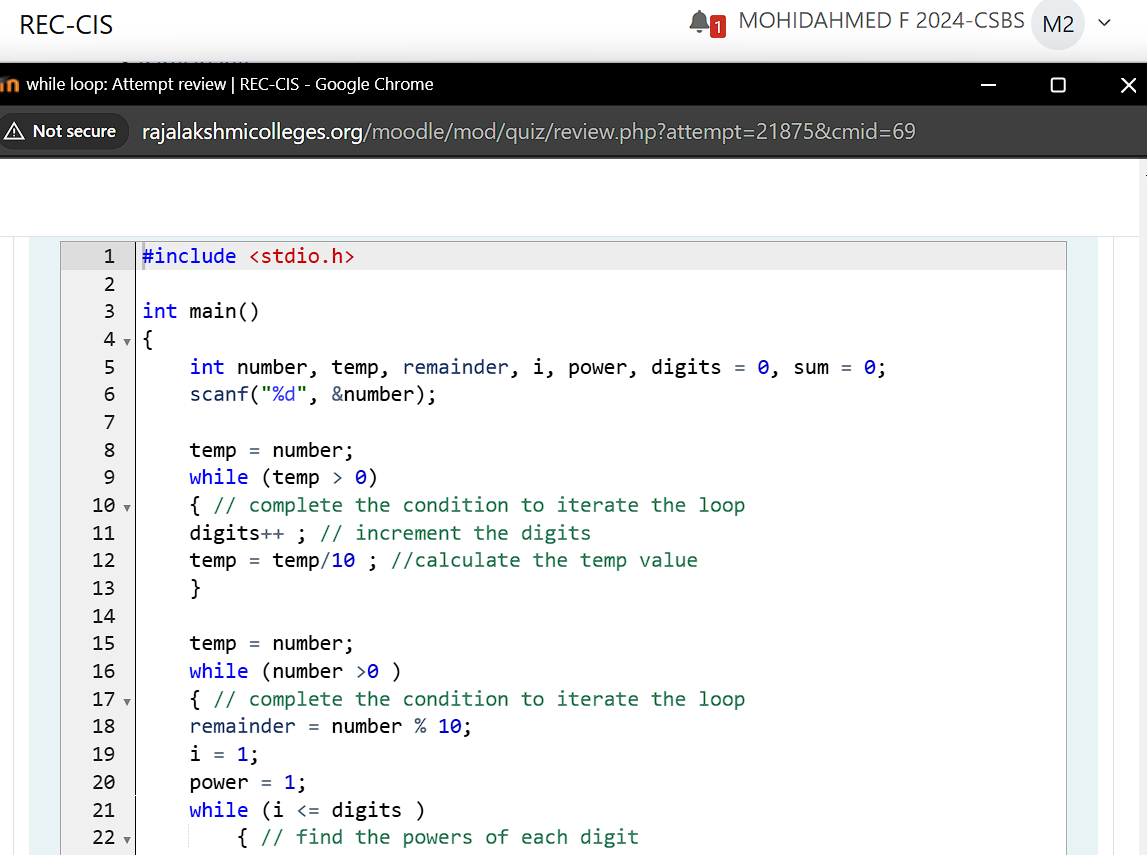
**For example:**

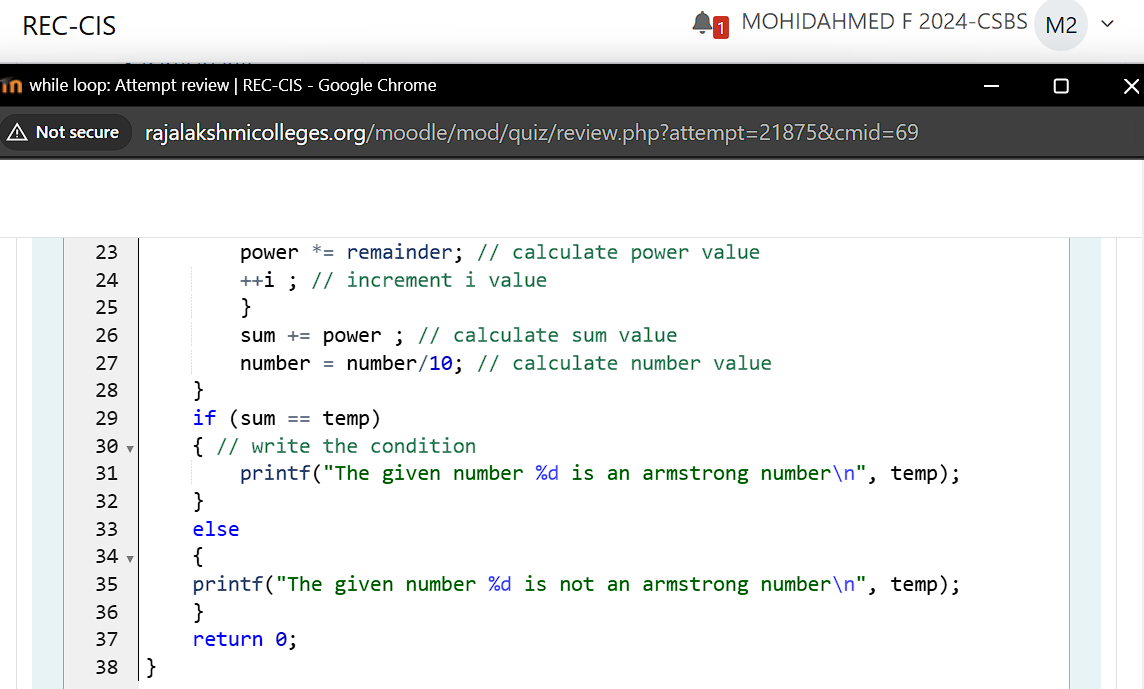
**| Input | Result |**

**| 777 | The given number 777 is not an armstrong number |**

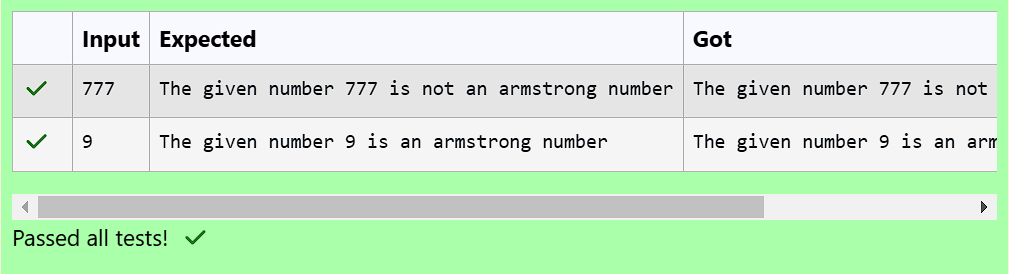
**| 9 | The given number 9 is an armstrong number |**

**Program:**

****

****

**Output:**

****

**Question 27:**

**A for-loop is used to iterate over a range of values using a loop counter, which is a variable taking a range of values in some orderly sequence (e.g., starting at 0 and ending at 10 in increments of 1).**

**The value stored in a loop counter is changed with each iteration of the loop, providing a unique value for each individual iteration. The loop counter is used to decide when to terminate the loop.**

**A for-loop construct can be termed as an entry controlled loop.**

**Below is the syntax of a for-loop:**

**for (initialization; condition; update)**

**{**

**statement(s);**

**}**

**\* The initialization expression initializes the loop counter; it is executed once at the start of the loop.**

**\* The loop continues to execute as long as the condition expression evaluates to true.**

**\* The update expression is executed after each iteration through the loop, to increment, decrement or change the loop counter.**

**Example with code:**

**int i;**

**for (i = 0; i < 10; i++)**

**{**

**printf("%d\n", i);**

**}**

**\* Above for-loop statement initializes an integer variable i (which is the loop counter) as part of the initialization expression.**

**\* In the update section, it increments the variable i by 1 using the post-increment expression i++.**

**\* The expression in condition is i < 10. The for-loop keeps on executing the code inside the loop body as long as the condition evaluates to true. And the loop terminates when the condition evaluates to false.**

