1.text

When an if-else construct appear 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;
      }
}
```

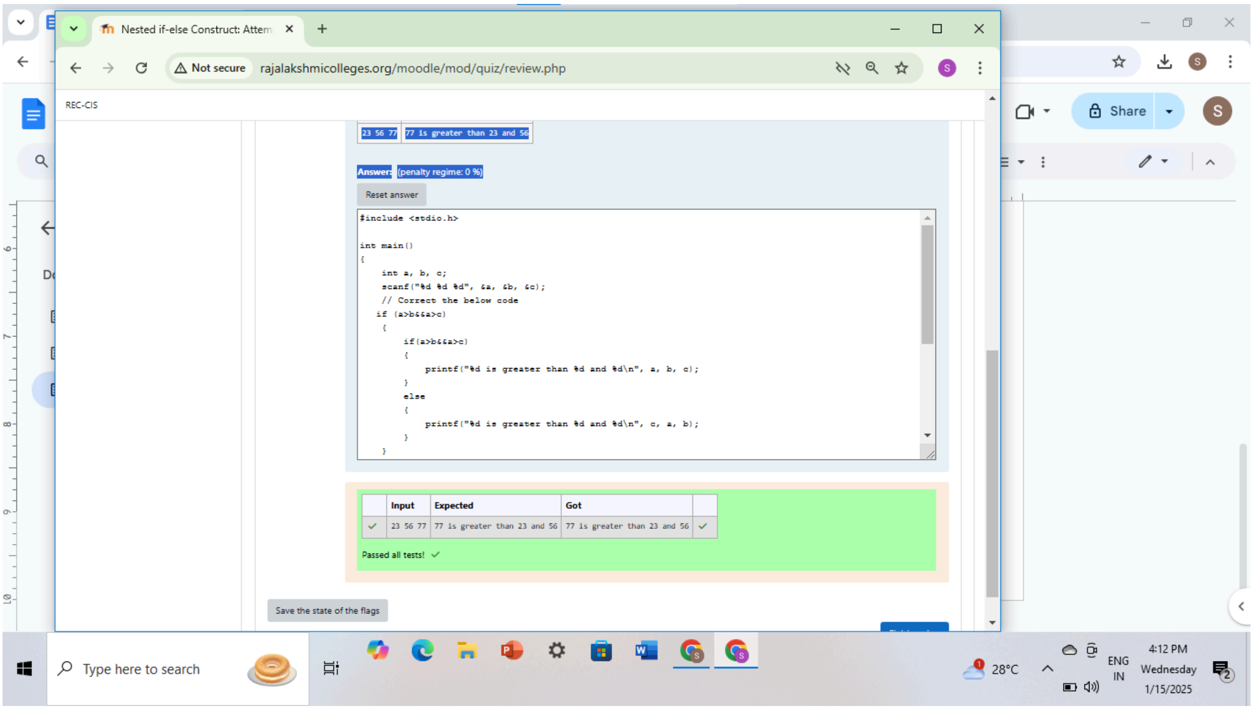
In the above syntax, the statement_2 will be executed only when the conditions in expression_1, expression_2 and expression_3 evaluates to 1 (true).

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

Answer:(penalty regime: 0 %)



Nested if-else Construct: Attem
Not secure
rajalakshmicolleges.org/moodle/mod/quiz/review.php
REC-CIS
Share
Type here to search
28°C
ENG
IN
4:12 PM
Wednesday
1/15/2025

23 56 77	77 is greater than 23 and 56

Answer: (penalty regime: 0 %)

```
#include <stdio.h>

int main()
{
    int a, b, c;
    scanf("%d %d %d", &a, &b, &c);
    // Correct the below code
   if (a>b&&a>c)
    {
        if(a>b&&a>c)
        {
            printf("%d is greater than %d and %d\n", a, b, c);
        }
        else
        {
            printf("%d is greater than %d and %d\n", c, a, b);
        }
    }
```

	Input	Expected	Got	
✓	23 56 77	77 is greater than 23 and 56	77 is greater than 23 and 56	✓

Passed all tests! ✓

Save the state of the flags

2.1.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.

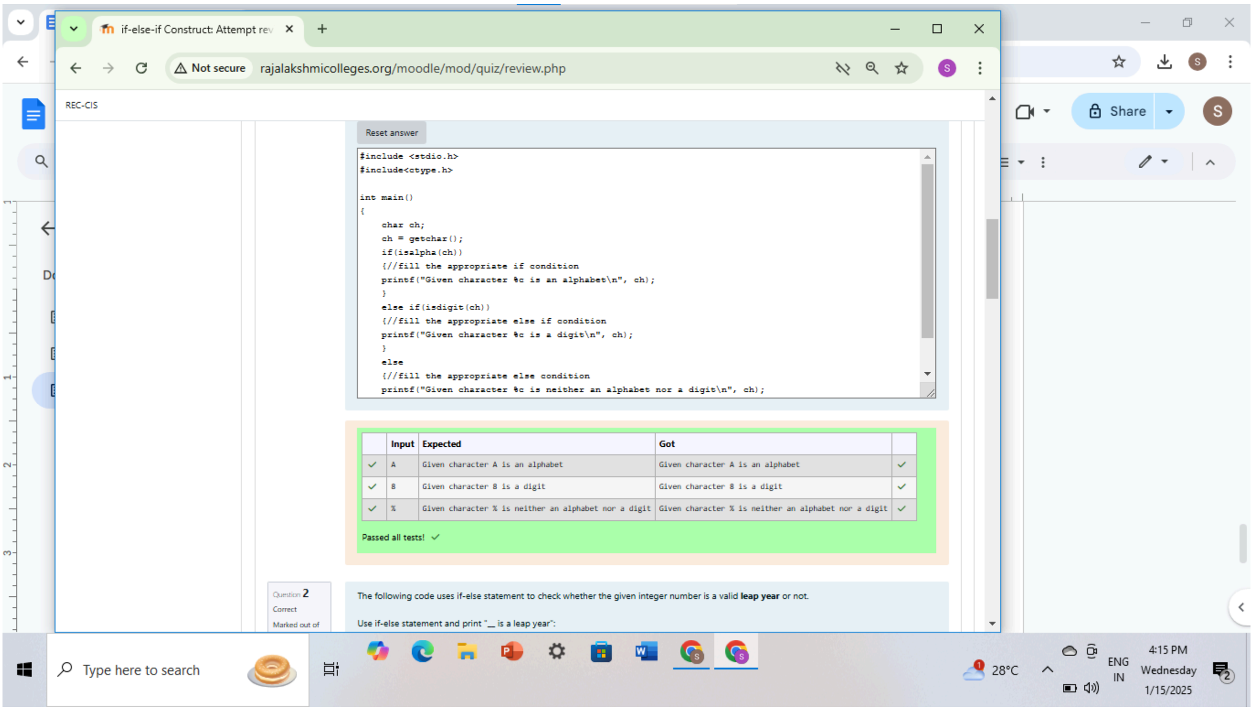
For example, if the condition in the expression_2 is the first condition to evaluate to true after executing statement_2 the control comes out of the complete if-else-if construct.

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

Ans



```
#include <stdio.h>
#include<ctype.h>

int main()
{
    char ch;
    ch = getchar();
    if(isalpha(ch))
    {//fill the appropriate if condition
    printf("Given character %c is an alphabet\n", ch);
    }
    else if(isdigit(ch))
    {//fill the appropriate else if condition
    printf("Given character %c is a digit\n", ch);
    }
    else
    {//fill the appropriate else condition
    printf("Given character %c is neither an alphabet nor a digit\n", ch);
```

	Input	Expected	Got	
✓	A	Given character A is an alphabet	Given character A is an alphabet	✓
✓	8	Given character 8 is a digit	Given character 8 is a digit	✓
✓	%	Given character % is neither an alphabet nor a digit	Given character % is neither an alphabet nor a digit	✓

Passed all tests! ✓

Question 2
Correct
Marked out of

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":

2.2.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.

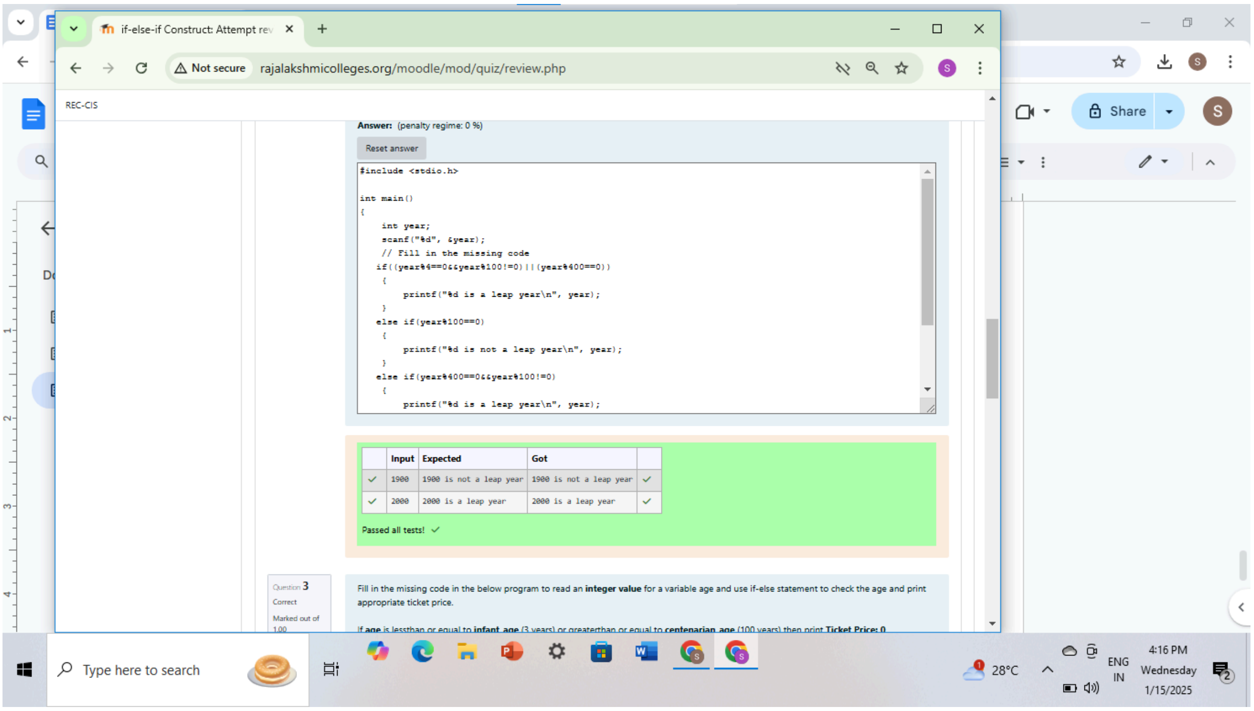
Otherwise, print "__ is not a leap year".

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`

Answer:(penalty regime: 0 %)



Answer: (penalty regime: 0 %)

