

# Week 2:

The screenshot shows a Moodle quiz interface titled "Profit Calculator". The navigation bar on the left includes links for Dashboard, Site home, Site pages, My courses (GE23131-PUC-2024), Participants, Competencies, Grades, General, Skill Test-01-MCQ & Coding, Lecture Notes, Week-01-Overview of C, Constants, Variables and Da..., and Assessment-01-. A "Done" button is visible at the top right of the main content area.

Attempts allowed: 1  
Time limit: 1 hour 30 mins

Your attempts

Attempt 1	
Status	Finished
Started	Monday, 23 December 2024, 5:33 PM
Completed	Saturday, 26 October 2024, 10:22 PM
Duration	57 days 19 hours

Review

REC-CIS

The screenshot shows a Moodle question page for a profit calculation problem. The question details are as follows:

**Question 1**  
Correct  
Marked out of 1.00  
 Flag question

Each Sunday, a newspaper agency sells X copies of a certain newspaper for Rs.A per copy. The cost to the agency of each newspaper is Rs.B . 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 X, A and B.

**Input Format:**  
Input consists of 3 integers: X, A and B. X is the number of copies sold, A is the cost per copy and B 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

REC-CIS

REC-CIS

For example:

Input	Result
1000	900
2	
1	

Answer: (penalty regime: 0 %)

```
#include<stdio.h>
int main()
{
    int a,b,c,d;
    scanf("%d%d%d", &a, &b, &c);
    d=((a*b)-(a*c))-100;
    printf("%d", d);
    return 0;
}
```

84°F Mostly sunny

REC-CIS

	Input	Expected	Got
✓	1000	900	900
	2		
	1		

Passed all tests! ✓

Save the state of the flags

Finish review

84°F Mostly sunny

REC-CIS

Question 1  
Correct  
Marked out of 1.00  
 Flag question

Baba is very kind to beggars and every day Baba donates half of the amount he has when ever a beggar requests him. The money M left in Baba's hand is passed as the input and the number of beggars B 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 M.  
The second line denotes the value of B.

**Output Format:**

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

**Example Input/Output:**

Input:

REC-CIS

**Answer:** (penalty regime: 0 %)

```
#include<stdio.h>
int main()
{
    int a,b,i,amt;
    scanf("%d%d",&a,&b);
    amt=a;
    for(i=1;i<=b;i++)
    {
        amt=amt*2;
    }
    printf("%d",amt);
    return 0;
}
```

REC-CIS

}

	Input	Expected	Got	
✓	100 2	400	400	✓

Passed all tests! ✓

Save the state of the flags

Finish review

REC-CIS

Duration 34 days 18 hours

Question 1

Correct

Marked out of 1.00

Flag question

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 I for the starting day (ie on Monday) is passed as the input to the program. The number of days N an employee came on time consecutively starting from Monday is also passed as the input. The program must calculate and print the "Punctuality Incentive" P of the employee.

**Input Format:**

The first line denotes the value of I.  
The second line denotes the value of N.

**Output Format:**

The first line denotes the value of P.

REC-CIS

```
#include<stdio.h>
int main()
{
    int a,b,i,tot;
    scanf("%d%d",&a,&b);
    for(i=1;i<=b;i++)
    {
        tot+=a+200;
    }
    printf("%d",tot);
    return 0;
}
```

	Input	Expected	Got	
✓	500 3	2100	2100	✓
✓	100 3	900	900	✓

REC-CIS

Question 1  
Correct  
Marked out of 1.00  
 Flag question

Bajan Lal distributes C chocolates to school N students every Friday. The C chocolates are distributed among N students equally 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

84°F Mostly sunny      Search      ENG IN 15:36 15-01-2025

The screenshot shows a web browser window with a URL of [rajalakshmicolleges.org/moodle/mod/quiz/review.php](http://rajalakshmicolleges.org/moodle/mod/quiz/review.php). The page displays a C program in a code editor window. The code is as follows:

```
#include<stdio.h>
int main()
{
    int a,b,c;
    scanf("%d%d", &a, &b);
    c=a/b;
    printf("%d", a-(b*c));
    return 0;
}
```

The screenshot shows a web browser window with a URL of [rajalakshmicolleges.org/moodle/mod/quiz/review.php](http://rajalakshmicolleges.org/moodle/mod/quiz/review.php). The page displays a table showing test results and a message indicating all tests passed.