**\* It is a good practice to always keep the loop body (which contains the code to be executed) within an opening-brace { and a closing-brace }.**

**Note: No ; at the end of the for statement.**

**Complete the below code to check your understanding of the for-loop syntax. The completed code should print numbers from 10 to 20, one per line.**

**For example:**

**Result**

**10**

**11**

**12**

**13**

**14**

**15**

**16**

**17**

**18**

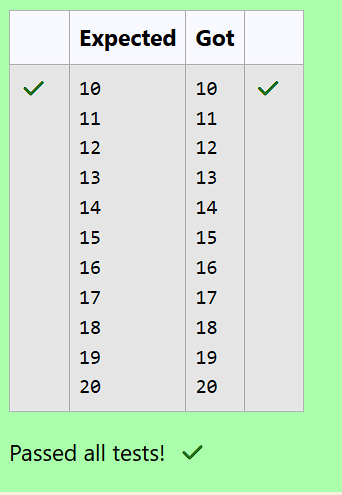
**19**

**20**

**Program:**

****

**Output:**

****

**Question 28:**

**Fill in the missing code in the below program to calculate the value of a^n, given two positive non-zero integers a and n.**

**The code in the main() function reads two integers from standard input and stores them in the variables a and n.**

**It uses a for-loop to multiply a with itself n number of times.**

**Variable a\_power\_n is used to store the computed value of a^n.**

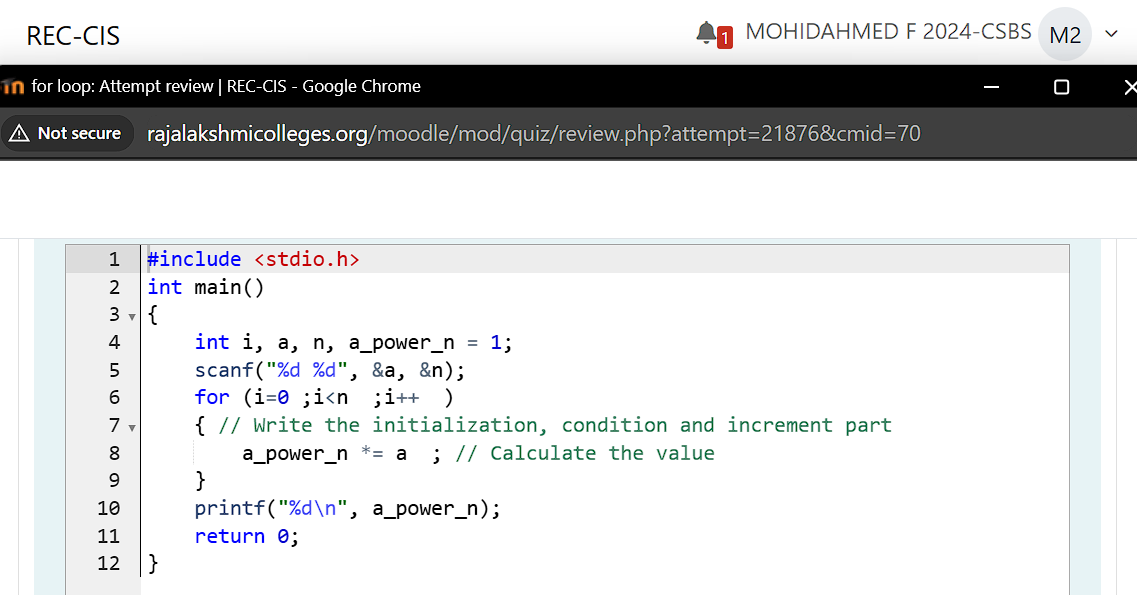
**After the execution of for-loop is completed, the final value of a\_power\_n is printed to the standard output.**