```
#include <stdio.h>

int main()
{
    int year;
    scanf("%d", &year);
    // Fill in the missing code
   if((year%4==0&&year%100!=0)||(year%400==0))
    {
        printf("%d is a leap year\n", year);
    }
   else if(year%100==0)
    {
        printf("%d is not a leap year\n", year);
    }
   else if(year%400==0&&year%100!=0)
    {
        printf("%d is a leap year\n", year);
```

	Input	Expected	Got	
✓	1900	1900 is not a leap year	1900 is not a leap year	✓
✓	2000	2000 is a leap year	2000 is a leap year	✓

Passed all tests! ✓

Question 3
Correct
Marked out of 1.00

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

If **age** is lessthan or equal to **infant_age** (3 years) or greaterthan or equal to **centenarian_age** (100 years) then print **Ticket Price: 0**

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

If age is lessthan or equal to infant_age (3 years) or greaterthan or equal to centenarian_age (100 years) then print Ticket Price: 0.

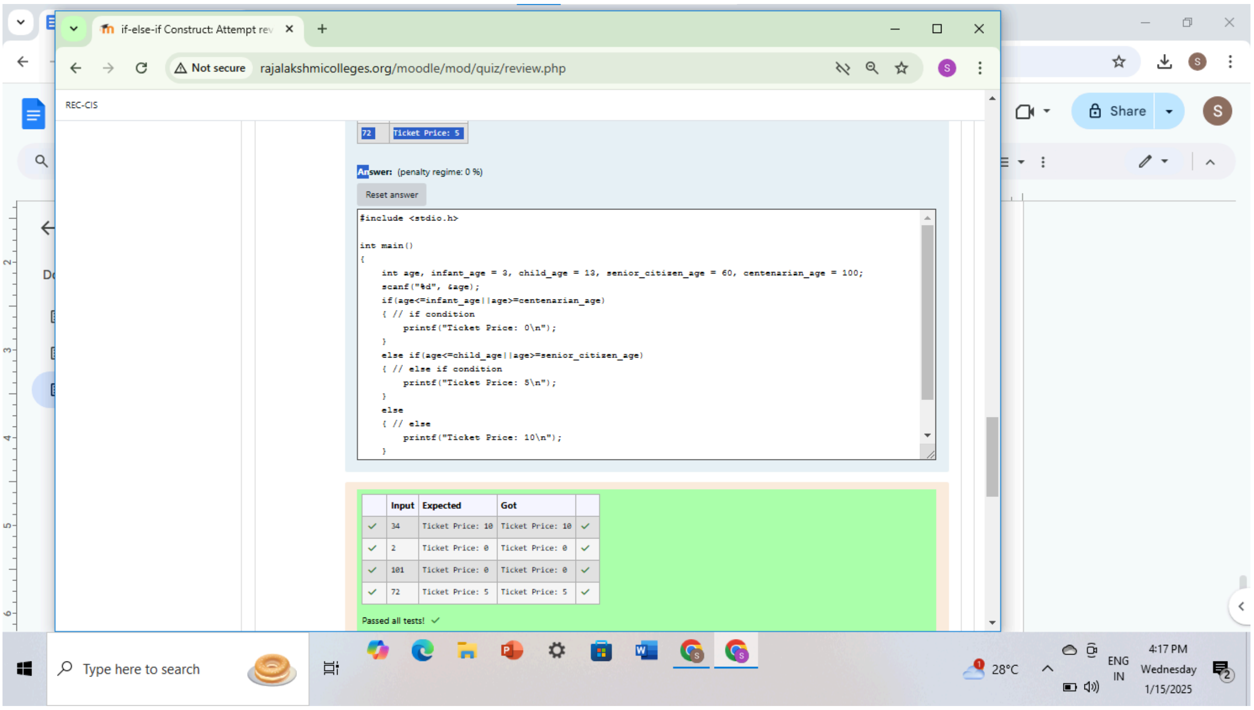
Otherwise, If age is lessthan or equal to child_age (13 years) or greaterthan or equal to senior_citizen_age (60 years) then print Ticket Price: 5.

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

An



72	Ticket Price: 5

Answer: (penalty regime: 0 %)

```
#include <stdio.h>

int main()
{
    int age, infant_age = 3, child_age = 13, senior_citizen_age = 60, centenarian_age = 100;
    scanf("%d", &age);
    if(age<=infant_age||age>=centenarian_age)
    { // if condition
        printf("Ticket Price: 0\n");
    }
    else if(age<=child_age||age>=senior_citizen_age)
    { // else if condition
        printf("Ticket Price: 5\n");
    }
    else
    { // else
        printf("Ticket Price: 10\n");
    }
```

	Input	Expected	Got	
✓	34	Ticket Price: 10	Ticket Price: 10	✓
✓	2	Ticket Price: 0	Ticket Price: 0	✓
✓	101	Ticket Price: 0	Ticket Price: 0	✓
✓	72	Ticket Price: 5	Ticket Price: 5	✓

Passed all tests! ✓

2.4See 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.

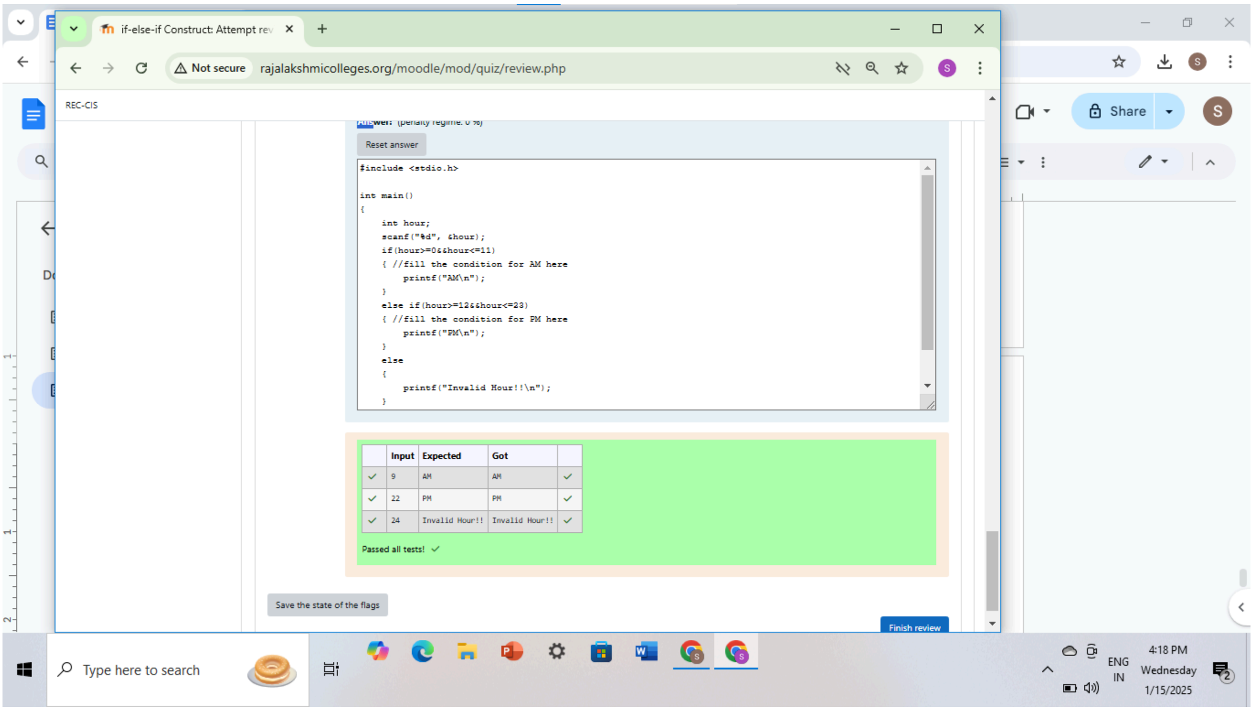
Fill in the if condition to check if the given hour is between 0 and 11 (both inclusive) for AM. Fill in the else if condition to check if the given hour is between 12 and 23 (both inclusive) for PM.

For example:

Input	Result
9	AM
22	PM

24	Invalid Hour!!

Ans


if-else-if Construct: Attempt rev
Not secure
rajalakshmicolleges.org/moodle/mod/quiz/review.php
REC-CIS
Reset answer
Save the state of the flags
Finish review
Type here to search
Share
4:18 PM
Wednesday
1/15/2025

```
#include <stdio.h>

int main()
{
    int hour;
    scanf("%d", &hour);
    if(hour>=0&&hour<=11)
    { //fill the condition for AM here
        printf("AM\n");
    }
    else if(hour>=12&&hour<=23)
    { //fill the condition for PM here
        printf("PM\n");
    }
    else
    {
        printf("Invalid Hour!!\n");
    }
```

	Input	Expected	Got	
✓	9	AM	AM	✓
✓	22	PM	PM	✓
✓	24	Invalid Hour!!	Invalid Hour!!	✓

Passed all tests! ✓

3.1A 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.

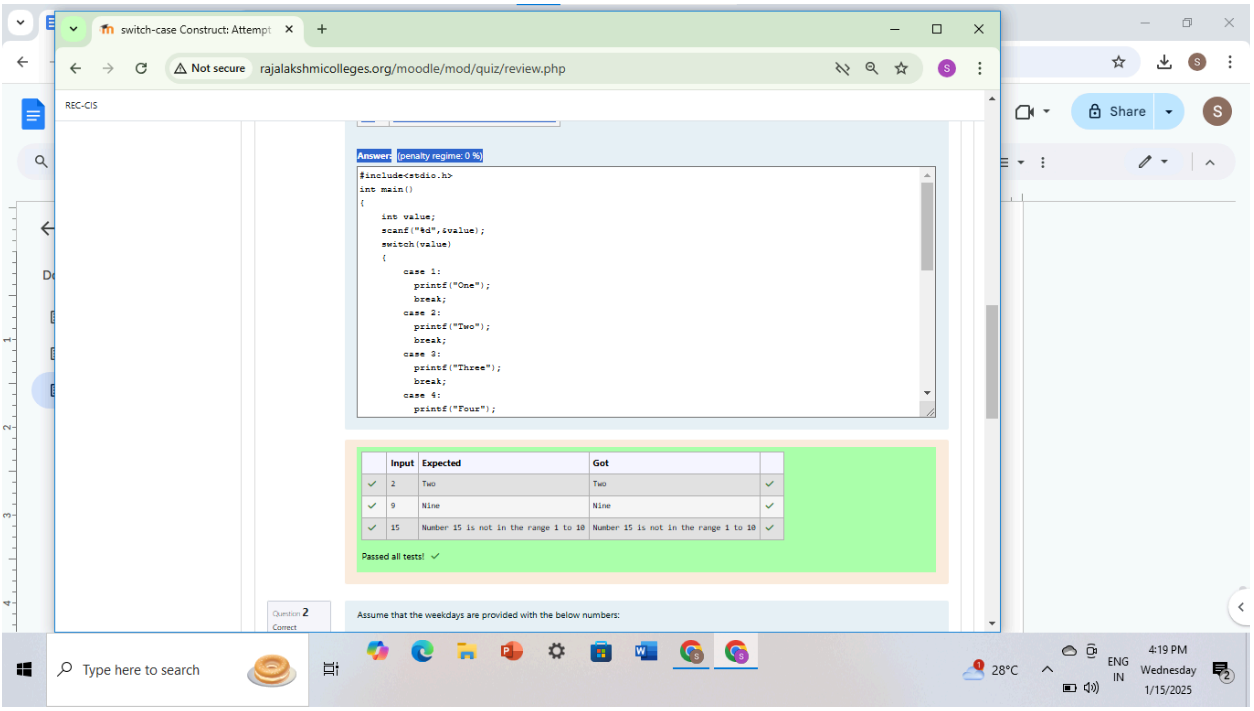
```
#include <stdio.h>
int main()
{
        int value;
```