	Input	Expected	Got	
✓	300 45	30	30	✓

Passed all tests! ✓

Save the state of the flags

Finish review



REC-CIS

Question 1  
Correct  
Marked out of 1.00  
 Flag question

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;
```

```
#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 if the condition returns 1 (true) then the control enters in to 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

REC-CIS

Answer: (penalty regime: 0 %)

Reset answer

```
#include <stdio.h>

int main()
{
    int num;
    scanf("%d", &num);
    if(num%3==0)
    {
        printf("Given number %d is divisible by 3\n", num);
    }
    else
    {
        printf("Given number %d is not divisible by 3\n", num);
    }
    return 0;
}
```

Rain showers Sunday

REC-CIS

```
}
```

	Input	Expected	Got	
✓	9	Given number 9 is divisible by 3	Given number 9 is divisible by 3	✓
✓	7	Given number 7 is not divisible by 3	Given number 7 is not divisible by 3	✓

Passed all tests! ✓

Save the state of the flags

Finish review

Rain showers Sunday

REC-CIS

Show one page at a time

Finish review

Question 1  
Correct  
Marked out of 1.00  
 Flag question

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 a 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*)

```
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.

REC-CIS

Answer: (penalty regime: 0 %)

Reset answer

```
#include <stdio.h>

int main()
{
    int marks, distinction_marks = 75;
    scanf("%d", &marks);
    if(marks>distinction_marks)
        // Write the if condition
        printf("User secured distinction.\n");
    else
        // Write else part
        printf("User did not secure distinction.\n");
    return 0;
}
```

Finance headline India Passenger...

REC-CIS

	Input	Expected	Got	
✓	76	User secured distinction.	User secured distinction.	✓
✓	21	User did not secure distinction.	User did not secure distinction.	✓

Passed all tests! ✓

Question 2

Correct

Marked out of  
1.00

Flag question

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:



REC-CIS

**Answer:** (penalty regime: 0 %)

```
#include<stdio.h>
int main()
{
    float a;
    scanf("%f",&a);
    if(a<1000)
        printf("Balance is low");
    else
        printf("Sufficient balance");
    return 0;
}
```

Finance headline  
India Passenger...

REC-CIS

	Input	Expected	Got	
✓	1225	Sufficient balance	Sufficient balance	✓
✓	999.55	Balance is low	Balance is low	✓

Passed all tests! ✓

Upcoming Earnings

Question 3  
Correct  
Marked out of 1.00  
 Flag question

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%

The image shows a Windows desktop with two Google Chrome windows open. Both windows have the URL [rajalakshmicolleges.org/moodle/mod/quiz/review.php](http://rajalakshmicolleges.org/moodle/mod/quiz/review.php).

**Top Window Content:**

```
#include <stdio.h>

int main()
{
    int maths_marks, computers_marks, physics_marks, chemistry_marks,
        english_marks, spanish_marks, total;
    float percentage;

    scanf("%d%d%d%d", &maths_marks, &computers_marks, &physics_marks, &chemistry_marks,
          &english_marks, &spanish_marks);
    // Read marks

    total=maths_marks+computers_marks+physics_marks+chemistry_marks+english_marks+
        spanish_marks;
    percentage=(total/600.0)*100;
    // Calculate total and percentage

    if(percentage>=60)
    {
        // Write the condition
        printf("Student secured a first class with %5.2f%%\n", percentage);
    }
    else
    {
        // Write the else part
        printf("Student did not secure a first class with %5.2f%%\n",
               percentage);
    }
    return 0;
}
```

**Bottom Window Content:**

```
Reset answer
// Read marks

total=maths_marks+computers_marks+physics_marks+chemistry_marks+english_marks+
spanish_marks;
percentage=(total/600.0)*100;
// Calculate total and percentage

if(percentage>=60)
{
    // Write the condition
    printf("Student secured a first class with %5.2f%%\n", percentage);
}
else
{
    // Write the else part
    printf("Student did not secure a first class with %5.2f%%\n",
           percentage);
}
return 0;
```

REC-CIS

	Input	Expected	Got
✓	45 67 34 57 68 81	Student did not secure a first class with 58.67%	Student did not secur
✓	67 68 65 56 59 69	Student secured a first class with 64.00%	Student secured a fir

Passed all tests! ✓

Question 4

Correct

Marked out of  
1.00

Flag question

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 :

REC-CIS

**Answer:** (penalty regime: 0 %)