**For example:**

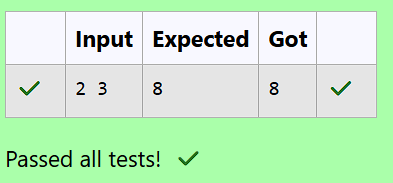
**| Input | Result |**

**| 2 3 | 8 |**

**Program:**

****

**Output:**

****

**Question 29:**

**Write a program to find sum and mean of n numbers.**

**Constraints:**

**\* 1 <= n <= 10^6**

**\* 10^-3 <= elements <= 10^3**

**\* Result of mean should print upto 2 decimal places.**

**Sample test case:**

**4 ----------------------------------------------------------------> First line of input is the value of n.**

**3 5 7 8 -------------------------------------------------------> Second line of input is n space separated integer values.**

**Sum: 23 ----------------------------------------------------------> Third line prints the Sum as required.**

**Mean: 5.75 ------------------------------------------------------> Fourth line prints the Mean as required.**

**Instruction: To run your custom test cases strictly map your input and output layout with the visible test case format.**

**For example:**

**| Input | Result |**

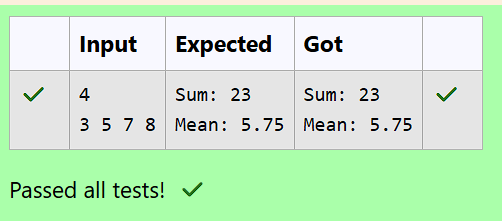
**| 4 | Sum: 23 |**

**| 3 5 7 8 | Mean: 5.75 |**

**Program:**

****

**Output:**

****

**Question 30:**

**Fill in the missing code in the below program to print the Fibonacci series i.e., 0 1 1 2 3 5 8 13 21....., up to the nth term.**

**The code in the main() function reads one integer variable n. It uses a for loop to iterate from 0 to n and print the Fibonacci series.**

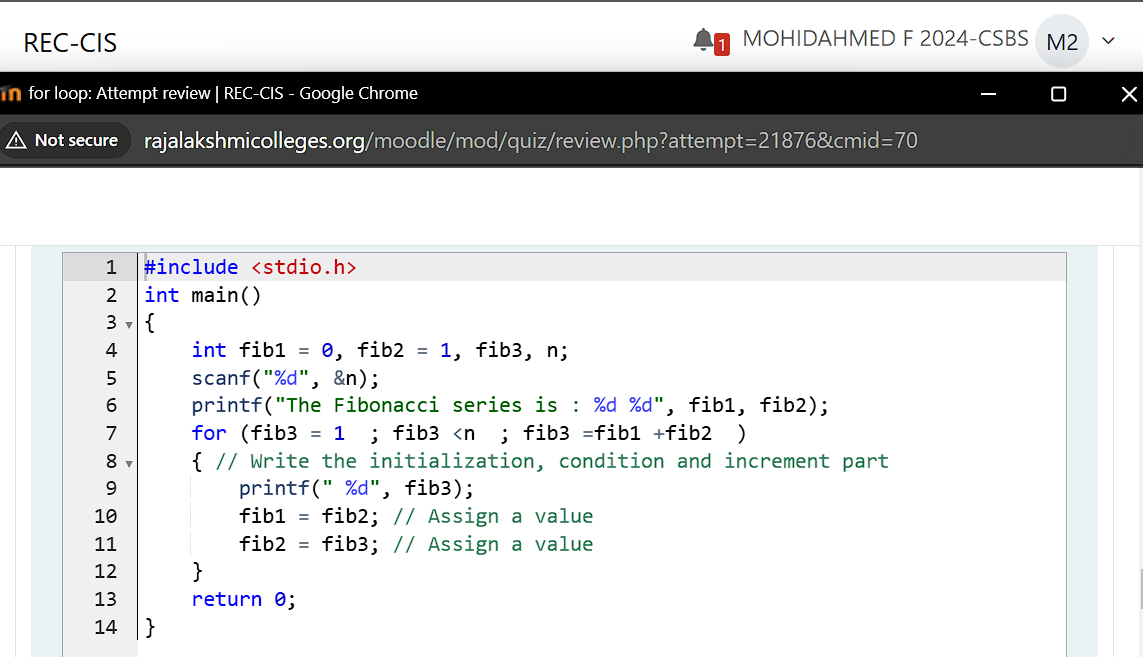
**By definition, the first two numbers in the Fibonacci sequence are 0 and 1, and each subsequent number is the sum of the previous two.**