```
      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

Answer:(penalty regime: 0 %)



Answer: (penalty regime: 0 %)

```
#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");
```

	Input	Expected	Got	
✓	2	Two	Two	✓
✓	9	Nine	Nine	✓
✓	15	Number 15 is not in the range 1 to 10	Number 15 is not in the range 1 to 10	✓

Passed all tests! ✓

Assume that the weekdays are provided with the below numbers:

3.2.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:

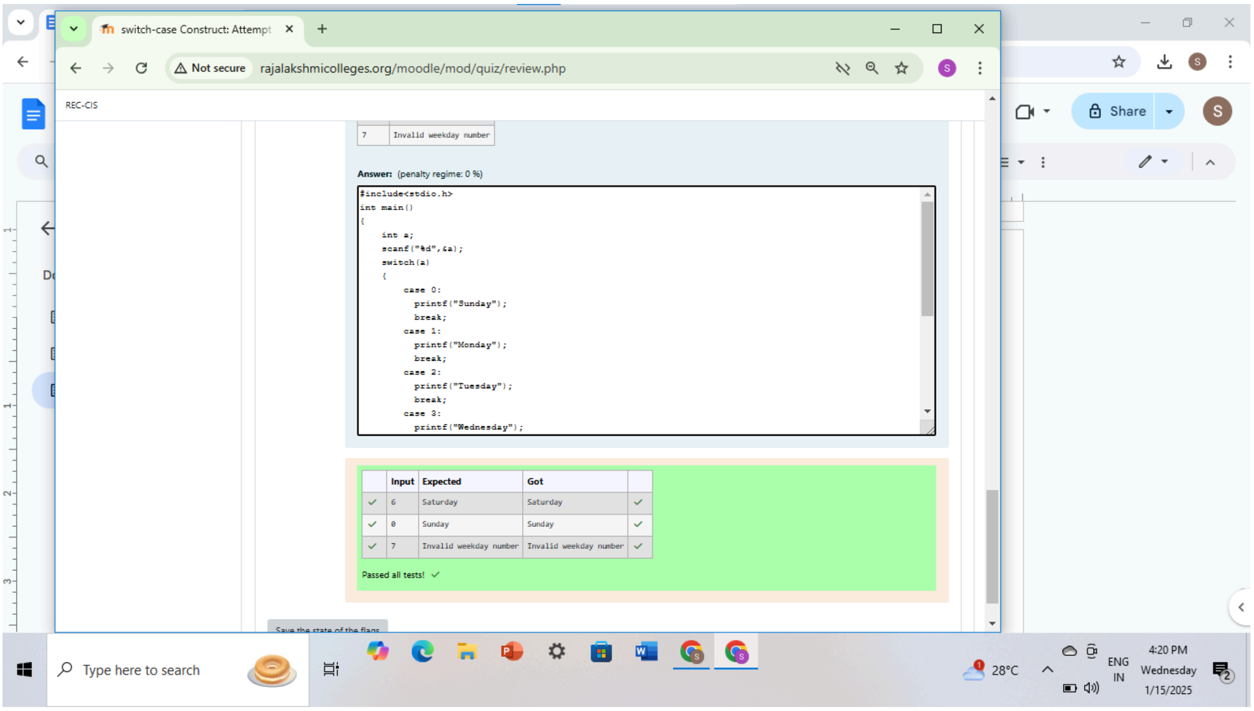
```
Invalid weekday number
```

For example:

Input	Result

6	Saturday
0	Sunday
7	Invalid weekday number

Answer:(penalty regime: 0 %)



switch-case Construct: Attempt
Not secure
rajalakshmicolleges.org/moodle/mod/quiz/review.php
REC-CIS
Share
Type here to search
28°C
ENG
IN
4:20 PM
Wednesday
1/15/2025

7	Invalid weekday number

Answer: (penalty regime: 0 %)

```
#include<stdio.h>
int main()
{
    int a;
    scanf("%d",&a);
    switch(a)
    {
        case 0:
          printf("Sunday");
          break;
        case 1:
          printf("Monday");
          break;
        case 2:
          printf("Tuesday");
          break;
        case 3:
          printf("Wednesday");
```

	Input	Expected	Got	
✓	6	Saturday	Saturday	✓
✓	0	Sunday	Sunday	✓
✓	7	Invalid weekday number	Invalid weekday number	✓

Passed all tests! ✓

4.1.Most of the programming languages provide a special construct/statement using which we can repeatedly execute one or more statement 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 is evaluated and only if the condition evaluates to true the body of the loop is executed. After executing the body the control is automatically 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.

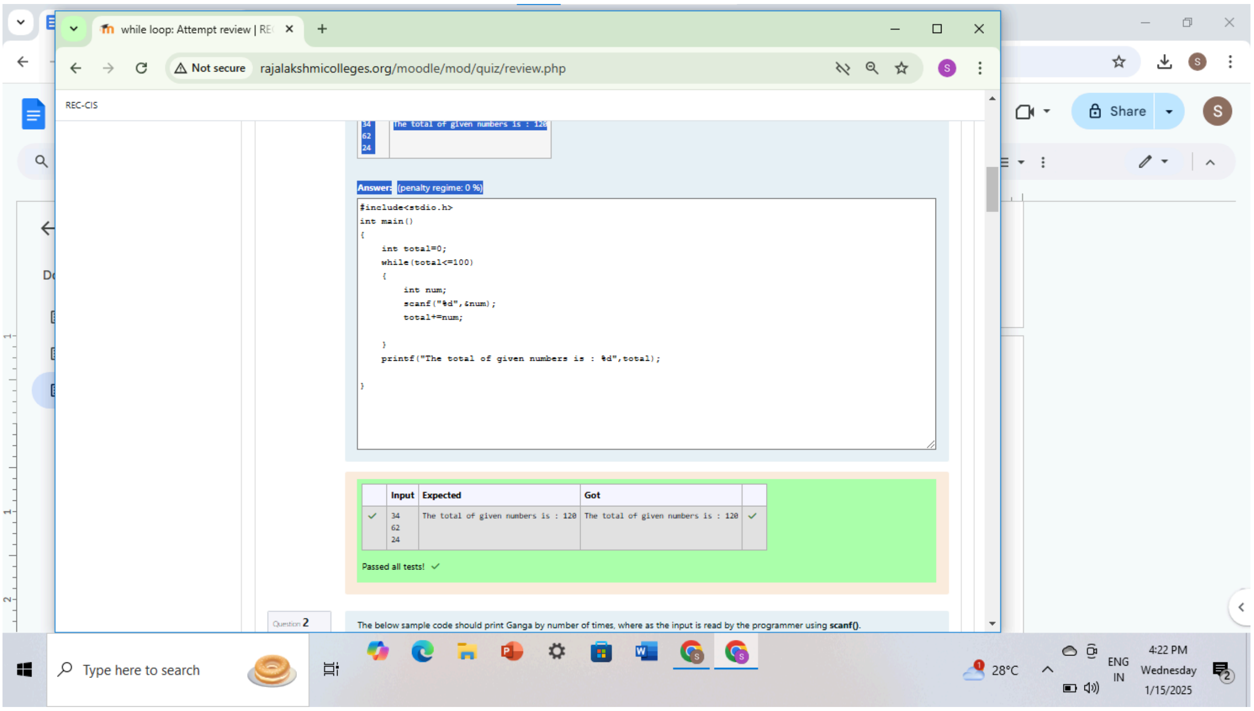
```
#include <stdio.h>

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 62 24	The total of given numbers is : 120

Answer:(penalty regime: 0 %)



34 62 24	The total of given numbers is : 120

Answer: (penalty regime: 0 %)

```
#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);

}
```

	Input	Expected	Got	
✓	34 62 24	The total of given numbers is : 120	The total of given numbers is : 120	✓

Passed all tests! ✓

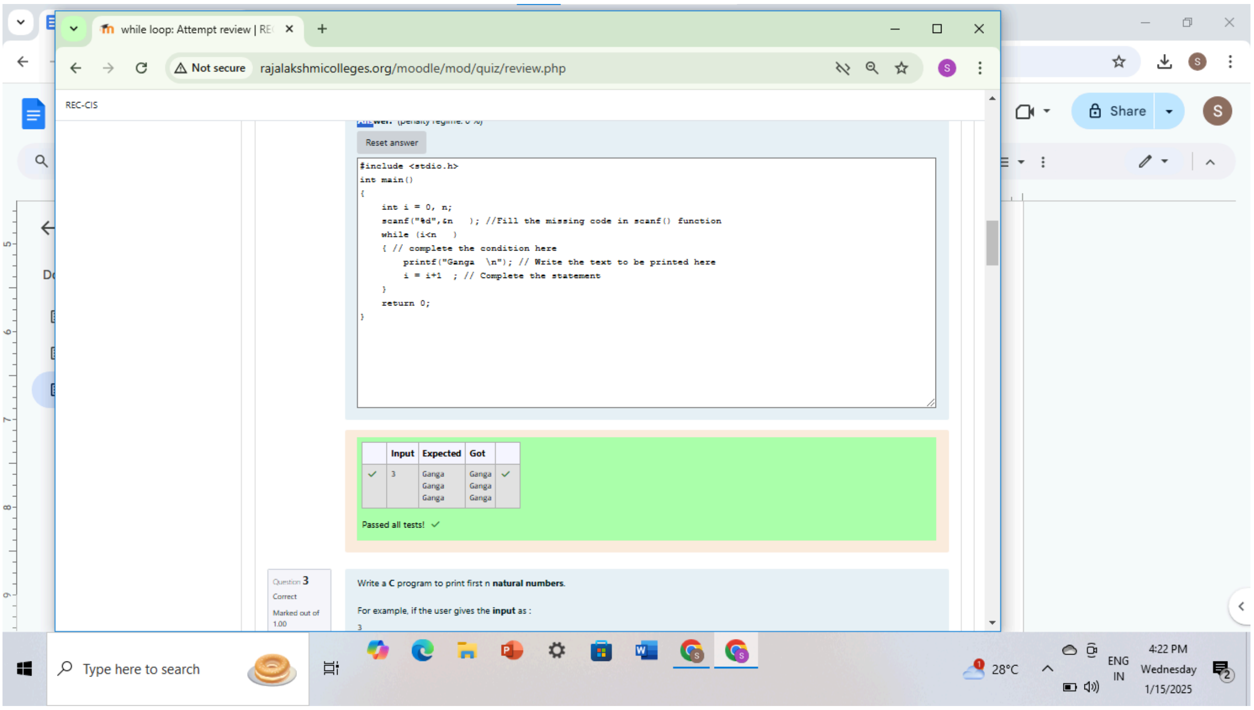
The below sample code should print Ganga by number of times, where as the input is read by the programmer using **scanf()**.

4.2.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

Ans


while loop: Attempt review | RE
Not secure
rajalakshmicolleges.org/moodle/mod/quiz/review.php
REC-CIS
Share
Type here to search
28°C
ENG
IN
4:22 PM
Wednesday
1/15/2025

Answer: (penalty regime: 0 %)

Reset answer

