



Not secure

rajalakshmicolleges.org/moodle/mod/quiz/view.php?id=84



RESHMASHRI S 2024-IT

R2

REC-CIS

GE23131-Programming Using C-2024

Dashboard / My courses / GE23131-PUC-2024 / Assessment-03-Decision Making and Branching - if, ... / Admission Eligibility

Navigation

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 Data...](#)[Assessment-01-Overview of C, Constants, Variables ...](#)[Week-02-Operators and Expressions, Managing](#)

Admission Eligibility

[✓ Done](#)

Attempts allowed: 1

Time limit: 30 mins

Your attempts

Attempt 1

Status	Finished
--------	----------

Started	Monday, 23 December 2024, 5:33 PM
---------	-----------------------------------

Completed	Saturday, 30 November 2024, 6:58 PM
-----------	-------------------------------------

Duration	22 days 22 hours
----------	------------------

[Review](#)

Admission Eligibility: Attempt review | REC-CIS - Google Chrome

Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=104541&cmid=84

REC-CIS

Quiz navigation

1

Finish review

Status	Finished
Started	Monday, 23 December 2024, 5:33 PM
Completed	Saturday, 30 November 2024, 6:58 PM
Duration	22 days 22 hours

Question 1

Correct

Marked out of 1.00

Flag question

Write a C program to find the eligibility of admission for a professional course based on the following criteria:

Marks in Maths \geq 65

Marks in Physics \geq 55

Marks in Chemistry \geq 50

Or

Total in all three subjects \geq 180

Sample Test Cases

Test Case 1

Input

70 60 80

Output

The candidate is eligible

Test Case 2

Input

50 80 80

Output

The candidate is eligible

Admission Eligibility: Attempt review | REC-CIS - Google Chrome

Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=104541&cmid=84

REC-CIS

Test Case 2

Input

50 80 80

Output

The candidate is eligible

Test Case 3

Input

50 60 40

Output

The candidate is not eligible

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     int a,b,c,d;
5     scanf("%d%d%d",&a,&b,&c);
6     d=a+b+c;
7     if(d>=180)
8         printf("The candidate is eligible");
9     else
10        printf("The candidate is not eligible");
11        return 0;
12 }
13 }
```

Admission Eligibility: Attempt review | REC-CIS - Google Chrome

Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=104541&cmid=84

REC-CIS

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     int a,b,c,d;
5     scanf("%d%d%d",&a,&b,&c);
6     d=a+b+c;
7     if(d>=180)
8         printf("The candidate is eligible");
9     else
10        printf("The candidate is not eligible");
11    return 0;
12 }
```

	Input	Expected	Got	
✓	70 60 80	The candidate is eligible	The candidate is eligible	✓
✓	50 80 80	The candidate is eligible	The candidate is eligible	✓

Passed all tests! ✓

9 28°C Mostly cloudy

Search

myhp

11:30 15-01-2025

REC-CIS

RESHMASHRI S 2024-IT R2

Dashboard / My courses / GE23131-PUC-2024 / Assessment-03-Decision Making and Branching - it, ... / Calculator

Navigation

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...

Assessment-01-Overview of C, Constants, Variables ...

Week-02-Operators and Expressions, Managing Input ...

Assessment-02-Operators and Expressions, Managing

...

Calculator

✓ Done

Re-attempt quiz

Attempts allowed: 2

Time limit: 30 mins

Grading method: Highest grade

Your attempts

Attempt 1

Status	Finished
Started	Monday, 13 January 2025, 8:34 PM
Completed	Monday, 13 January 2025, 9:02 PM
Duration	27 mins 36 secs

Review



Search



ENG
IN



11:30
15-01-2025

Calculator: Attempt review | REC-CIS - Google Chrome

Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=158393&cmid=85

REC-CIS

Quiz navigation

1

Finish review

Status	Finished
Started	Monday, 13 January 2025, 8:34 PM
Completed	Monday, 13 January 2025, 9:02 PM
Duration	27 mins 36 secs

Question 1
Incorrect
Marked out of 1.00
[Flag question](#)

Complete the calculator program with Basic operations (+, -, *, /, %) of two numbers using switch statement.

Sample Test Cases

Test Case 1

Input

45
45
+

Output

Result: $45 + 45 = 90.000000$

Test Case 2

Calculator: Attempt review | REC-CIS - Google Chrome

Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=158393&cmid=85

REC-CIS

	<p>Input</p> <p>56 8 %</p> <p>Output</p> <p>Result: $56 \% 8 = 0.000000$</p> <p>Test Case 3</p> <p>Input</p> <p>50 70 \$</p> <p>Output</p> <p>Invalid operation.</p> <p>Result: $50 \\$ 70 = 0.000000$</p>

Calculator: Attempt review | REC-CIS - Google Chrome

Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=158393&cmid=85

REC-CIS

```
1 #include<stdio.h>
2 int main()
3 {
4     double num1,num2;
5     char op;
6     scanf("%lf %lf %c",&num1,&num2,&op);
7     switch(op)
8     {
9         case '+':
10            printf("%.6lf\n",num1+num2);
11            break;
12         case '-':
13            printf("%.6lf\n",num1-num2);
14            break;
15         case '*':
16            printf("%.6lf\n",num1*num2);
17            break;
18         case '/':
19             if(num2!=0)
20                 printf("%.6lf\n",num1/num2);
21             else
22
23                 printf("undefined\n");
24
25
26
27         break;
28         case '%':
29             if((int)num2!=0)
30                 printf("%d\n",(int)num1 % (int)num2);
31             else
32                 printf("undefined\n");
33             break;
34         default:
35             printf("Invalid operator\n");
36     }
37     return 0;
38 }
```

9 28°C Mostly cloudy

Search

myhp

11:31 ENG IN 15-01-2025

Calculator: Attempt review | REC-CIS - Google Chrome

Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=158393&cmid=85

REC-CIS

```
27     break;
28     case '%':
29     if((int)num2!=0)
30         printf("%d\n", (int)num1 % (int)num2);
31     else
32         printf("undefined\n");
33     break;
34     default:
35         printf("Invalid operator\n");
36     }
37     return 0;
38 }
```

	Input	Expected	Got	
✗	45 45 +	Result: 45 + 45 = 90.000000	90.000000	✗
✗	56 8 %	Result: 56 % 8 = 0.000000	0	✗

Some hidden test cases failed, too.
Your code must pass all tests to earn any marks. Try again.