**For example:**

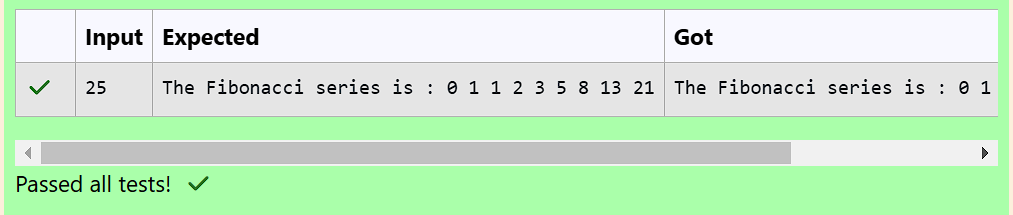
**| Input | Result |**

**| 25 | The Fibonacci series is : 0 1 1 2 3 5 8 13 21 |**

**Program:**

****

**Output:**

****

**Question 31:**

**Write a program that will print all the English alphabets from A to Z, each in a new line.**

**Hints**

**\* The code in the main() function can use a for loop to iterate over the characters 'A' to 'Z'.**

**\* Note that char data type is a numeric type and can be used in a for loop as a loop counter.**

**\* You can declare and initialize a loop counter char i and initialize it to 'A' (eg: char i = 'A';). The condition can similarly be i <= 'Z', and the update statement can be i++.**

**\* You can then print i directly which is of type char, using the printf() function with a newline character (\n).**

**For example:**

**Result**

**A**

**B**

**C**

**D**

**E**

**F**

**G**

**H**

**I**

**J**

**K**

**L**

**M**

**N**

**O**

**P**

**Q**

**R**

**S**

**T**

**U**

**V**

**W**

**X**

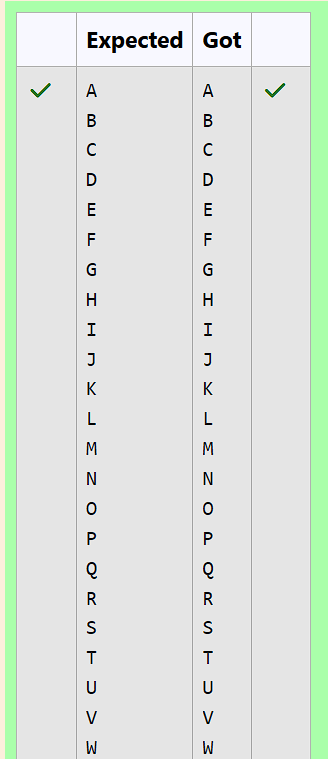
**Y**

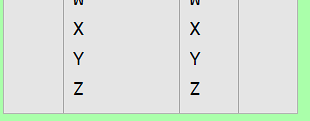
**Z**

**Program:**

****

**Output:**

****

****

**Question 32:**

**Write a program to read n numbers from the user and then count number of "Odd" and "Even" numbers.**

**Constraints:**

**\* 1 <= n <= 10^6**

**\* 10^-3 <= elements <= 10^3**

**Sample test case:**

**3----------> First line of input is n i.e. 3.**

**5 6 7-------> Second line of input is n space separated integer values/elements.**

**Even: 1-----> Third line prints the output (the count of even elements).**

**Odd: 2------->Fourth line prints the output (the count of odd elements).**

**Note: Do use the printf() function with a newline character (\n) to print your results on newline.**

**Instruction: To run your custom test cases strictly map your input and output layout with the visible test case format.**