```
#include <stdio.h>

int main()
{
    int number;
    scanf("%d", &number);

    if(number%2==0)
    {
        printf("The given number %d is an even number", number);
    }
    else
    {
        printf("The given number %d is an odd number", number);
    }
    return 0;
}
```

REC-CIS

	Input	Expected	Got
✓	35	The given number 35 is an odd number	The given number 35 is an odd number ✓
✓	10	The given number 10 is an even number	The given number 10 is an even number ✓

Passed all tests! ✓

Question 5  
Correct  
Marked out of 1.00  
 Flag question

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

REC-CIS

Input	Result
W	Given character W is an alphabet
7	Given character 7 is not an alphabet

**Answer:** (penalty regime: 0 %)

```
#include<stdio.h>
int main()
{
    char a;
    scanf("%c",&a);
    if((a>=65 && a<=90) || (a>=97 && a<=122))
        printf("Given character %c is an alphabet",a);
    else
        printf("Given character %c is not an alphabet",a);
    return 0;
}
```

REC-CIS

	Input	Expected	Got	
✓	W	Given character W is an alphabet	Given character W is an alphabet	✓
✓	7	Given character 7 is not an alphabet	Given character 7 is not an alphabet	✓

Passed all tests! ✓

Save the state of the flags Finish review

REC-CIS

Question 1  
Correct  
Marked out of 1.00  Flag question

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;
    }
}
```

REC-CIS

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

REC-CIS

Finish review

Question 1  
Correct  
Marked out of 1.00  
 Flag question

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;
}
```

REC-CIS

**Answer:** (penalty regime: 0 %)

[Reset answer](#)

```
#include <stdio.h>

int main()
{
    char ch;
    ch = getchar();
    if((ch>='A' && ch<='Z') || (ch>='a' && ch<='z'))//fill the appropriate if condition
        printf("Given character %c is an alphabet\n", ch);
    else if(ch>='0' && ch<='9')//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);
    return 0;
}
```

REC-CIS

	Input	Expected	Got
✓	A	Given character A is an alphabet	Given character A is an alpha
✓	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

Passed all tests! ✓

**Question 2**  
Correct  
Marked out of 1.00  
 Flag question

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”.



REC-CIS

Reset answer

```
#include <stdio.h>

int main()
{
    int year;
    scanf("%d", &year);
    // Fill in the missing code
    if(year%4==0 && year%100!=0)
        printf("%d is a leap year\n", year);

    else if(year%400==0)
        printf("%d is a leap year\n", year);

    else
        printf("%d is not a leap year\n", year);

    return 0;
}
```

REC-CIS

	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  
 Flag question

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**.

```
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");
    }
    return 0;
}
```

	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! ✓

#### Question 4

Correct

Marked out of  
1.00

Flag  
question

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**.



REC-CIS

Answer: (penalty regime: 0 %)

Reset answer

```
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");
    }
    return 0;
}
```

85°F Mostly cloudy 15-01-2025

REC-CIS

```
}
```

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

Passed all tests! ✓

Save the state of the flags Finish review

85°F Mostly cloudy 15-01-2025

REC-CIS

Question 1  
Correct  
Marked out of 1.00  
 Flag question

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);
    }
}
```

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"); 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);
    }
}
```

REC-CIS

	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! ✓

Question 2

Correct

Marked out of  
1.00

Flag question

Assume that the weekdays are provided with the below numbers:

Sunday → 0  
Monday → 1  
Tuesday → 2  
Wednesday → 3  
Thursday → 4  
Friday → 5  
Saturday → 6

REC-CIS

Marked out of  
1.00

Flag question

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

REC-CIS

Answer: (penalty regime: 0 %)

```
#include<stdio.h>
int main()
{
    int n;
    scanf("%d",&n);
    switch(n)
    {
        case 0 : printf("Sunday"); break;
        case 1 : printf("Monday"); break;
        case 2 : printf("Tuesday"); break;
        case 3 : printf("Wednesday"); break;
        case 4 : printf("Thursday"); break;
        case 5 : printf("Friday"); break;
        case 6 : printf("Saturday"); break;
        default : printf("Invalid weekday number"); break;
    }
}
```

REC-CIS