Show differences

Finish review



REC-CIS

RESHMASHRI S 2024-IT R2

Dashboard / My courses / GE23131-PUC-2024 / Assessment-03-Decision Making and Branching - if, ... / Finding the Second Largest Element

Navigation

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 Data...

Assessment-01-Overview of C, Constants, Variables ...

Week-02-Operators and Expressions, Managing Input ...

Assessment-02-Operators and Expressions, Managing

Finding the Second Largest Element

✓ Done

Attempts allowed: 2

Time limit: 30 mins

Grading method: Highest grade

Your attempts

Attempt 2

Status	Finished
Started	Monday, 23 December 2024, 5:33 PM
Completed	Saturday, 30 November 2024, 7:45 PM
Duration	22 days 21 hours

Review

Attempt 1

Status	Finished
Started	Monday, 23 December 2024, 5:33 PM
Completed	Saturday, 30 November 2024, 7:32 PM
Duration	22 days 22 hours

Review

No more attempts are allowed

Finding the Second Largest Element: Attempt review | REC-CIS - Google Chrome

Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=104577&cmid=86

REC-CIS

1

Finish review

Status Finished
Started Monday, 23 December 2024, 5:33 PM
Completed Saturday, 30 November 2024, 7:45 PM
Duration 22 days 21 hours

Question 1
Correct
Marked out of 1.00
Flag question

You are given a sequence of integers as input, terminated by a -1. (That is, the input integers may be positive, negative or 0. A -1 in the input signals the end of the input.)
-1 is not considered as part of the input.
Find the second largest number in the input. You may not use arrays.

Sample Test Cases

Test Case 1
Input
-840 -288 -261 -337 -335 488 -1
Output
-261

Test Case 2
Input
-840 -335 -1
Output
-840

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 #include<limits.h>
3 int main()
4 {
```

Finding the Second Largest Element: Attempt review | REC-CIS - Google Chrome

Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=104577&cmid=86

REC-CIS

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 #include<limits.h>
3 int main()
4 {
5     int num,largest = INT_MIN,second_largest = INT_MIN;
6     while(1)
7     {
8         scanf("%d",&num);
9         if(num == -1)
10            break;
11         if(num > largest)
12         {
13             second_largest = largest;
14             largest = num;
15         }
16         else if(num > second_largest && num < largest)
17         {
18             second_largest = num;
19         }
20     }
21     if(second_largest == INT_MIN)
22     {
23         printf("No second largest number found.\n");
24     }
25     else
26     {
27         printf("%d\n",second_largest);
28     }
29 }
30 }
```

Input **Expected** **Got**

29°C Partly sunny

Search

myhp

11:32 15-01-2025

Finding the Second Largest Element: Attempt review | REC-CIS - Google Chrome

Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=104577&cmid=86

REC-CIS

```
13     second_largest = largest;
14     largest = num;
15 }
16     else if(num > second_largest && num < largest)
17 {
18     second_largest = num;
19 }
20     if(second_largest == INT_MIN)
21 {
22     printf("No second largest number found.\n");
23 }
24     else
25 {
26     printf("%d\n",second_largest);
27 }
28     return 0;
29 }
30 }
```

	Input	Expected	Got	
✓	-840 -288 -261 -337 -335 488 -1	-261	-261	✓
✓	-840 -335 -1	-840	-840	✓

Passed all tests! ✓

Finish review



RESHMASHRI S 2024-IT

R2

REC-CIS

Navigation

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 Data...](#)[Assessment-01-Overview of C, Constants, Variables ...](#)[Week-02-Operators and Expressions, Managing Input ...](#)[Assessment-02-Operators and Expressions, Managing](#)

...

[Week-03-Decision](#)

Triangle - Smallest Side

[Done](#)

Attempts allowed: 1

Time limit: 30 mins

Your attempts

Attempt 1

Status	Finished
Started	Monday, 23 December 2024, 5:33 PM
Completed	Sunday, 8 December 2024, 11:43 AM
Duration	15 days 5 hours

[Review](#)

No more attempts are allowed

[Back to the course](#)

Triangle - Smallest Side: Attempt review | REC-CIS - Google Chrome

Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=112366&cmid=87

REC-CIS

Quiz navigation

1

Finish review

Status	Finished
Started	Monday, 23 December 2024, 5:33 PM
Completed	Sunday, 8 December 2024, 11:43 AM
Duration	15 days 5 hours

Question 1

Correct

Marked out of 1.00

Flag question

The lengths of the sides of a triangle X, Y and Z are passed as the input. The program must print the smallest side as the output.

Input Format:

The first line denotes the value of X.
The second line denotes the value of Y.
The third line denotes the value of Z.

Output Format:

The first line contains the length of the smallest side.