```
#include <stdio.h>
int main()
{
    int i = 0, n;
    scanf("%d",&n   ); //Fill the missing code in scanf() function
    while (i<n   )
    { // complete the condition here
        printf("Ganga  \n"); // Write the text to be printed here
        i = i+1  ; // Complete the statement
    }
    return 0;
}
```

	Input	Expected	Got	
✓	3	Ganga Ganga Ganga	Ganga Ganga Ganga	✓

Passed all tests! ✓

Question 3
Correct
Marked out of 1.00

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

For example, if the user gives the **input** as :
3

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

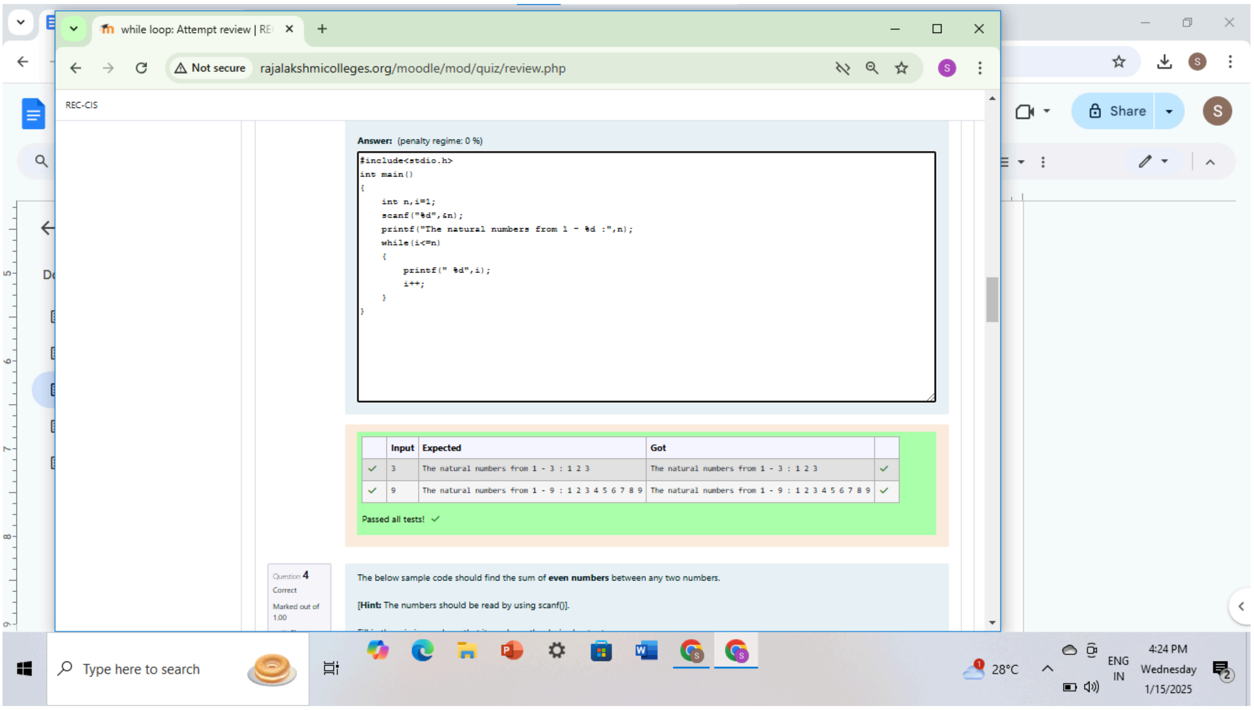
For example, if the user gives the input as :
3
then the program should print the result as:

```
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

Answ



REC-CIS

Answer: (penalty regime: 0 %)

```
#include<stdio.h>
int main()
{
    int n,i=1;
    scanf("%d",&n);
    printf("The natural numbers from 1 - %d :",n);
    while(i<=n)
    {
        printf(" %d",i);
        i++;
    }
}
```

	Input	Expected	Got	
✓	3	The natural numbers from 1 - 3 : 1 2 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	The natural numbers from 1 - 9 : 1 2 3 4 5 6 7 8 9	✓

Passed all tests! ✓

Question 4
Correct
Marked out of 1.00

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

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

while loop: Attempt review | RE
Not secure
rajalakshmicolleges.org/moodle/mod/quiz/review.php
Share
Type here to search
28°C
ENG
IN
4:24 PM
Wednesday
1/15/2025

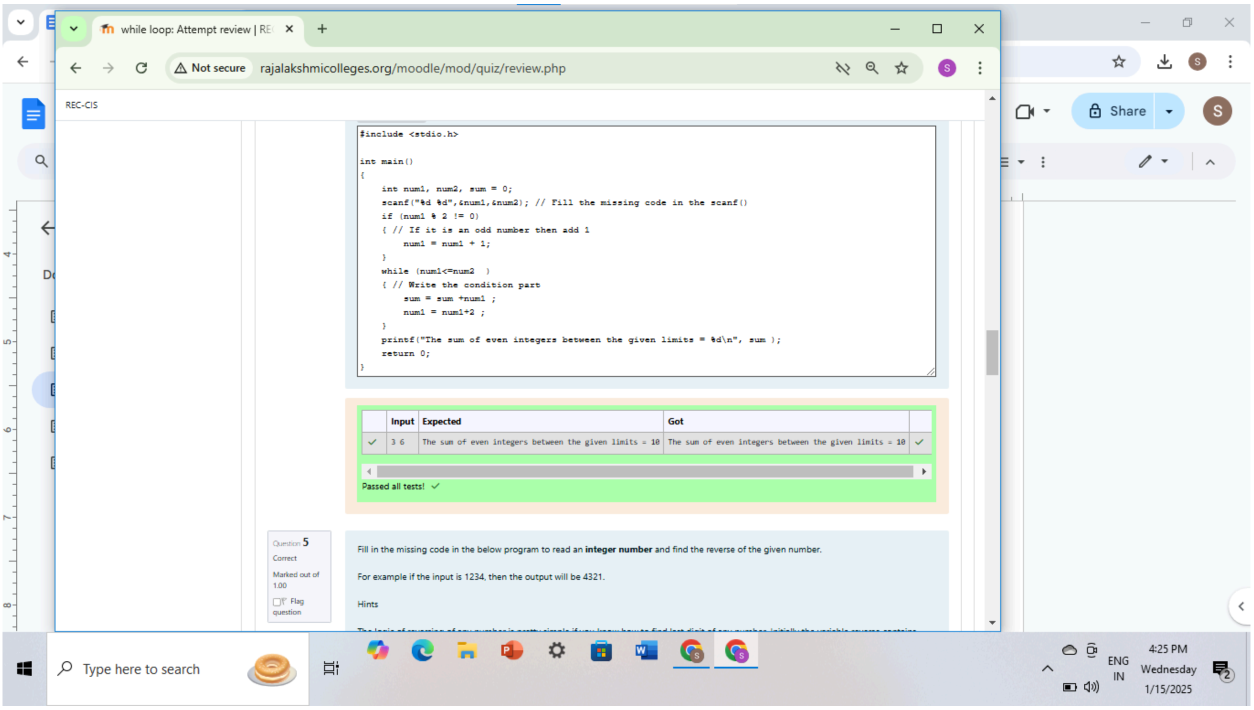
4.4.The below sample code should find the sum of even numbers between any two numbers.

[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

Answer:(penalty regime: 0 %)



```
#include <stdio.h>

int main()
{
    int num1, num2, sum = 0;
    scanf("%d %d",&num1,&num2); // Fill the missing code in the scanf()
    if (num1 % 2 != 0)
    { // If it is an odd number then add 1
        num1 = num1 + 1;
    }
    while (num1<=num2  )
    { // Write the condition part
        sum = sum +num1 ;
        num1 = num1+2 ;
    }
    printf("The sum of even integers between the given limits = %d\n", sum );
    return 0;
}
```

	Input	Expected	Got	
✓	3 6	The sum of even integers between the given limits = 10	The sum of even integers between the given limits = 10	✓

Passed all tests! ✓

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

4.5.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.

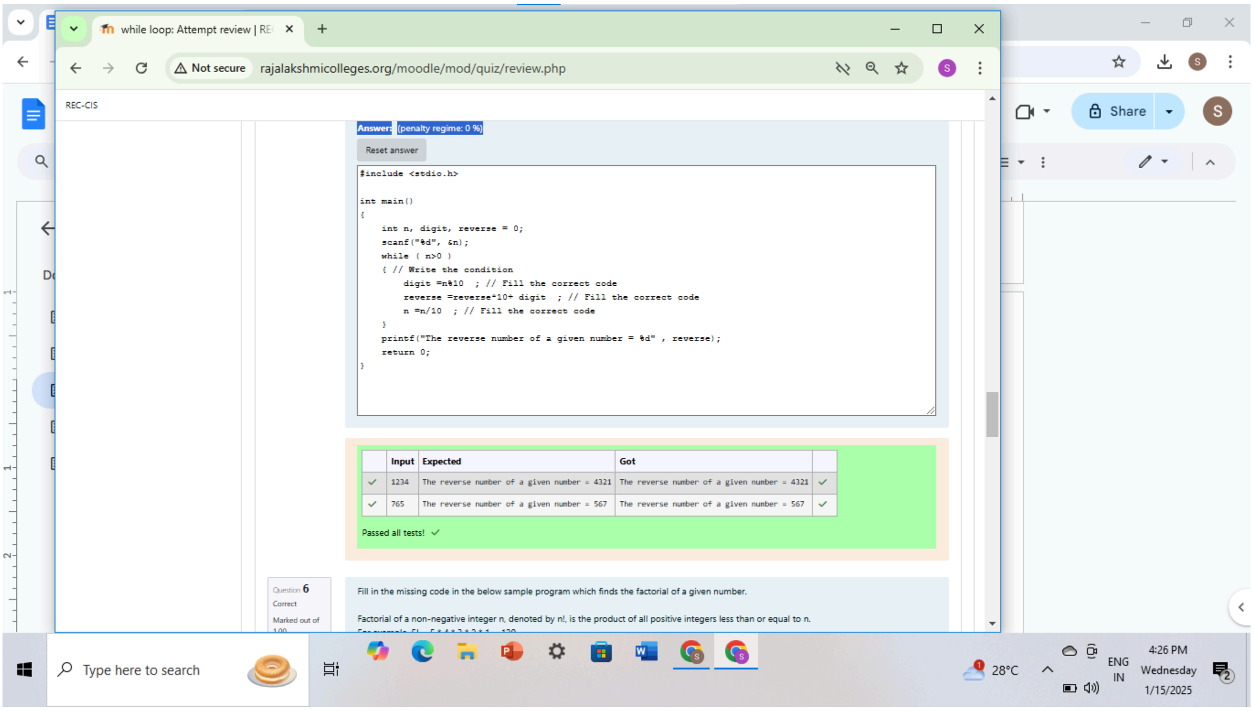
Repeat the above four steps till the original number becomes 0 and finally we will be left with the reversed number in reverse variable.