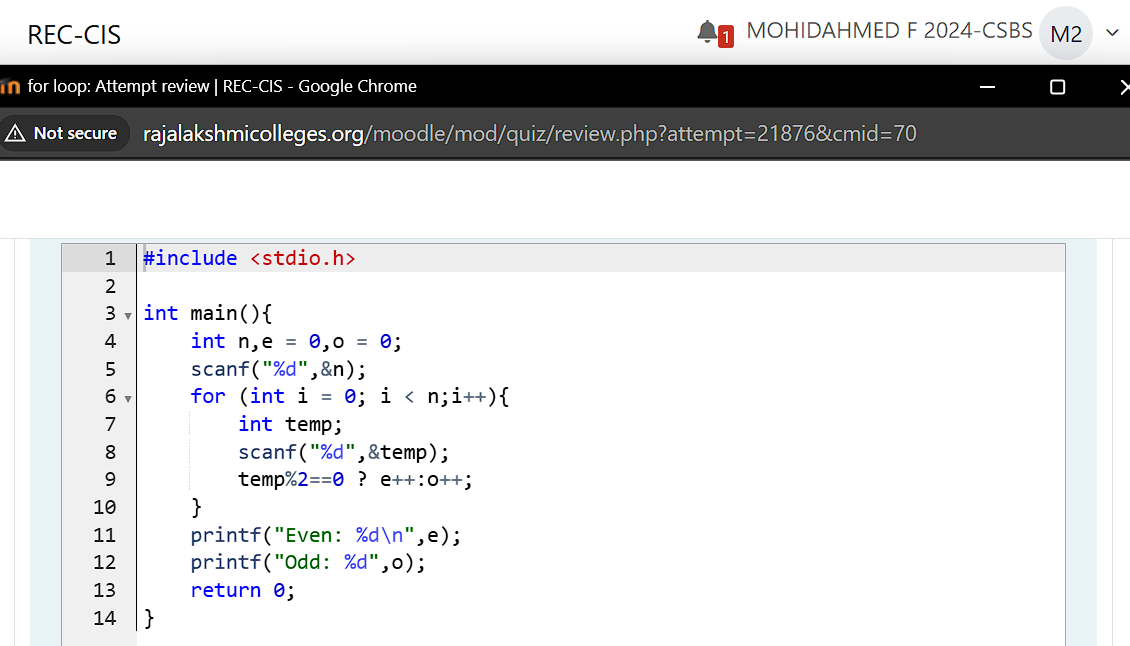
**For example:**

**| Input | Result |**

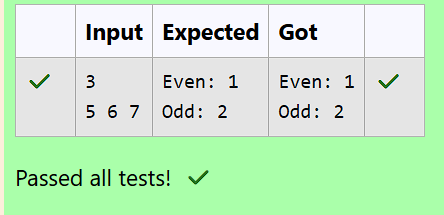
**| 3 | Even: 1 |**

**| 5 6 7 | Odd: 2 |**

**Program:**

****

**Output:**

****

**Question 33:**

**Fill in the missing code in the below program to verify whether the given number is perfect, abundant or deficient.**

**A number is said to be perfect if it equals the sum of its proper divisors. For example, 6 and 28 can be called as: 6 = 1 + 2 + 3 and 28 = 1 + 2 + 4 + 7 + 14.**

**Alternatively, if the sum of a number's proper divisors exceeds the number itself, it is said to be abundant, whereas if the sum of the number's proper divisors is less-than the number itself, it is said to be deficient.**