```
default : printf("Invalid weekday number"); break;
}
```

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

Passed all tests! ✓

Save the state of the flags

Finish review



The screenshot shows a Google Chrome window with four tabs open. The active tab is titled "while loop: Attempt review | RE" and displays a Moodle quiz review page. The page header includes "REC-CIS" and "Question 1". The question asks about the general syntax for a while statement, which is shown as:

```
while (condition)
{
    statement_1;
    statement_2;
    ...
}
```

The text explains that the block of code inside the braces is called the "while-loop body". It also states that a while statement executes code repeatedly as long as the condition is true. The condition is described as an expression that should always evaluate to true or false, with two bullet points: "If it evaluates to true, the body containing one or more code statements is executed." and "If the expression evaluates to false, the control skips executing the while-loop body."

The screenshot shows a Google Chrome window with four tabs open. The active tab is titled "while loop: Attempt review | RE" and displays a Moodle quiz review page for a question about a C program. The question asks for the answer, which is provided in a box:

**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);
    return 0;
}
```

The status bar at the bottom of the screen shows the date and time as 15-01-2025 and 15:51.

REC-CIS

while loop: Attempt review | RE

Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php

Google Chrome isn't your default browser Set as default

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

Passed all tests! ✓

Question 2  
Correct  
Marked out of 1.00  
 Flag question

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

REC-CIS

while loop: Attempt review | RE

Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php

Google Chrome isn't your default browser Set as default

85°F Mostly cloudy

Search

ENG IN 15:52 15-01-2025

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;
}
```

85°F Mostly cloudy

Search

ENG IN 15:52 15-01-2025

REC-CIS

while loop: Attempt review | RE

Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php

Google Chrome isn't your default browser Set as default

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

Passed all tests! ✓

Question 3  
Correct  
Marked out of 1.00  
 Flag question

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

REC-CIS

while loop: Attempt review | RE

Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php

Google Chrome isn't your default browser Set as default

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

**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++;
    }
    return 0;
}
```

REC-CIS

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

Passed all tests! ✓

Question 4

Correct

Marked out of  
1.00

Flag  
question

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:

REC-CIS

```
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 be

REC-CIS

Question 5  
Correct  
Marked out of 1.00  
 Flag question

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

REC-CIS

```
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 =

REC-CIS

Question 6  
Correct  
Marked out of 1.00  
 Flag question

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

REC-CIS

```
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	✓

REC-CIS

Passed all tests! ✓

Question 7  
Correct  
Marked out of 1.00  
 Flag question

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 %)

Reset answer

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

REC-CIS

```
#include <stdio.h>

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

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

REC-CIS

```

    }
    if (count == 0 )
    { // complete the condition to check the count
        printf("The given number %d is a prime number\n", n);
    }
    else
    {
        printf("The given number %d is not a prime number\n", n);
    }
    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! ✓

85°F Mostly cloudy

REC-CIS

**Question 8**

Correct  
Marked out of 1.00  Flag question

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,  
 $9 = 9^1 = 9$   
 $371 = 3^3 + 7^3 + 1^3 = 27 + 343 + 1 = 371$   
 $38208 = 3^4 + 8^4 + 2^4 + 0^4 + 8^4 = 4096 + 16 + 0 + 4096 = 8208$

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

The screenshot shows a Google Chrome window with the URL [rajalakshmicolleges.org/moodle/mod/quiz/review.php](http://rajalakshmicolleges.org/moodle/mod/quiz/review.php). The page title is "while loop: Attempt review | RE". A "Set as default" button is visible at the top right. The main content area contains a code editor with the following C code:

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

int main()
{
    int num,i,og,r,n=0,sum=0;
    scanf("%d", &num);
    og=num;
    i=og;
    while(i!=0)
    {
        n++;
        i=i/10;
    }

    i=og;
    while(i!=0)
    {

```

The screenshot shows the continuation of the C program from the previous screen. The code editor now includes the following part:

```
    i=og;
    while(i!=0)
    {
        r=i%10;
        sum+= pow(r,n);
        i=i/10;
    }
    if(sum==num)
        printf("The given number %d is an armstrong number",num);
    else
        printf("The given number %d is not an armstrong number",num);

    return 0;
}
```

Below the code editor is a table titled "Test Cases" with a green border, showing two rows of input, expected output, and got output:

	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

The taskbar at the bottom of the screen displays the Windows Start button, a search bar, and various pinned application icons. On the right side, it shows the date (15-01-2025), time (15:55), battery level (ENG IN), signal strength, and a notification icon.