For example:

Input	**Result**

1234	The reverse number of a given number = 4321
765	The reverse number of a given number = 567

Answer:(penalty regime: 0 %)



while loop: Attempt review | RE
Not secure
rajalakshmicolleges.org/moodle/mod/quiz/review.php
REC-CIS
Share
Answer: (penalty regime: 0 %)
Reset answer
Type here to search
28°C
ENG
IN
4:26 PM
Wednesday
1/15/2025

```
#include <stdio.h>

int main()
{
    int n, digit, reverse = 0;
    scanf("%d", &n);
    while ( n>0 )
    { // Write the condition
        digit =n%10  ; // Fill the correct code
        reverse =reverse*10+ digit  ; // Fill the correct code
        n =n/10  ; // Fill the correct code
    }
    printf("The reverse number of a given number = %d" , reverse);
    return 0;
}
```

	Input	Expected	Got	
✓	1234	The reverse number of a given number = 4321	The reverse number of a given number = 4321	✓
✓	765	The reverse number of a given number = 567	The reverse number of a given number = 567	✓

Passed all tests! ✓

Question 6
Correct
Marked out of 1.00

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.

4.6.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.

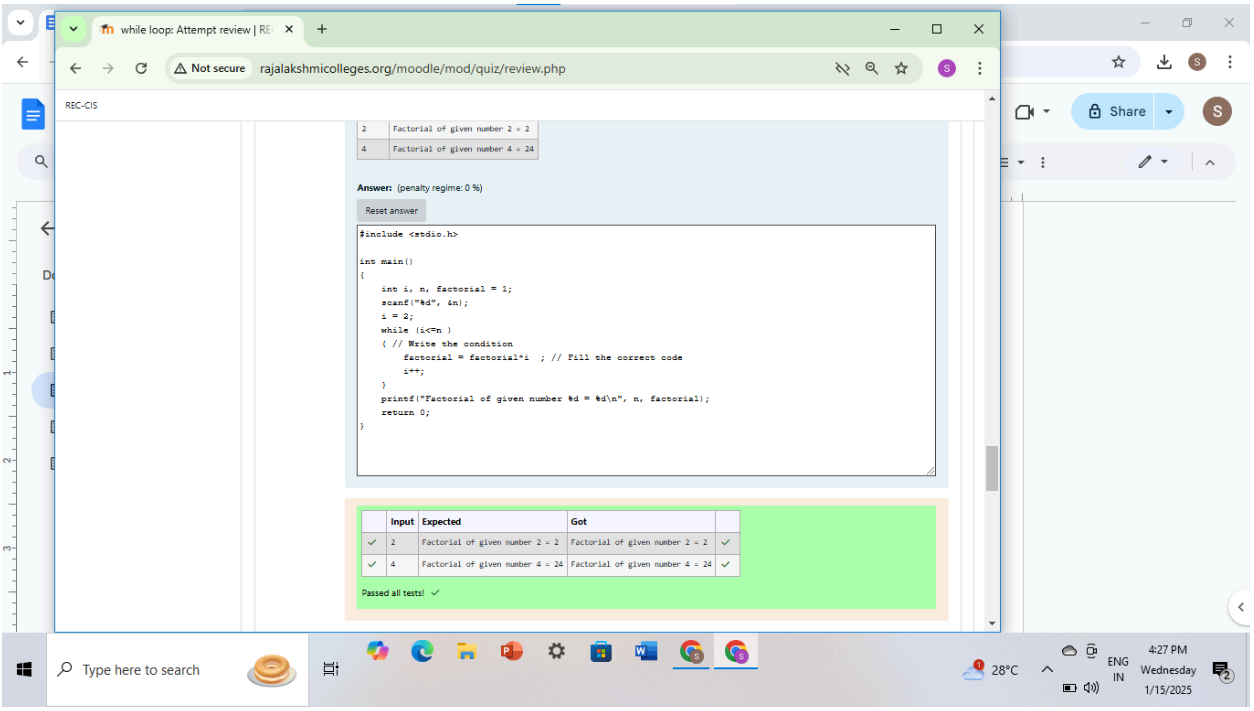
The main() function declares an integer variable factorial and initializes it to 1, which it will use to store the computed factorial value.

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

For example:

Input	Result
2	Factorial of given number 2 = 2
4	Factorial of given number 4 = 24

An



while loop: Attempt review | RE
Not secure
rajalakshmicolleges.org/moodle/mod/quiz/review.php
REC-CIS
Share
Type here to search
28°C
ENG
IN
4:27 PM
Wednesday
1/15/2025

2	Factorial of given number 2 = 2
4	Factorial of given number 4 = 24

Answer: (penalty regime: 0 %)

Reset answer

```
#include <stdio.h>

int main()
{
    int i, n, factorial = 1;
    scanf("%d", &n);
    i = 2;
    while (i<=n )
    { // Write the condition
        factorial = factorial*i  ; // Fill the correct code
        i++;
    }
    printf("Factorial of given number %d = %d\n", n, factorial);
    return 0;
}
```

	Input	Expected	Got	
✓	2	Factorial of given number 2 = 2	Factorial of given number 2 = 2	✓
✓	4	Factorial of given number 4 = 24	Factorial of given number 4 = 24	✓

Passed all tests! ✓

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

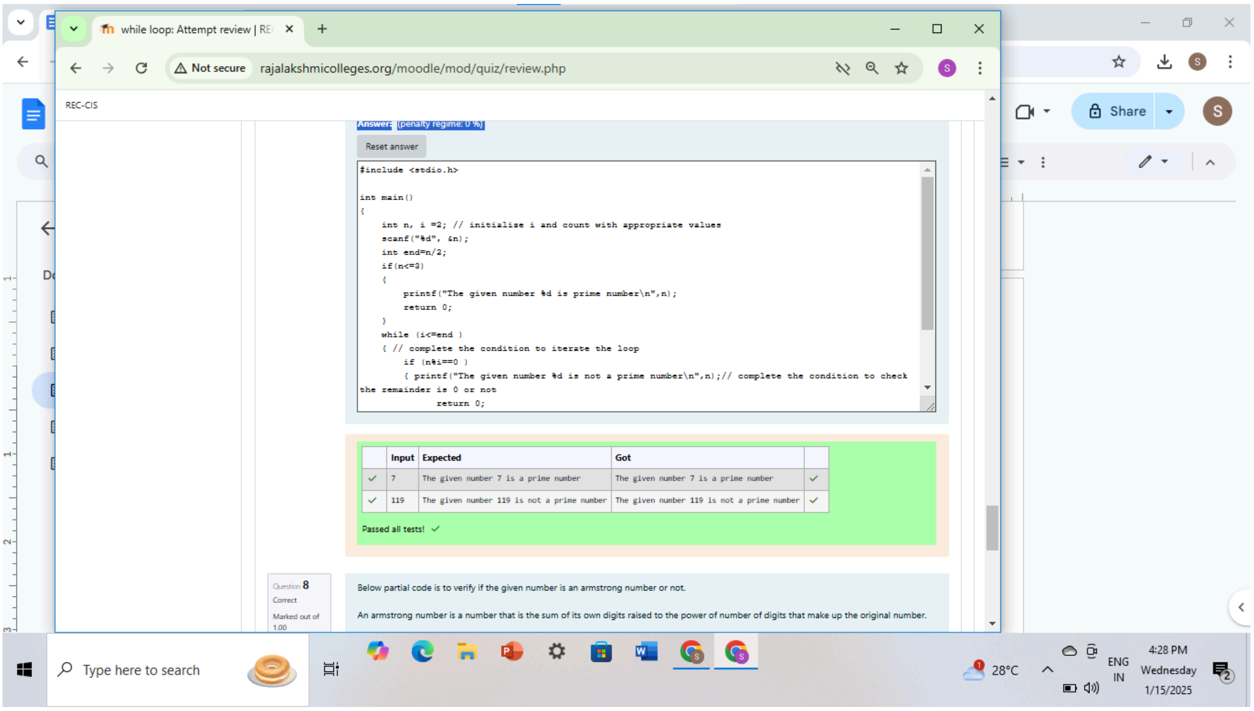
A prime number is a positive integer greater than 1, which is not divisible by any other number other than 1 and itself. Examples of a few prime numbers are 2, 3, 5, 7, 11, 13, 17, 19, etc.

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

Answer:(penalty regime: 0 %)



while loop: Attempt review | RE
Not secure
rajalakshmicolleges.org/moodle/mod/quiz/review.php
REC-CIS
Answer: (penalty regime: 0 %)
Reset answer
Share
Type here to search
28°C
ENG IN
4:28 PM
Wednesday
1/15/2025

```
#include <stdio.h>

int main()
{
    int n, i =2; // initialize i and count with appropriate values
    scanf("%d", &n);
    int end=n/2;
    if(n<=3)
    {
        printf("The given number %d is prime number\n",n);
        return 0;
    }
    while (i<=end )
    { // complete the condition to iterate the loop
        if (n%i==0 )
        { printf("The given number %d is not a prime number\n",n);// complete the condition to check
the remainder is 0 or not
              return 0;
```

	Input	Expected	Got	
✓	7	The given number 7 is a prime number	The given number 7 is a prime number	✓
✓	119	The given number 119 is not a prime number	The given number 119 is not a prime number	✓

Passed all tests! ✓

Question 8
Correct
Marked out of 1.00

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 that make up the original number.

4.8.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 that make up 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 ($1^3 + 5^3 + 3^3$) 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 the power of 4 of each digit ($9^4 + 4^4 + 7^4 + 4^4$) is equal to the same number 9474. Such a number is known as an armstrong number.

Similarly,

$9 = 9_1 = 9$

$371 = 3_3 + 7_3 + 1_3 = 27 + 343 +1 = 371$

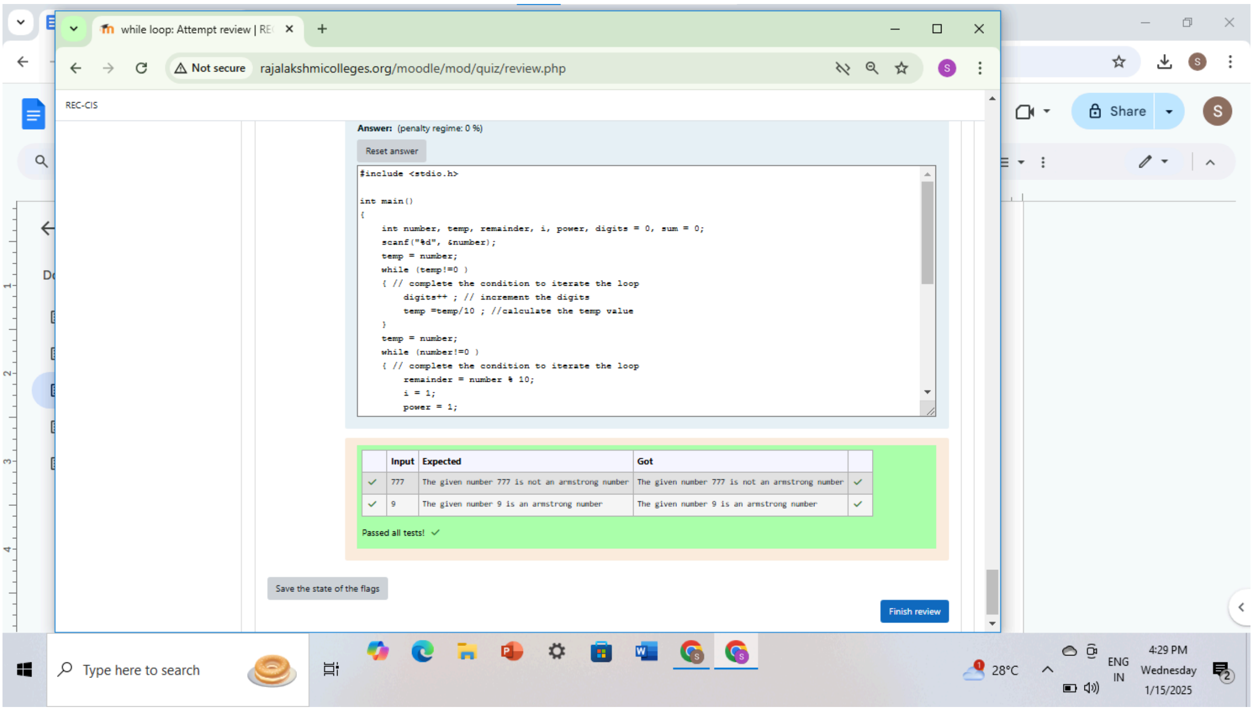
$3\ 8208 = 8_4 + 2_4+0_4 + 8_4 = 4096 + 16 + 0 + 4096 = 8208$