**For example:**

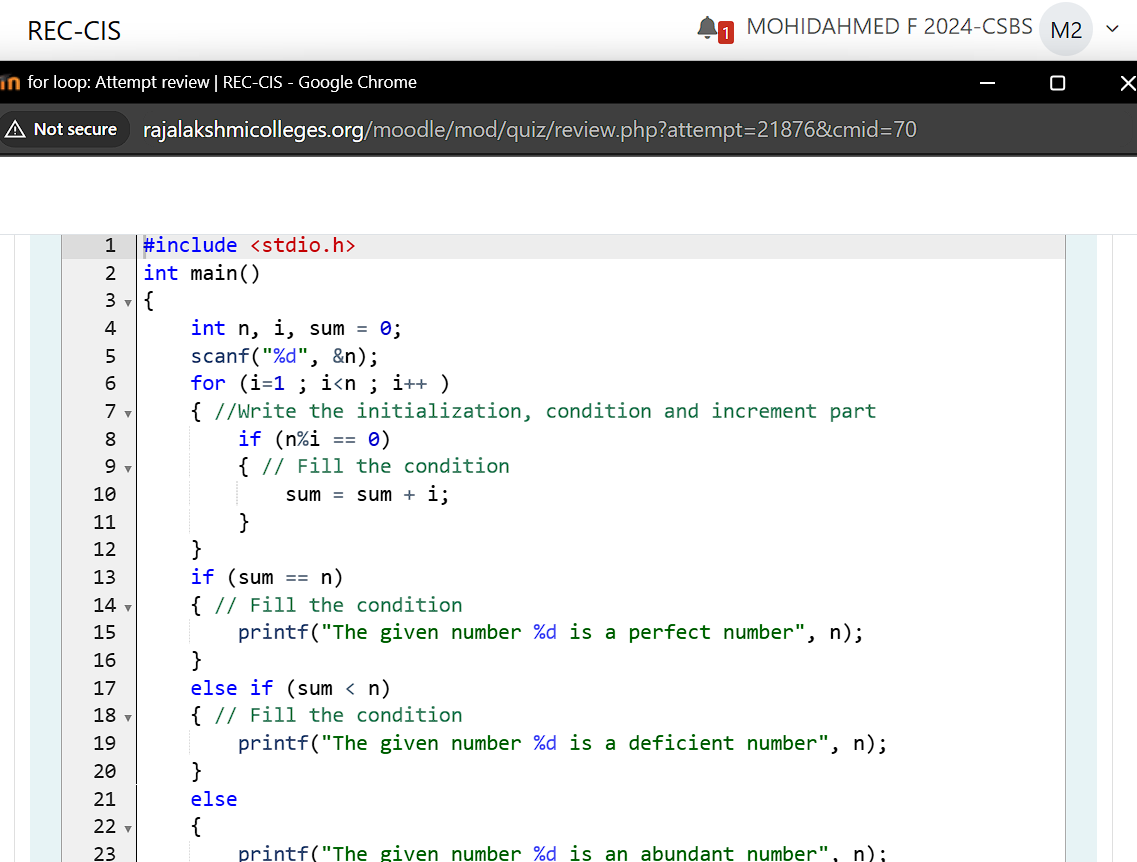
**| Input | Result |**

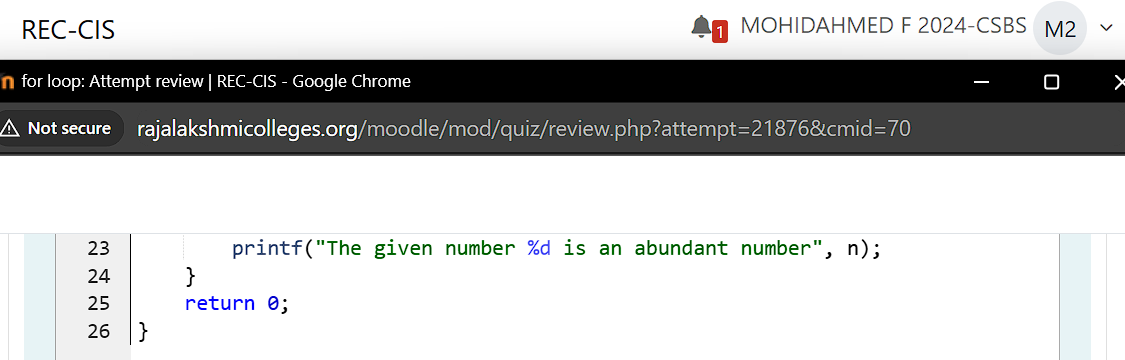
**| 6 | The given number 6 is a perfect number |**