REC-CIS

Show one page at a time

Finish review

Question 1  
Correct  
Marked out of 1.00  
 Flag question

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++)
{
```

REC-CIS

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;
}
```

REC-CIS

Passed on test

Question 2  
Correct  
Marked out of 1.00  
 Flag question

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

REC-CIS

85°F Mostly cloudy

Search

for loop: Attempt review | REC-CIS

Reset answer

**Answer:** (penalty regime: 0 %)

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

85°F Mostly cloudy

Search

for loop: Attempt review | REC-CIS

The screenshot shows a web browser window with a green header bar indicating a successful test run. The URL is [rajalakshmicolleges.org/moodle/mod/quiz/review.php](http://rajalakshmicolleges.org/moodle/mod/quiz/review.php). A table at the top shows input values 2 3 and expected output 8, with the got value also being 8, resulting in a green checkmark. Below the table, a green box displays the message "Passed all tests! ✓".

Question 3  
Correct  
Marked out of 1.00  
 Flag question

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

**Constraints:**

- $1 \leq n \leq 10^6$
- $10^{-3} \leq \text{elements} \leq 10^3$
- 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.

The screenshot shows a web browser window with a pink header bar. The URL is [rajalakshmicolleges.org/moodle/mod/quiz/review.php](http://rajalakshmicolleges.org/moodle/mod/quiz/review.php). A table at the top shows input values 4 and 3 5 7 8, with the result column showing Sum: 23 and Mean: 5.75 respectively. Below the table, a pink box displays the answer code:

```
#include<stdio.h>
int main()
{
    int n,a,i,sum=0;
    float mean;
    scanf("%d",&n);
    for(i=1;i<=n;i++)
    {
        scanf("%d",&a);
        sum=sum+a;
    }
    mean= (sum*1.0)/n;
    printf("Sum: %d\n",sum);
    printf("Mean: %.2f",mean);
    return 0;
}
```

REC-CIS

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

Passed all tests! ✓

**Question 4**

Correct

Marked out of  
1.00 Flag  
question

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

REC-CIS

```
#include <stdio.h>
int main()
{
    int fib1 = 0, fib2 = 1, fib3,n;
    scanf("%d", &n);
    printf("The Fibonacci series is : %d %d ", fib1, fib2);
    fib3=fib1+fib2;
    printf("%d ",fib3);
    for (int i=3 ; i<n ; i++)
    { // Write the initialization, condition and increment part

        fib1 = fib2 ; // Assign a value
        fib2 = fib3 ;
        fib3 = fib1+fib2;
        // Assign a value
        if(fib3>n)
            break;
        printf("%d ",fib3);
    }
}
```

REC-CIS

Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php Set Chrome as your default

Google Chrome isn't your default browser Set as default

	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

Passed all tests! ✓

Question 5  
Correct  
Marked out of 1.00  
 Flag question

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).

REC-CIS

85°F Mostly cloudy Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php Set Chrome as your default

Google Chrome isn't your default browser Set as default

**Answer:** (penalty regime: 0 %)

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

85°F Mostly cloudy Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php Set Chrome as your default

Google Chrome isn't your default browser Set as default

REC-CIS

Question 6  
Correct  
Marked out of 1.00  
 Flag question

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

**Constraints:**

- $1 \leq n \leq 10^6$
- $10^{-3} \leq \text{elements} \leq 10^3$

**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	Even: 1

REC-CIS

**Answer:** (penalty regime: 0 %)

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

REC-CIS

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

Passed all tests! ✓

Question 7

Correct

Marked out of  
1.00

Flag question

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

REC-CIS

```
#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
        printf("The given number %d is a deficient number", n);
    }
}
```

REC-CIS

Input	Expected	Got

REC-CIS

```

    }
else
{
    printf("The given number %d is an abundant number", n);
}
return 0;
}

```

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! ✓

Question 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 %)

Reset answer

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

REC-CIS

**Answer:** (penalty regime: 0 %)

[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=1 ; 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);
    }
}
```

85°F Mostly cloudy    Search    ENG IN 15-01-2025

REC-CIS

```
if (sum==temp )
{ // Fill the condition
    printf("The given number %d is a strong number\n", temp);
}
else
{
    printf("The given number %d is not a strong number\n", temp);
}
return 0;
}
```

	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 r

Passed all tests! ✓

85°F Mostly cloudy    Search    ENG IN 16:00 15-01-2025