Boundary Conditions:

$1 \leq X \leq 999999$
 $1 \leq Y \leq 999999$
 $1 \leq Z \leq 999999$

Example Input/Output 1:

Input:
40
30
50

Output:
30

Triangle - Smallest Side: Attempt review | REC-CIS - Google Chrome

Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=112366&cmid=87

REC-CIS

```
30
50

Output:
30
```

Example Input/Output 2:

```
Input:
15
15
15

Output:
15
```

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     int a,b,c;
5     scanf("%d%d%d",&a,&b,&c);
6     if((a<b)&&(a<c))
7     {
8         printf("%d",a);
9     }
10    else if(b<c)
11    {
12        printf("%d",b);
13    }
14    else
15    {
16        printf("%d",c);
17    }
18 }
19 return 0;
```

8 Breaking news MHA grants app... Search Windows Start File Explorer Edge myhp Microsoft Store Help & Support Microsoft Edge OneDrive Google Chrome

11:33 ENG IN 15-01-2025

Triangle - Smallest Side: Attempt review | REC-CIS - Google Chrome

Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=112366&cmid=87

REC-CIS

```
1 #include<stdio.h>
2 int main()
3 {
4     int a,b,c;
5     scanf("%d%d%d",&a,&b,&c);
6     if((a&&(a<c))
7     {
8         printf("%d",a);
9     }
10    else if(b<c)
11    {
12        printf("%d",b);
13    }
14    else
15    {
16        printf("%d",c);
17    }
18    return 0;
19 }
```

	Input	Expected	Got	
✓	40 30 50		30	✓
✓	15 15 15		15	✓

Passed all tests! ✓



RESHMASHRI S 2024-IT

R2

REC-CIS

Dashboard / My courses / GE23131-PUC 2024 / Assessment 03: Decision Making and Branching / Formal and Actual Arguments

Navigation

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 Data...

Assessment-01-Overview of C, Constants, Variables ...

Week-02-Operators and Expressions, Managing Input ...

Assessment-02-Operators and Expressions, Managing

...

Formal and Actual Arguments

Done

Re-attempt quiz

Time limit: 1 hour

Grading method: Highest grade

Your attempts

Attempt 1

Status	Finished
Started	Monday, 23 December 2024, 5:33 PM
Completed	Sunday, 8 December 2024, 11:57 AM
Duration	15 days 5 hours

Review

Formal and Actual Arguments: Attempt review | REC-CIS - Google Chrome

Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=112371&cmid=88

REC-CIS

GE23131-Programming Using C-2024

Quiz navigation

1

Finish review

Status	Finished
Started	Monday, 23 December 2024, 5:33 PM
Completed	Sunday, 8 December 2024, 11:57 AM
Duration	15 days 5 hours

Question 1
Correct
Marked out of 1.00
[Flag question](#)

An argument is an expression which is passed to a function by its caller in order for the function to perform its task. It is an expression in the comma-separated list bound by the parentheses in a function call expression.

A function may be called by the portion of the program with some arguments and these arguments are known as actual arguments (or) original arguments.

Actual arguments are local to the particular function. These variables are placed in the **function declaration** and **function call**. These arguments are defined in the **calling function**.

The parameters are variables defined in the function to receive the arguments.

Formal parameters are those parameters which are present in the **function definition**.

Formal parameters are available only with in the specified function. Formal parameters belong to the **called function**.

Formal parameters are also the local variables to the function. So, the formal parameters are occupied memory when the function execution starts and they are destroyed when the function execution completed.

Let us consider the below example:

Formal and Actual Arguments: Attempt review | REC-CIS - Google Chrome

Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=112371&cmid=88

REC-CIS

```
#include <stdio.h>
int add(int, int);
int main()
{
    int a = 10, b = 20;
    printf("Sum of two numbers = %d\n", add(a, b)); // variables a, b are called actual arguments
    return 0;
}

int add(int x, int y)
{
    // variables x, y are called formal parameters
    return(x + y);
}
```

In the above code whenever the function call add(a, b) is made, the execution control is transferred to the function definition of add().

The values of actual arguments a and b are copied in to the formal arguments x and y respectively.

The formal parameters x and y are available only with in the function definition of add(). After completion of execution of add(), the control is transferred back to the main().

See & retype the below code which will demonstrate about formal and actual arguments.

```
#include <stdio.h>

int sum(int);

int main()
{
    int number;
    scanf("%d", &number);
    printf("Sum of %d natural numbers = %d\n", number, sum(number));
    return 0;
}
```



9+ NIFTY +0.26% 12:17 ENG IN 15-01-2025

Formal and Actual Arguments: Attempt review | REC-CIS - Google Chrome

Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=112371&cmid=88

REC-CIS

```
    printf("Sum of %d natural numbers = %d\n", number, sum(number));
    return 0;
}

int sum(int value)
{
    int i, total = 0;
    for (i = 1; i <= value; i++)
    {
        total = total + i;
    }
    return(total);
}
```

For example:

Input	Result
5	Sum of 5 natural numbers = 15

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int sum(int);
3 int main()
4 {
5     int n;
6     scanf("%d",&n);
7     printf("Sum of %d natural numbers = %d\n",n,sum(n));
8     return 0;
9 }
10 int sum(int value)
11 {
12     int i,t = 0;
13     for(i=1;i<=value;i++)
```

Local and Global Variables: Attempt review | REC-CIS - Google Chrome

Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=112384&cmid=89

REC-CIS

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int a=20;
3 void test();
4 int main()
5 {
6     printf("In main() function a = %d\n",a);
7     test();
8     a=a+15;
9     printf("In main() function a = %d\n",a);
10    return 0;
11 }
12 void test()
13 {
14     a=a+20;
15     printf("In test() function a = %d\n",a);
16 }
```

Expected	Got	
<input checked="" type="checkbox"/> In main() function a = 20 In test() function a = 40 In main() function a = 55	<input checked="" type="checkbox"/> In main() function a = 20 In test() function a = 40 In main() function a = 55	<input checked="" type="checkbox"/>

Passed all tests! ✓

if e

9 Rain Sunday

Search

11:36 ENG IN 15-01-2025



RESHMASHRI S 2024-IT

R2

REC-CIS

Dashboard / My courses / GE23131-PUC-2024 / Assessment-03-Decision Making and Branching - if, ... / Local and Global Variables

Navigation

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...](#)[Assessment-01-Overview of C, Constants, Variables ...](#)[Week-02-Operators and Expressions, Managing Input ...](#)[Assessment-02-Operators and Expressions, Managing](#)

...

Local and Global Variables

 [Done](#)[Re-attempt quiz](#)

Time limit: 1 hour

Grading method: Highest grade

Your attempts

Attempt 1

Status	Finished
Started	Monday, 23 December 2024, 5:33 PM
Completed	Sunday, 8 December 2024, 12:40 PM
Duration	15 days 4 hours

[Review](#)

Local and Global Variables: Attempt review | REC-CIS - Google Chrome

Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=112384&cmid=89

REC-CIS

[Finish review](#)

Question 1
Correct
Marked out of 1.00
[Flag question](#)

A local variable is declared inside a function.

A **local variable** is visible only inside their function, only statements inside function can access that local variable.

Local variables are declared when the function execution started and local variables gets destroyed when control exits from function.

Let us consider an example:

```
#include <stdio.h>
void test();
int main()
{
    int a = 22, b = 44;
    test();
    printf("Values in main() function a = %d and b = %d\n", a, b);
    return 0;
}

void test()
{
    int a = 50, b = 80;
    printf("Values in test() function a = %d and b = %d\n", a, b);
}
```

In the above code we have 2 functions main() and test(), in these functions local variables are declared with same variable names a and b but they are different.

Operating System calls the main() function at the time of execution. the **local variables** with in the main() are created when the main() starts execution.

when a call is made to test() function, first the control is transferred from main() to test(), next the local variables with in the test() are created and they are available only with in the test() function.



Local and Global Variables: Attempt review | REC-CIS - Google Chrome

Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=112384&cmid=89

REC-CIS

After completion of execution of test() function, the local variables are destroyed and the control is transferred back to the main() function.

See & retype the below code which will demonstrate about local variables.

```
#include <stdio.h>
void test();

int main()
{
    int a = 9, b = 99;
    test();
    printf("Values in main() function a = %d and b = %d\n", a, b);
    return 0;
}

void test()
{
    int a = 5, b = 55;
    printf("Values in test() function a = %d and b = %d\n", a, b);
}
```

For example:

Result
Values in test() function a = 5 and b = 55 Values in main() function a = 9 and b = 99

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 void test();
```

if e

9 Rain Sunday

Search

myhp

11:35 ENG IN 15-01-2025

Local and Global Variables: Attempt review | REC-CIS - Google Chrome

Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=112384&cmid=89

REC-CIS

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 void test();
3 int main()
4 {
5     int a=9,b=99;
6     test();
7     printf("Values in main() function a = %d and b = %d\n",a,b);
8     return 0;
9 }
10 void test()
11 {
12     int a=5, b=55;
13     printf("Values in test() function a = %d and b = %d\n",a,b);
14 }
```

	Expected	Got	
✓	Values in test() function a = 5 and b = 55 Values in main() function a = 9 and b = 99	Values in test() function a = 5 and b = 55 Values in main() function a = 9 and b = 99	✓

Passed all tests! ✓

if e

9 Rain Sunday

Search

myhp

11:35 ENG IN 15-01-2025

Local and Global Variables: Attempt review | REC-CIS - Google Chrome

Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=112384&cmid=89

REC-CIS

Question 2

Correct
Marked out of 1.00

Flag question

Global variables are declared outside of any function.

A **global variable** is visible to any every function and can be used by any piece of code.

Unlike **local variable**, **global variables** retain their values between function calls and throughout the program execution.

Let us consider an example:

```
#include <stdio.h>
int a = 20; // Global declaration
void test();
int main()
{
    printf("In main() function a = %d\n", a); // Prints 20
    test();
    a = a + 15; // Uses global variable
    printf("In main() function a = %d\n", a); // Prints 55
    return 0;
}
void test()
{
    a = a + 20; // Uses global variable
    printf("In test() function a = %d\n", a); // Prints 40
}
```

In the above code the **global variable** a is declared outside of all the functions. So, the variable a can be accessed in every function.

Operating System calls the main() function at the time of execution. the variable a has no local declaration, so it access the global variable a.

In test() function also there is no local declaration of variable a, the variable a gets access from the global.

The global variables are destroyed only after completion of execution of entire program.

See & retyp the below code which will demonstrate about global variables.

Local and Global Variables: Attempt review | REC-CIS - Google Chrome

Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=112384&cmid=89

REC-CIS

```
#include <stdio.h>

int a = 20;

void test();

int main()
{
    printf("In main() function a = %d\n", a);
    test();
    a = a + 15;
    printf("In main() function a = %d\n", a);
    return 0;
}

void test()
{
    a = a + 20;
    printf("In test() function a = %d\n", a);
}
```

For example:

Result
In main() function a = 20 In test() function a = 40 In main() function a = 55

Answer: (penalty regime: 0 %)

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

if e

9 Rain Sunday

Search

Windows Start button

File

File Explorer

Microsoft Edge

myhp

Windows Store

Help

Bookmarks

OneDrive

Google Chrome

Cloud

ENG IN

11:35

15-01-2025

Local and Global Variables: Attempt review | REC-CIS - Google Chrome

Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=112384&cmid=89

REC-CIS

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int a=20;
3 void test();
4 int main()
5 {
6     printf("In main() function a = %d\n",a);
7     test();
8     a=a+15;
9     printf("In main() function a = %d\n",a);
10    return 0;
11 }
12 void test()
13 {
14     a=a+20;
15     printf("In test() function a = %d\n",a);
16 }
```

Expected	Got	
<input checked="" type="checkbox"/> In main() function a = 20 In test() function a = 40 In main() function a = 55	<input checked="" type="checkbox"/> In main() function a = 20 In test() function a = 40 In main() function a = 55	<input checked="" type="checkbox"/>

Passed all tests! ✓

if e

9 Rain Sunday

Search

11:36 ENG IN 15-01-2025

Local and Global Variables: Attempt review | REC-CIS - Google Chrome

Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=112384&cmid=89

REC-CIS

Question 3
Correct
Marked out of 1.00
[Flag question](#)

Local variables are declared and used **inside a function** (or) in a **block of statements**.