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

Answer:(penalty regime: 0 %)


while loop: Attempt review | RE
Not secure
rajalakshmicolleges.org/moodle/mod/quiz/review.php
REC-CIS
Reset answer
Save the state of the flags
Finish review
Type here to search
28°C
ENG
IN
4:29 PM
Wednesday
1/15/2025

Answer: (penalty regime: 0 %)

```
#include <stdio.h>

int main()
{
    int number, temp, remainder, i, power, digits = 0, sum = 0;
    scanf("%d", &number);
    temp = number;
    while (temp!=0 )
    { // complete the condition to iterate the loop
        digits++ ; // increment the digits
        temp =temp/10 ; //calculate the temp value
    }
    temp = number;
    while (number!=0 )
    { // complete the condition to iterate the loop
        remainder = number % 10;
        i = 1;
        power = 1;
```

	Input	Expected	Got	
✓	777	The given number 777 is not an armstrong number	The given number 777 is not an armstrong number	✓
✓	9	The given number 9 is an armstrong number	The given number 9 is an armstrong number	✓

Passed all tests! ✓

5.8.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);
}
```

1. The initialization expression initializes the loop counter; it is executed once at the start of the loop.
2. The loop continues to execute as long as the condition expression evaluates to true.
3. 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);
}
```

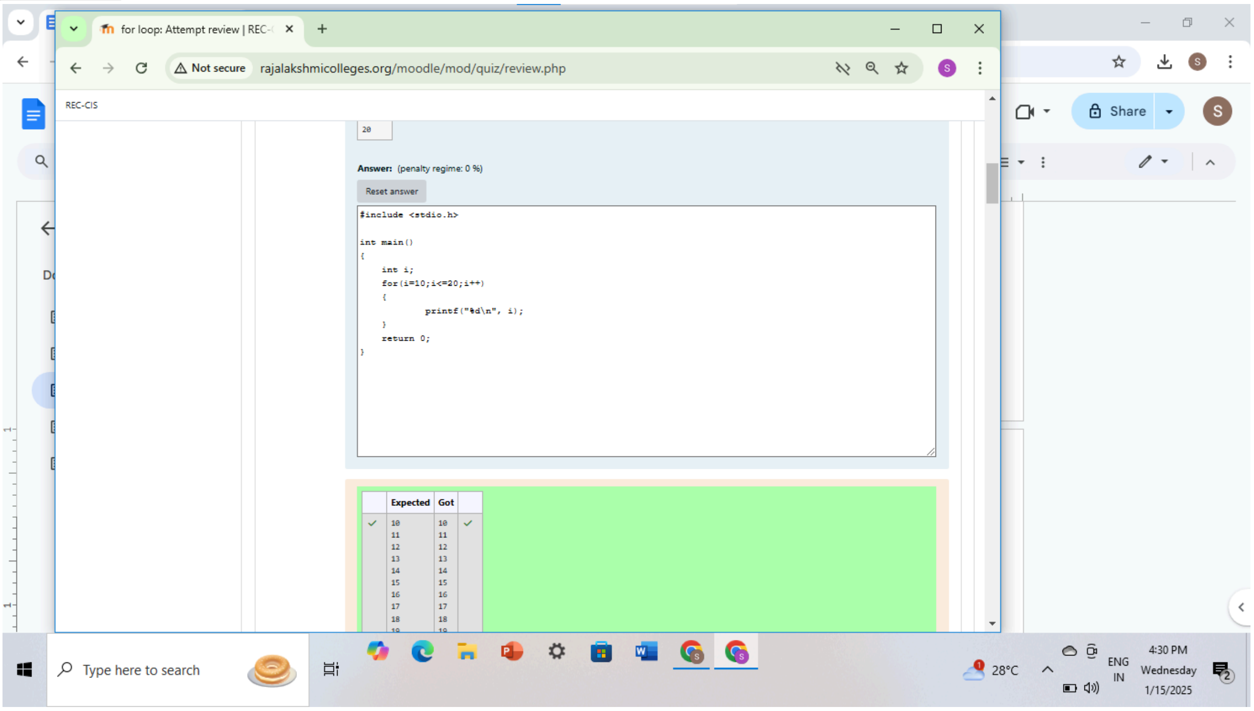
1. Above for-loop statement initializes an integer variable i (which is the loop counter) as part of the initialization expression.
2. In the update section, it increments the variable i by 1 using the post-increment expression i++.
3. The expression in condition is i < 10. The for-loop keeps on executing the code inside the loop body as long as this condition evaluates to true. And the loop terminates when the condition evaluates to false.
4. 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

1


for loop: Attempt review | REC-
Not secure
rajalakshmicolleges.org/moodle/mod/quiz/review.php
REC-CIS
Share
Type here to search
28°C
ENG
IN
4:30 PM
Wednesday
1/15/2025

20

Answer: (penalty regime: 0 %)

Reset answer

```
#include <stdio.h>

int main()
{
    int i;
    for(i=10;i<=20;i++)
    {
            printf("%d\n", i);
    }
    return 0;
}
```

	Expected	Got	
✓	10 11 12 13 14 15 16 17 18 19	10 11 12 13 14 15 16 17 18 19	✓

5.2.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.

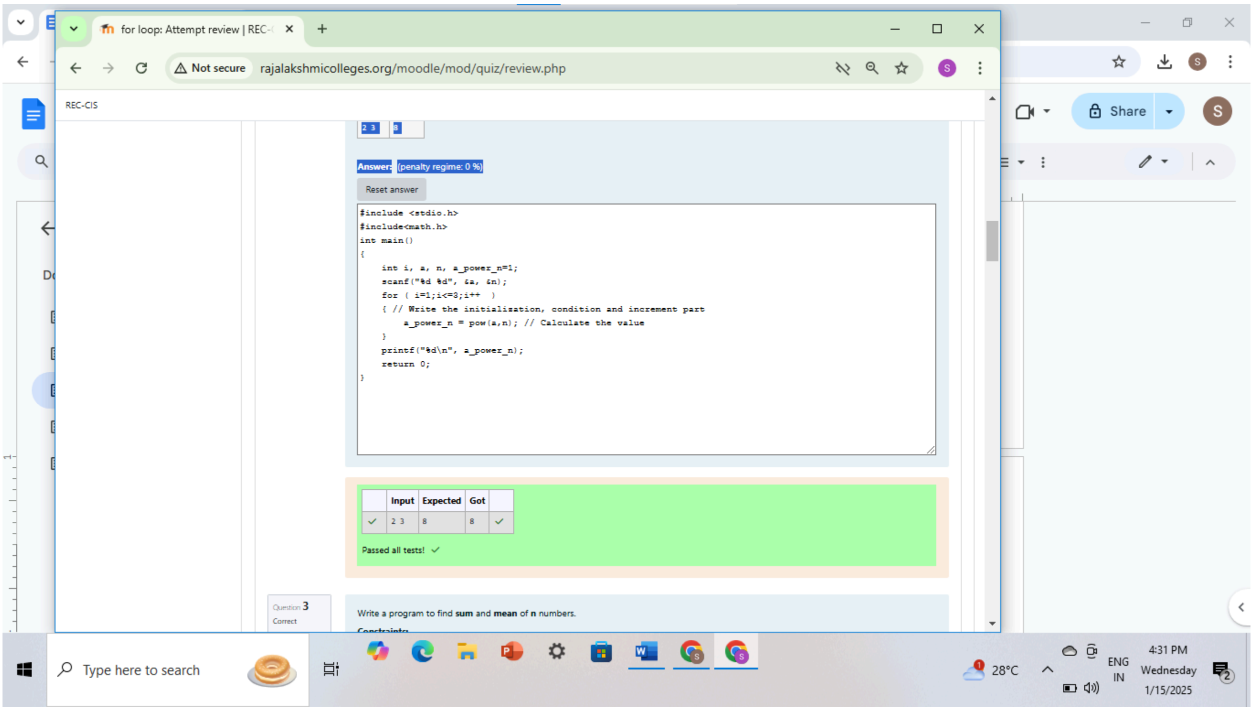
Variable a_power_n is used to store the computed value of a$_n$.

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

Answer:(penalty regime: 0 %)



2 3	8

Answer: (penalty regime: 0 %)

```
#include <stdio.h>
#include<math.h>
int main()
{
    int i, a, n, a_power_n=1;
    scanf("%d %d", &a, &n);
    for ( i=1;i<=3;i++  )
    { // Write the initialization, condition and increment part
        a_power_n = pow(a,n); // Calculate the value
    }
    printf("%d\n", a_power_n);
    return 0;
}
```

	Input	Expected	Got	
✓	2 3	8	8	✓

Passed all tests! ✓

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

Constraints:

5.3.Write a program to find sum and mean of n numbers.
Constraints:

- 1 <= n <= 106
- 10-3 <= elements <= 103
- Result of mean should print upto 2 decimal places.

Sample test case:

4------------------------> First line of input is the value on 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.