**| 10 | The given number 10 is a deficient number |**

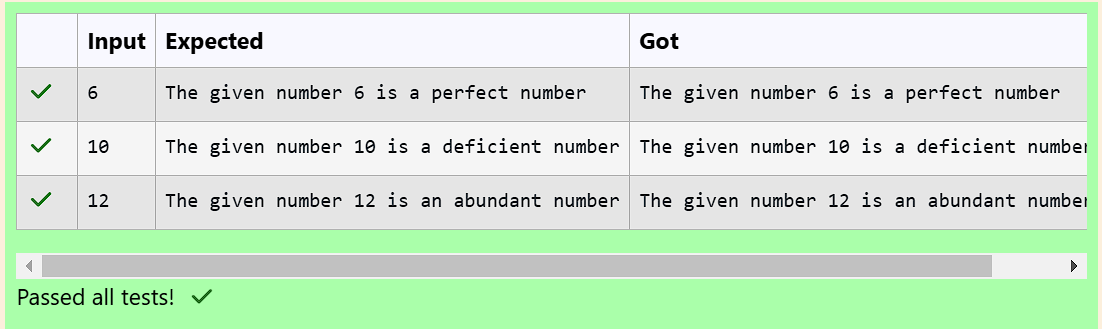
**| 12 | The given number 12 is an abundant number |**

**Program:**

****

****

**Output:**

****

**Question 34:**

**Fill in the missing code in the below program to check whether the given number is a strong number or not.**

**A number is called strong number if sum of the factorials of its digits is equal to number itself. For example: 145 is a strong number since 1! + 4! + 5! = 1 + 24 + 120 = 145.**

**The code in the below main() function reads a number from standard input and performs the verification for a strong number by extracting the individual digits and calculating their factorials.**

**For example:**

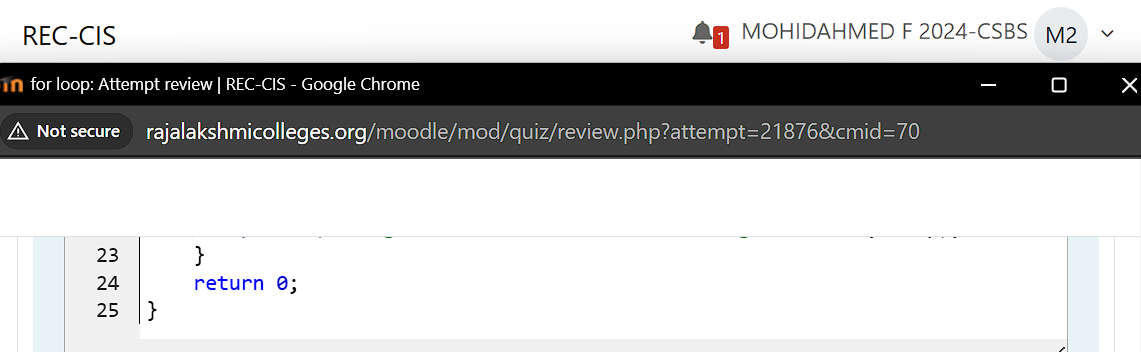
**| Input | Result |**

**| 145 | The given number 145 is a strong number |**

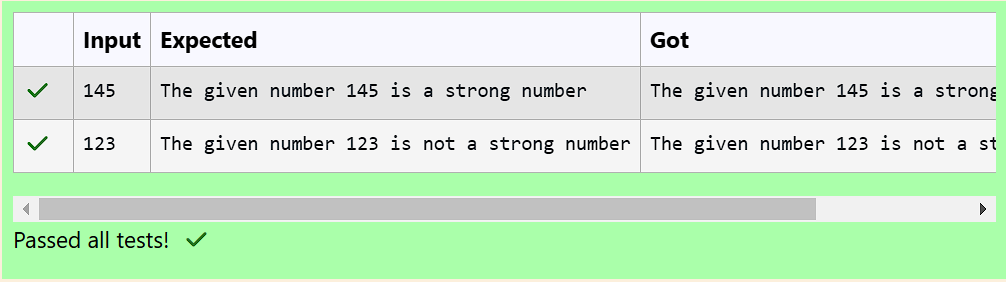
**| 123 | The given number 123 is not a strong number |**

**Program:**

****

****

**Output:**

****