Local variables are created at the time of function call and destroyed when the function execution is completed.

Local variables are accessible only within the particular function where those variables are declared.

Global variables are declared outside of all the function blocks and these variables can be used in all functions.

Global variables are created at the time of program beginning and reside until the end of the entire program.

Global variables are accessible in the entire program.

If a **local** and **global** variable have the same name, then **local variable** has the **highest precedence** to access within the function.

Let us consider an example:

```
#include <stdio.h>
void change();
int x = 20; // Global Variable x

int main()
{
    int x = 10; // Local Variable x
    change();
    printf("%d", x); // The value 10 is printed
    return 0;
}
void change()
{
    printf("%d", x); // The value 20 is printed
}
```

In the above code the global and local variables have the same variable name x, but they are different.

if e

9 Rain Sunday

Search

myhp

11:36 ENG IN 15-01-2025

Local and Global Variables: Attempt review | REC-CIS - Google Chrome

Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=112384&cmid=89

REC-CIS

In main() function the **local** variable x is only accessed, so it prints the value 10.

In change() function the variable x is not declared locally so it access **global** variable x, so it prints 20.

See & retype the below code which will demonstrate about local and global variables.

```
#include <stdio.h>

int x = 15;

void change1(int x)
{
    printf("In change1() function x = %d\n", x);
}

void change2()
{
    printf("In change2() function x = %d\n", x);
}

int main()
{
    int x = 10;
    printf("In main() function x = %d\n", x);
    change1(x);
    change2();
    printf("In main() function x = %d\n", x);
    return 0;
}
```

For example:

if e

9 Rain Sunday

Search

myhp

11:36 ENG IN 15-01-2025

Local and Global Variables: Attempt review | REC-CIS - Google Chrome

Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=112384&cmid=89

REC-CIS

}

For example:

Result
In main() function x = 10 In change1() function x = 10 In change2() function x = 15 In main() function x = 10

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int x=15;
3 void change1(int x)
4 {
5     printf("In change1() function x = %d\n",x);
6 }
7 void change2()
8 {
9     printf("In change2() function x = %d\n",x);
10}
11 int main()
12 {
13     int x = 10;
14     printf("In main() function x = %d\n",x);
15     change1(x);
16     change2();
17     printf("In main() function x = %d\n",x);
18     return 0;
19 }
```

if e

9 Rain Sunday

Search

myhp

11:36 ENG IN 15-01-2025

Local and Global Variables: Attempt review | REC-CIS - Google Chrome

Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=112384&cmid=89

REC-CIS

```
5     printf("In change1() function x = %d\n",x);
6 }
7 void change2()
8 {
9     printf("In change2() function x = %d\n",x);
10 }
11 int main()
12 {
13     int x = 10;
14     printf("In main() function x = %d\n",x);
15     change1(x);
16     change2();
17     printf("In main() function x = %d\n",x);
18     return 0;
19 }
```

Expected	Got
✓ In main() function x = 10 In change1() function x = 10 In change2() function x = 15 In main() function x = 10	In main() function x = 10 In change1() function x = 10 In change2() function x = 15 In main() function x = 10

Passed all tests! ✓

Finish review



Navigation

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 Data...](#)[Assessment-01-Overview of C, Constants, Variables ...](#)[Week-02-Operators and Expressions, Managing Input ...](#)[Assessment-02-Operators and Expressions, Managing](#)

...

[Week-03-Decision](#)

Different categories of Functions

 [Done](#)[Re-attempt quiz](#)

Time limit: 3 hours

Grading method: Highest grade

Your attempts

Attempt 1

Status	Finished
Started	Monday, 23 December 2024, 5:33 PM
Completed	Sunday, 8 December 2024, 1:30 PM
Duration	15 days 4 hours

[Review](#)

Local and Global Variables

Jump to

Week-04-01-Practice Session-Coding

Different categories of Functions: Attempt review | REC-CIS - Google Chrome

Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=112421&cmid=90

REC-CIS

Quiz navigation

1 2 3 4 5 6
7 8

Show one page at a time

Finish review

Status Finished
Started Monday, 23 December 2024, 5:33 PM
Completed Sunday, 8 December 2024, 1:30 PM
Duration 15 days 4 hours

Question 1
Correct
Marked out of 1.00
Flag question

All the **C** functions can be called either with **arguments** or without arguments in a C program. These functions may or may not **return values** to the calling function.

Depending on the **arguments** and **return values** functions are classified into 4 categories.

1. Function without arguments and without return value
2. Function with arguments and without return value
3. Function without arguments and with return value
4. Function with arguments and with return value

When a function has **no arguments**, it does not receive any data from the calling function.

Similarly, when a function **does not return a value**, the calling function does not receive any data from the called function.

In effect, there is no data transfer between the calling function and the called function in the category **function without arguments and without return value**.

Let us consider an example of a function without arguments and without return value:

```
#include <stdio.h>
void india_capital(void);
int main()
{
    india_capital();
    return 0;
}
```

Local and Global Variables Jump to Week-04-01_Practice Session_Coding

NIFTY +0.33% 11:37 ENG IN 15-01-2025

Different categories of Functions: Attempt review | REC-CIS - Google Chrome

Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=112421&cmid=90

REC-CIS

```
{  
    india_capital();  
    return 0;  
}  
void india_capital()  
{  
    printf("New Delhi is the capital of India\n");  
}
```

In the above sample code the function void india_capital(void); specifies that the function does not receive any arguments and does not return any value to the main() function.

Identify the below errors and correct them.

For example:

Result
New Delhi is the capital of India

Answer: (penalty regime: 0 %)

Reset answer