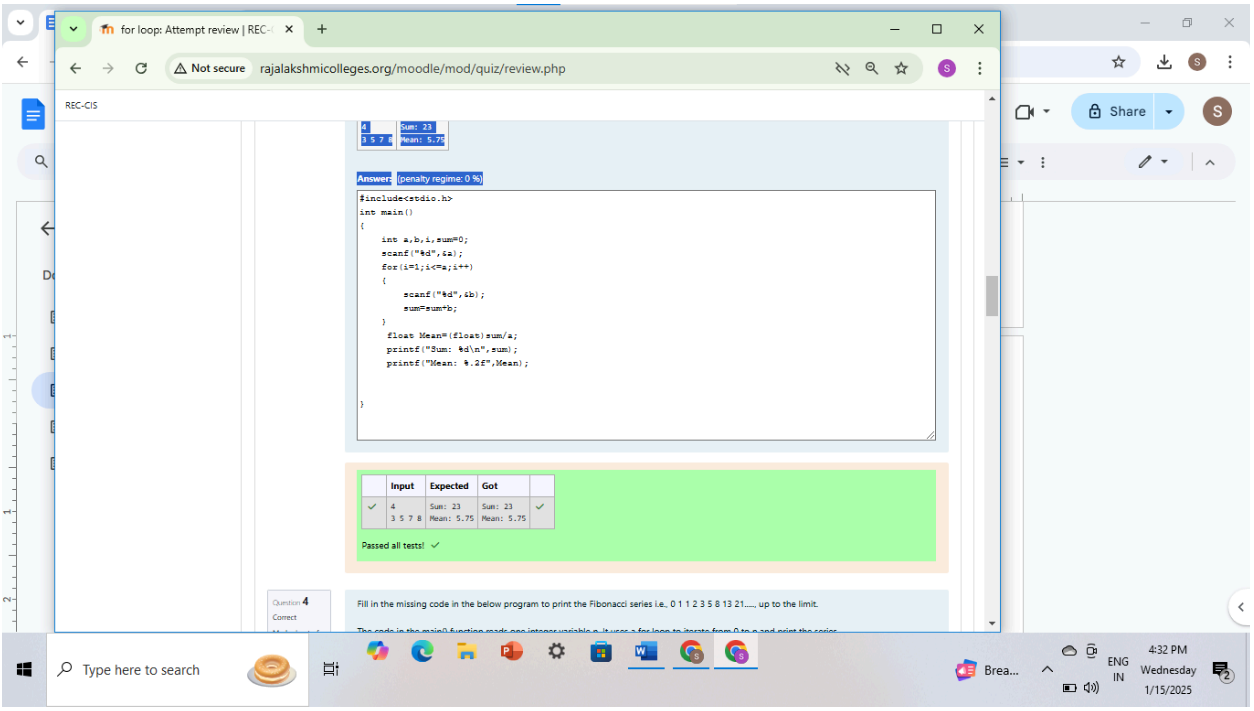
Instruction: To run your custom test cases strictly map your input and output layout with the visible test cases.

For example:

Input	Result

4
3 5 7
8
	Sum: 23
	Mean:
	5.75

Answer:(penalty regime: 0 %)



for loop: Attempt review | REC-
Not secure
rajalakshmicolleges.org/moodle/mod/quiz/review.php
REC-CIS
Share
Type here to search
Brea...
ENG
IN
4:32 PM
Wednesday
1/15/2025

4 3 5 7 8	Sum: 23 Mean: 5.75

Answer: (penalty regime: 0 %)

```
#include<stdio.h>
int main()
{
    int a,b,i,sum=0;
    scanf("%d",&a);
    for(i=1;i<=a;i++)
    {
        scanf("%d",&b);
        sum=sum+b;
    }
     float Mean=(float)sum/a;
     printf("Sum: %d\n",sum);
     printf("Mean: %.2f",Mean);


}
```

	Input	Expected	Got	
✓	4 3 5 7 8	Sum: 23 Mean: 5.75	Sum: 23 Mean: 5.75	✓

Passed all tests! ✓

Question 4
Correct

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 limit.

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 series

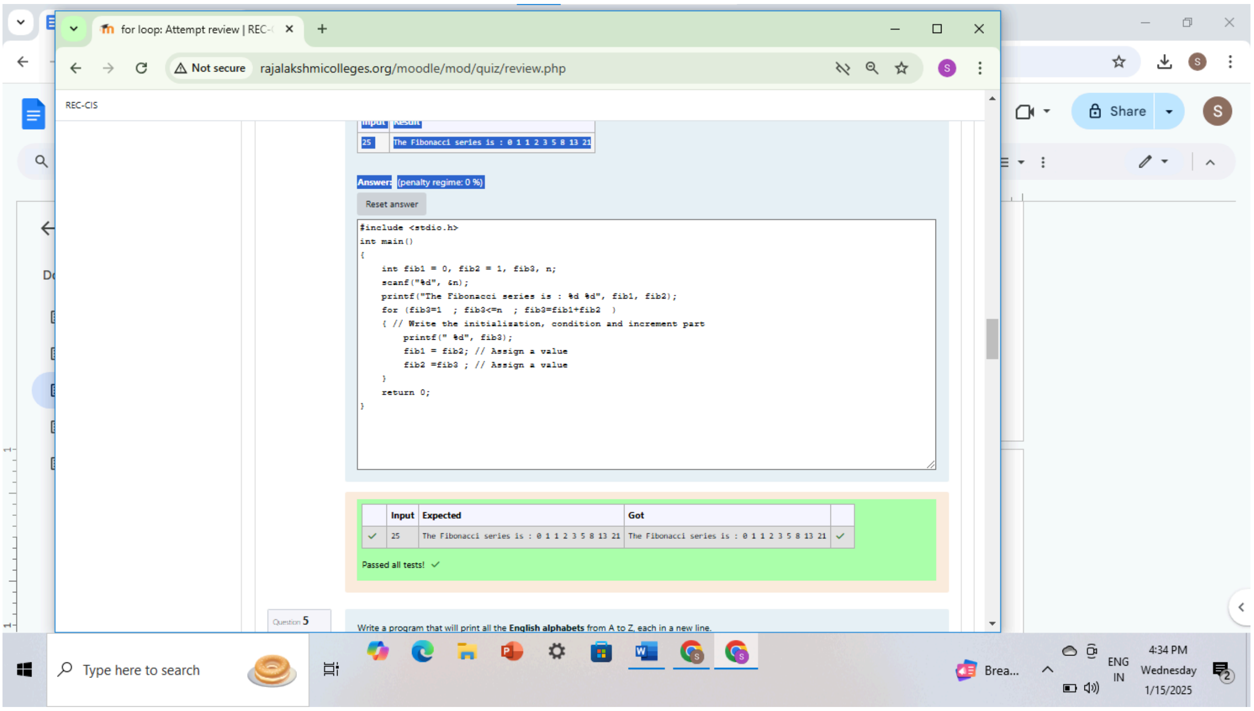
5.4.Fill in the missing code in the below program to print the Fibonacci series i.e., 0 1 1 2 3 5 8 13 21....., up to the limit.

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 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

Answer:(penalty regime: 0 %)



Input	Result
25	The Fibonacci series is : 0 1 1 2 3 5 8 13 21

Answer: (penalty regime: 0 %)

Reset answer

```
#include <stdio.h>
int main()
{
    int fib1 = 0, fib2 = 1, fib3, n;
    scanf("%d", &n);
    printf("The Fibonacci series is : %d %d", fib1, fib2);
    for (fib3=1  ; fib3<=n  ; fib3=fib1+fib2  )
    { // Write the initialization, condition and increment part
        printf(" %d", fib3);
        fib1 = fib2; // Assign a value
        fib2 =fib3 ; // Assign a value
    }
    return 0;
}
```

	Input	Expected	Got	
✓	25	The Fibonacci series is : 0 1 1 2 3 5 8 13 21	The Fibonacci series is : 0 1 1 2 3 5 8 13 21	✓

Passed all tests! ✓

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

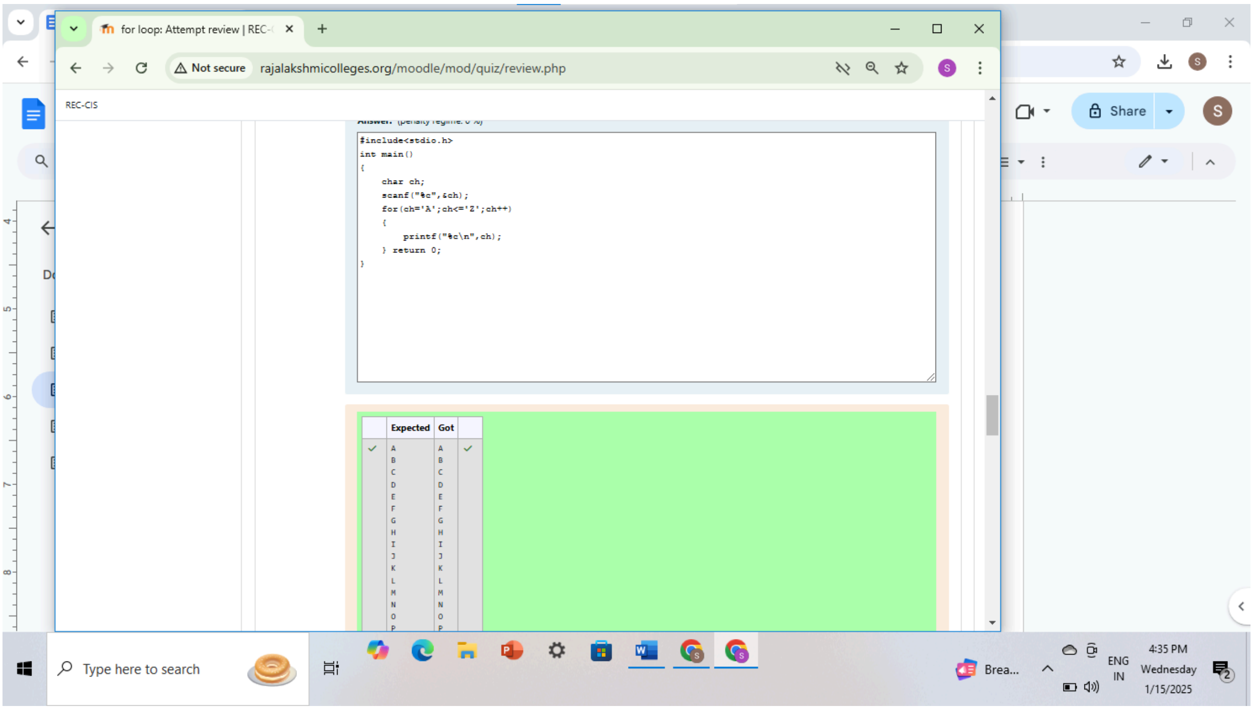
5.5.Write a program that will print all the English alphabets from A to Z, each in a new line.

Hints

1. The code in the main() function can use a for loop to iterate over the characters 'A' to 'Z'.
2. Note that char data type is a numeric type and can be used in a for loop as a loop counter.
3. 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++.
4. You can then print i directly which is of type char, using the printf() function with a newline character (\n).

For example:

Resu


for loop: Attempt review | REC-
Not secure
rajalakshmicolleges.org/moodle/mod/quiz/review.php
REC-CIS
Share
Type here to search
Brea...
ENG
IN
4:35 PM
Wednesday
1/15/2025

Answer: (penalty regime: 0 %)

```
#include<stdio.h>
int main()
{
    char ch;
    scanf("%c",&ch);
    for(ch='A';ch<='Z';ch++)
    {
        printf("%c\n",ch);
    } return 0;
}
```

	Expected	Got	
✓	A B C D E F G H I J K L M N O P	A B C D E F G H I J K L M N O P	✓

5.6.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.