```
1 #include <stdio.h>  
2  
3 void india_capital(void);  
4  
5 int main()  
6 {  
7     india_capital();  
8     return 0;  
9 }  
10  
11 void india_capital()  
12 {
```

Local and Global Variables Jump to Week-04-01_Practice Session_Coding

9 NIFTY +0.33% 11:37 ENG IN 15-01-2025

Different categories of Functions: Attempt review | REC-CIS - Google Chrome

Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=112421&cmid=90

REC-CIS

Reset answer

```
1 #include <stdio.h>
2
3 void india_capital(void);
4
5 int main()
6 {
7     india_capital();
8     return 0;
9 }
10
11 void india_capital()
12 {
13     printf("New Delhi is the capital of India\n");
14 }
```

Expected	Got
 New Delhi is the capital of India	 New Delhi is the capital of India

Passed all tests! 

Question 2 Local and Global Variables

Search  myhp  myhp  ENG IN 11:37 15-01-2025

NIFTY +0.33% Week-04-01_Practice Session_Coding

Different categories of Functions: Attempt review | REC-CIS - Google Chrome

Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=112421&cmid=90

REC-CIS

Question 2
Correct
Marked out of 1.00
Flag question

Write a C program to demonstrate functions without arguments and without return value.

Write the functions **print()** and **hello()**.

The output is:

```
...***...
Hello! REC
...***...
```

For example:

Result

```
...***...
Hello! REC
...***...
```

Answer: (penalty regime: 0 %)

Reset answer

```
1 #include <stdio.h>
2
3 // Write the functions
4
5 int main()
6 {
7     printf("...***...\n");
8     printf("Hello! REC\n");
9     printf("...***...");
10    return 0;
11 }
```

Local and Global Variables Jump to Week-04-01_Practice Session_Coding

NIFTY +0.33% 11:37 15-01-2025

Different categories of Functions: Attempt review | REC-CIS - Google Chrome

Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=112421&cmid=90

REC-CIS

```
1 #include <stdio.h>
2 // Write the functions
3
4 int main()
5 {
6     printf("...***...\n");
7     printf("Hello! REC\n");
8     printf("...***...");
9     return 0;
10 }
```

	Expected	Got	
✓	...***...	...***...	✓
	Hello! REC	Hello! REC	
	...***...	...***...	

Passed all tests! ✓

Local and Global Variables Jump to Week-04-01_Practice Session_Coding

Top Stories 9 Zero visibility in... Search

11:38 ENG IN 15-01-2025

Not secure

rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=112421&cmid=90

REC-CIS

Question 3

Correct

Marked out of
1.00[Flag question](#)

When a function definition has **arguments**, it receives data from the calling function.

The **actual arguments** in the function call must correspond to the **formal parameters** in the function definition, i.e. the number of actual arguments must be the same as the number of formal parameters, and each actual argument must be of the same data type as its corresponding formal parameter.

The **formal parameters** must be valid variable names in the function definition and the **actual arguments** may be variable names, expressions or constants in the function call.

The variables used in actual arguments must be assigned values before the **function call** is made. When a function call is made, copies of the values of actual arguments are passed to the **called function**.

What occurs inside the function will have no effect on the variables used in the **actual argument** list. There may be several different calls to the same function from various places with a program.

Let us consider an example of a function with arguments and without return value:

```
#include <stdio.h>
void largest(int, int);
int main()
{
    int a, b;
    printf("Enter two numbers : ");
    scanf("%d%d", &a, &b);
    largest(a, b);
    return 0;
}
void largest(int x, int y)
{
    if (x > y)
    {
        printf("Largest element = %d\n", x);
    }
}
```

Local and Global Variables

Jump to

Week-04-01_Practice Session_Coding

11:38

15-01-2025

Different categories of Functions: Attempt review | REC-CIS - Google Chrome

Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=112421&cmid=90

REC-CIS

```
        printf("Largest element = %d\n", x);
    }
else
{
    printf("Largest element = %d\n", y);
}
}
```

In the above sample code the function void largest(int, int); specifies that the function receives two integer arguments from the **calling function** and does not return any value to the **called function**.

When the function call largest(a, b) is made in the main() function, the values of actual arguments a and b are copied in to the formal parameters x and y.

After completion of execution of largest(int x, int y) function, it does not return any value to the main() function. Simply the control is transferred to the main() function.

Fill in the missing code in the below program to find the largest of two numbers using **largest()** function.

For example:

Input	Result
27 18	Largest element = 27
13 17	Largest element = 17

Answer: (penalty regime: 0 %)

Reset answer

```
1 #include <stdio.h>
2
3 void largest(int, int);
4
```

Local and Global Variables Jump to Week-04-01_Practice Session_Coding

Top Stories 9 Zero visibility in... Search

11:38 ENG IN 15-01-2025

Different categories of Functions: Attempt review | REC-CIS - Google Chrome

Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=112421&cmid=90

REC-CIS

Answer: (penalty regime: 0 %)

Reset answer

```
1 #include <stdio.h>
2
3 void largest(int, int);
4
5 int main()
6 {
7     int a, b;
8     scanf("%d%d", &a, &b);
9     largest(a,b);
10    return 0;
11 }
12
13 void largest(int x,int y)
14 {
15
16     if (x>y)
17     {
18
19         printf("Largest element = %d\n", x);
20     }
21     else
22     {
23         printf("Largest element = %d\n", y);
24     }
25 }
```

	Input	Expected	Got	
✓	27 18	Largest element = 27	Largest element = 27	✓

Local and Global Variables Jump to Week-04-01_Practice Session_Coding

9 Top Stories Zero visibility in... Search

11:38 ENG IN 15-01-2025

Different categories of Functions: Attempt review | REC-CIS - Google Chrome

Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=112421&cmid=90

REC-CIS

```
4
5 int main()
6 {
7     int a, b;
8     scanf("%d%d", &a, &b);
9     largest(a,b);
10    return 0;
11 }
12
13 void largest(int x,int y)
14 {
15
16     if (x>y)
17     {
18
19         printf("Largest element = %d\n", x);
20     }
21     else
22     {
23
24         printf("Largest element = %d\n", y);
25     }
}
```

	Input	Expected	Got	
✓	27 18	Largest element = 27	Largest element = 27	✓
✓	13 17	Largest element = 17	Largest element = 17	✓

Passed all tests! ✓

Different categories of Functions: Attempt review | REC-CIS - Google Chrome

Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=112421&cmid=90

REC-CIS

Question 4
Correct
Marked out of 1.00
Flag question

Fill the missing code to understand the concept of a function with arguments and without return value.

Note: Take **pi** value as **3.14**

The below code is to find the area of circle using functions.

For example:

Input	Result
11.23	Area of circle = 395.994476

Answer: (penalty regime: 0 %)

Reset answer

```
1 #include <stdio.h>
2
3 void area_circle(float radius)
4 {
5     float area = 3.14*radius*radius;
6     printf("Area of circle = %.6f\n",area);
7 }
8
9 int main()
10 {
11     float radius;
12     scanf("%f",&radius);
13     area_circle(radius);
14     return 0;
15 }
16
```

Local and Global Variables Jump to Week-04-01_Practice Session_Coding

Top Stories Zero visibility in... Search

11:38 ENG IN 15-01-2025

Different categories of Functions: Attempt review | REC-CIS - Google Chrome

Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=112421&cmid=90#question-112549-4

REC-CIS

Answer: (penalty regime: 0 %)

[Reset answer](#)

```
1 #include <stdio.h>
2
3 void area_circle(float radius)
4 {
5     float area = 3.14*radius*radius;
6     printf("Area of circle = %.6f\n",area);
7 }
8
9 int main()
10 {
11     float radius;
12     scanf("%f",&radius);
13     area_circle(radius);
14     return 0;
15 }
16
```

	Input	Expected	Got	
✓	11.23	Area of circle = 395.994476	Area of circle = 395.994476	✓

Passed all tests! ✓

9+ High UV Now

Search

myhp

12:18 15-01-2025

Different categories of Functions: Attempt review | REC-CIS - Google Chrome

Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=112421&cmid=90

REC-CIS

Question 5
Correct
Marked out of 1.00
[Flag question](#)

When a function has **no arguments**, it does not receive any data from the calling function.

When a function **return a value**, the calling function receives data from the called function.

Let us consider an example of a function without arguments and with return value:

```
#include <stdio.h>
int sum(void);
int main()
{
    printf("\nSum of two given values = %d\n", sum());
    return 0;
}
int sum() {
    int a, b, total;
    printf("Enter two numbers : ");
    scanf("%d%d", &a, &b);
    total = a + b;
    return total;
}
```

In the above sample code the function `int sum(void);` specifies that the function does not receive any arguments but return a value to the **calling function**.

Fill in the missing code in the below program to find sum of two integers.

For example:

Input	Result
9 5	Sum of two given values = 14

Local and Global Variables Jump to Week-04-01_Practice Session_Coding

9 Top Stories Zero visibility in... Search myhp 11:39 ENG IN 15-01-2025

Different categories of Functions: Attempt review | REC-CIS - Google Chrome

Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=112421&cmid=90

REC-CIS

Input	Result
9 5	Sum of two given values = 14
45 78	Sum of two given values = 123

Answer: (penalty regime: 0 %)

Reset answer

```
1 #include <stdio.h>
2
3 int sum(void);
4
5 int main()
6 {
7     printf("Sum of two given values = %d\n", sum());
8     return 0;
9 }
10
11 int sum()
12 {
13     int a,b,total;
14     scanf("%d %d",&a,&b);
15     total = a+b;
16     return total;
17 }
```

Local and Global Variables Jump to Week-04-01_Practice Session_Coding

Top Stories Zero visibility in... Search

11:39 ENG IN 15-01-2025

Different categories of Functions: Attempt review | REC-CIS - Google Chrome

Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=112421&cmid=90

REC-CIS

Reset answer

```
1 #include <stdio.h>
2
3 int sum(void);
4
5 int main()
6 {
7     printf("Sum of two given values = %d\n", sum());
8     return 0;
9 }
10
11 int sum()
12 {
13     int a,b,total;
14     scanf("%d %d",&a,&b);
15     total = a+b;
16     return total;
17 }
```

	Input	Expected	Got	
✓	9 5	Sum of two given values = 14	Sum of two given values = 14	✓
✓	45 78	Sum of two given values = 123	Sum of two given values = 123	✓

Passed all tests! ✓

Local and Global Variables Jump to Week-04-01_Practice Session-Coding

Top Stories Search myhp 11:39
Zero visibility in... ENG IN 15-01-2025

Different categories of Functions: Attempt review | REC-CIS - Google Chrome

Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=112421&cmid=90

REC-CIS

Question 6

Correct

Marked out of 1.00

Flag question

When a **function definition** has arguments, it receives data from the calling function.

After taking some desired action, only one value will be returned from **called function** to **calling function** through the return statement.

If a function returns a value, the **function call** may appear in any expression and the returned value used as an operand in the evaluation of the expression.

Let us consider an example of a function with arguments and with return value:

```
#include <stdio.h>
int largest(int, int, int);
int main()
{
    int a, b, c;
    printf("Enter three numbers : ");
    scanf("%d%d%d", &a, &b, &c);
    printf(" Largest of the given three numbers = %d\n", largest(a, b, c));
    return 0;
}
int largest(int x, int y, int z)
{
    if ((x > y) && (x > z))
    {
        return x;
    }
    else if (y > z)
    {
        return y;
    }
    else
    {
        return z;
    }
}
```

Local and Global Variables Jump to Week-04-01_Practice Session_Coding

Top Stories 9 Zero visibility in... Search myhp 11:39 ENG IN 15-01-2025

Different categories of Functions: Attempt review | REC-CIS - Google Chrome

Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=112421&cmid=90

REC-CIS

In the above sample code the function int largest(int, int, int); specifies that the function receives three values and returns a value to the **calling function**.

Fill in the missing code in the below program to find the largest of three numbers using **largest()** function.

For example:

Input	Result
99 49 29	Largest of the given three numbers = 99
45 67 35	Largest of the given three numbers = 67

Answer: (penalty regime: 0 %)

Reset answer