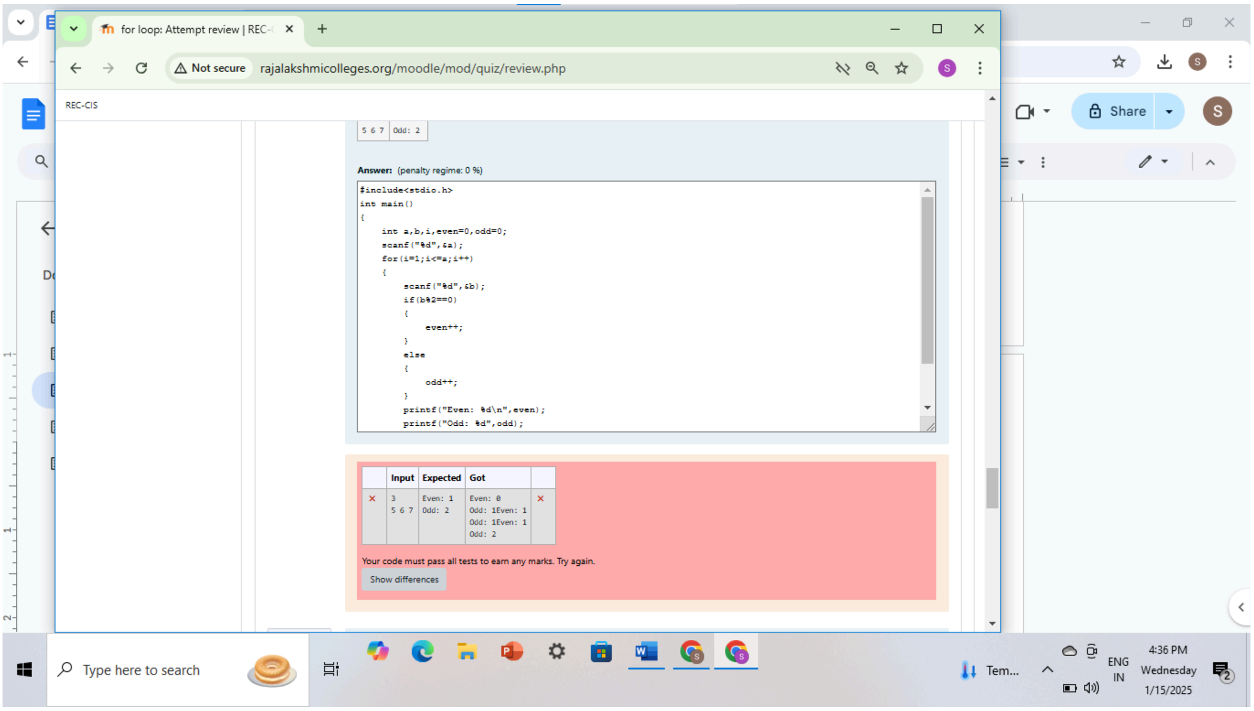
Instruction: To run your custom test cases strictly map your input and output layout with the visible test cases.

For example:

Input	Result

3 5 6 7	Even: 1 Odd: 2

Answ


for loop: Attempt review | REC-
Not secure
rajalakshmicolleges.org/moodle/mod/quiz/review.php
REC-CIS
Share
Type here to search
Tem...
ENG
IN
4:36 PM
Wednesday
1/15/2025

5 6 7	Odd: 2

Answer: (penalty regime: 0 %)

```
#include<stdio.h>
int main()
{
    int a,b,i,even=0,odd=0;
    scanf("%d",&a);
    for(i=1;i<=a;i++)
    {
        scanf("%d",&b);
        if(b%2==0)
        {
            even++;
        }
        else
        {
            odd++;
        }
        printf("Even: %d\n",even);
        printf("Odd: %d",odd);
```

	Input	Expected	Got	
✗	3 5 6 7	Even: 1 Odd: 2	Even: 0 Odd: 1Even: 1 Odd: 1Even: 1 Odd: 2	✗

Your code must pass all tests to earn any marks. Try again.

Show differences

5.7.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 perfect numbers as : 6 = 1 + 2 + 3 and 28 = 1 + 2 + 4 + 7 + 14.

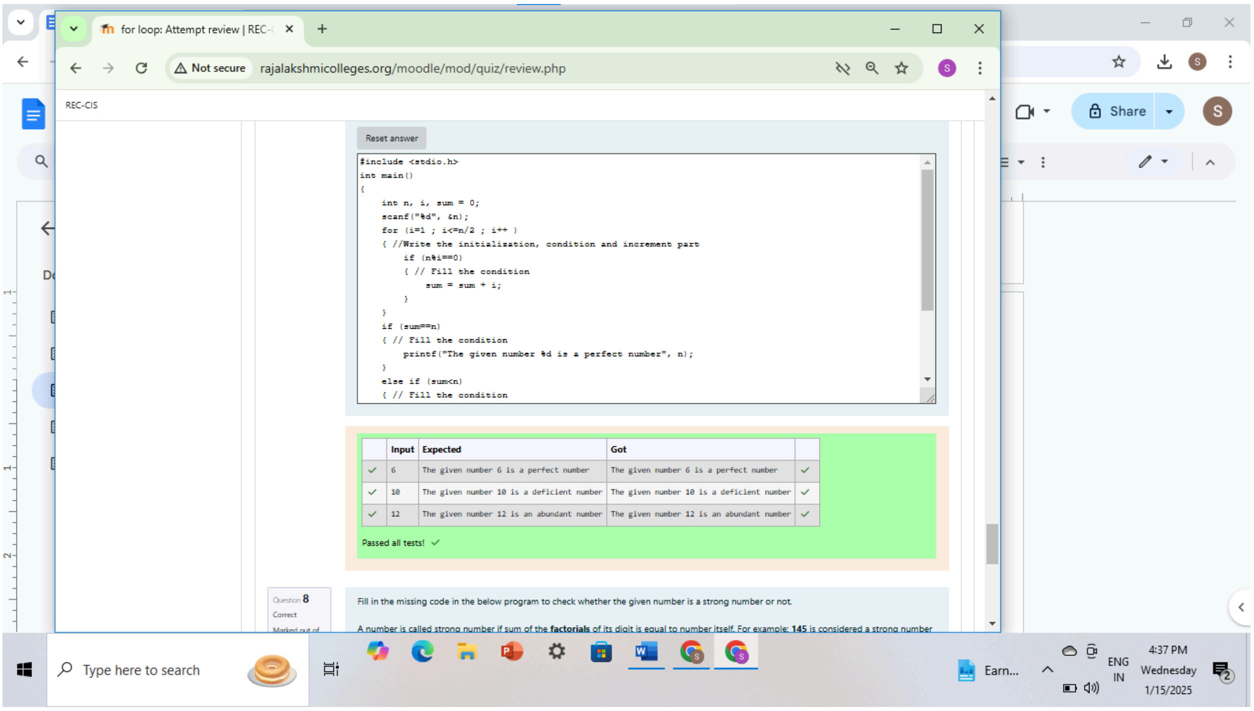
Alternatively, if the sum of a number's proper divisors exceeds the number itself, it is said to be abundant, while if the sum of a number's proper divisors is less-than the number itself, it is said to be deficient.

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

Answer:(penalty regime: 0 %)



```
#include <stdio.h>
int main()
{
    int n, i, sum = 0;
    scanf("%d", &n);
    for (i=1 ; i<=n/2 ; i++ )
    { //Write the initialization, condition and increment part
        if (n%i==0)
        { // Fill the condition
            sum = sum + i;
        }
    }
    if (sum==n)
    { // Fill the condition
        printf("The given number %d is a perfect number", n);
    }
    else if (sum<n)
    { // Fill the condition
```

	Input	Expected	Got	
✓	6	The given number 6 is a perfect number	The given number 6 is a perfect number	✓
✓	10	The given number 10 is a deficient number	The given number 10 is a deficient number	✓
✓	12	The given number 12 is an abundant number	The given number 12 is an abundant number	✓

Passed all tests! ✓

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 digit is equal to number itself. For example: **145** is considered a strong number

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

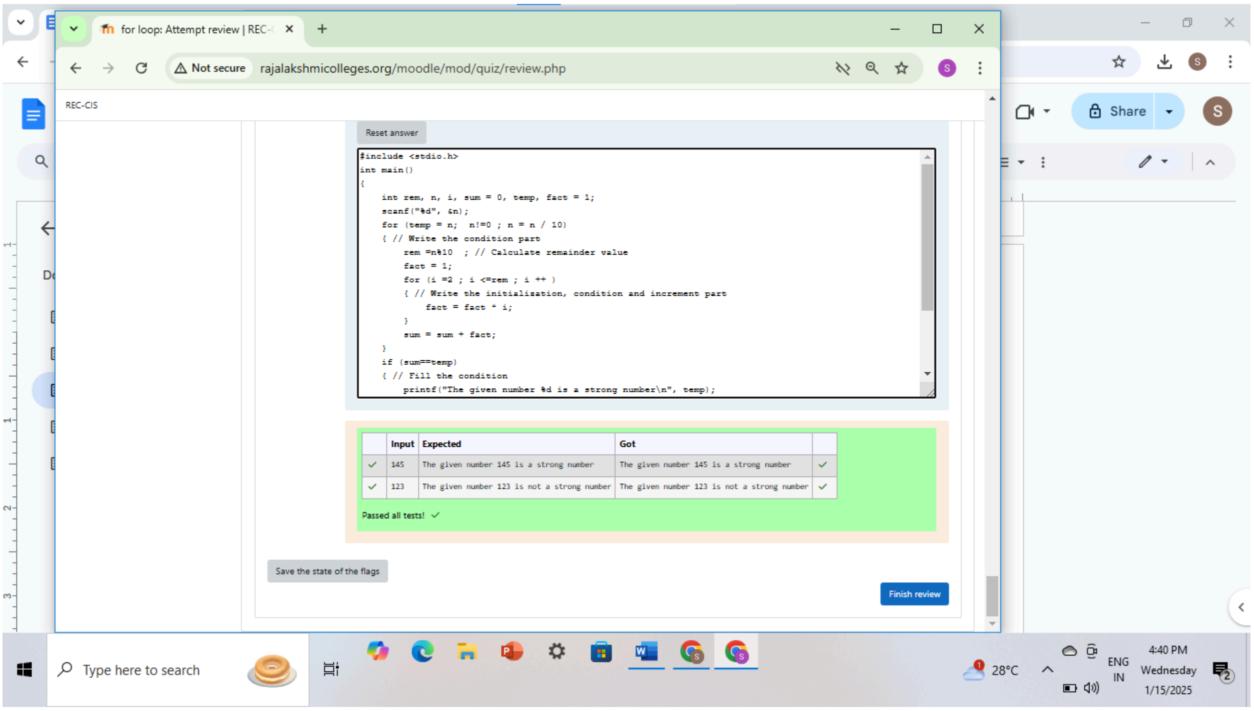
A number is called strong number if sum of the factorials of its digit is equal to number itself. For example: 145 is considered a strong number since 1! + 4! + 5! = 1 + 24 + 120 = 145.

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

Answer:(penalty regime: 0 %)



for loop: Attempt review | REC-
Not secure
rajalakshmicolleges.org/moodle/mod/quiz/review.php
REC-CIS
Reset answer

```
#include <stdio.h>
int main()
{
    int rem, n, i, sum = 0, temp, fact = 1;
    scanf("%d", &n);
    for (temp = n;  n!=0 ; n = n / 10)
    { // Write the condition part
        rem =n%10  ; // Calculate remainder value
        fact = 1;
        for (i =2 ; i <=rem ; i ++ )
        { // Write the initialization, condition and increment part
            fact = fact * i;
        }
        sum = sum + fact;
    }
    if (sum==temp)
    { // Fill the condition
        printf("The given number %d is a strong number\n", temp);
```

	Input	Expected	Got	
✓	145	The given number 145 is a strong number	The given number 145 is a strong number	✓
✓	123	The given number 123 is not a strong number	The given number 123 is not a strong number	✓

Passed all tests! ✓