```
1 #include <stdio.h>
2
3 int largest(int, int, int);
4
5 int main()
6 {
7     int a, b, c;
8     scanf("%d%d%d", &a, &b, &c);
9     printf("Largest of the given three numbers = %d\n", largest(a, b, c));
10    return 0;
11 }
12
13 int largest(int x, int y, int z)
14 {
15
16     if ((x>y)&&(x>z))
17     {
18
19         return x;
20     }
21 }
```

Local and Global Variables Jump to Week-04-01_Practice Session_Coding

11:39 ENG IN 15-01-2025

Different categories of Functions: Attempt review | REC-CIS - Google Chrome

Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=112421&cmid=90

REC-CIS

```
8     scanf( "%d%d%d", &a, &b, &c);
9     printf("Largest of the given three numbers = %d\n",largest(a, b, c));
10    return 0;
11 }
12
13 int largest(int x, int y, int z)
14 {
15
16     if ((x>y)&&(x>z))
17     {
18
19         return x;
20     }
21     else if (y>z)
22     {
23
24         return y ;
25     }
26     else
27     {
28         return z;
29     }
30 }
```

	Input	Expected	Got	
✓	99 49 29	Largest of the given three numbers = 99	Largest of the given three numbers = 99	✓
✓	45 67 35	Largest of the given three numbers = 67	Largest of the given three numbers = 67	✓

Passed all tests! ✓

Different categories of Functions: Attempt review | REC-CIS - Google Chrome

Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=112421&cmid=90

REC-CIS

Question 7
Correct
Marked out of 1.00
[Flag question](#)

Fill in the missing code in the below code to understand about function with arguments and with return value.

The below code is to find the factorial of a given number using functions.

For example:

Input	Result
3	Factorial of a given number 3 = 6

Answer: (penalty regime: 0 %)

[Reset answer](#)

```
1 #include <stdio.h>
2 int factorial(int num)
3 {
4     int i, fact = 1;
5     for(i=1;i<=num;i++)
6     {
7         fact *= i;
8     }
9     return fact;
10 }
11 int main()
12 {
13     int number;
14     scanf("%d",&number);
15     printf("Factorial of a given number %d = %d\n",number,factorial(number));
16     return 0;
17 }
```

Local and Global Variables Jump to Week-04-01_Practice Session_Coding

9+ 29°C Partly sunny Search myhp 11:40 ENG IN 15-01-2025

Different categories of Functions: Attempt review | REC-CIS - Google Chrome

Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=112421&cmid=90

REC-CIS

Answer: (penalty regime: 0 %)

[Reset answer](#)

```
1 #include <stdio.h>
2 int factorial(int num)
3 {
4     int i, fact = 1;
5     for(i=1;i<=num;i++)
6     {
7         fact *= i;
8     }
9     return fact;
10 }
11 int main()
12 {
13     int number;
14     scanf("%d",&number);
15     printf("Factorial of a given number %d = %d\n",number,factorial(number));
16     return 0;
17 }
```

	Input	Expected	Got	
✓	3	Factorial of a given number 3 = 6	Factorial of a given number 3 = 6	✓

Passed all tests! ✓

Local and Global Variables Jump to Week-04-01_Practice Session_Coding

9+ 29°C Partly sunny Search myhp 11:40 ENG IN 15-01-2025

Different categories of Functions: Attempt review | REC-CIS - Google Chrome

Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=112421&cmid=90

REC-CIS

Question 8
Correct
Marked out of 1.00
[Flag question](#)

Write a C program to demonstrate functions without arguments and with return value.

The below code is used to check whether the given number is a prime number or not.

Write the function **prime()**.

Sample Input and Output:

```
5
The given number is a prime number
```

For example:

Input	Result
5	The given number is a prime number
27	The given number is not a prime number
121	The given number is not a prime number
1	The given number is not a prime number

Answer: (penalty regime: 0 %)

[Reset answer](#)

```
1 #include <stdio.h>
2 int main()
3 {
4     int num,i,flag=0;
5     scanf("%d",&num);
6     if(num==0||num==1)
7     {
```

Local and Global Variables Jump to Week-04-01_Practice Session_Coding

9+ 29°C ENG IN 11:40 15-01-2025

Different categories of Functions: Attempt review | REC-CIS - Google Chrome

Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=112421&cmid=90

REC-CIS

Answer: (penalty regime: 0 %)

Reset answer

```
1 #include <stdio.h>
2 int main()
3 {
4     int num,i,flag=0;
5     scanf("%d",&num);
6     if(num==0||num==1)
7     {
8         flag = 1;
9     }
10    for(i=2;i*i<=num;i++)
11    {
12        if(num%i==0)
13        {
14            flag = 1;
15            break;
16        }
17    }
18    if(flag==0)
19    {
20        printf("The given number is a prime number\n");
21    }
22    else
23    {
24        printf("The given number is not a prime number\n");
25    }
26    return 0;
27 }
```

	Input	Expected	Got
Local and Global Variables			

Jump to: Week-04-01_Practice Session_Coding

9+ 29°C Partly sunny

Search

myhp

11:40 15-01-2025

Different categories of Functions: Attempt review | REC-CIS - Google Chrome

Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=112421&cmid=90

REC-CIS

```
9 }  
10 for(i=2;i*i<=num;i++)  
11 {  
12     if(num%i==0)  
13     {  
14         flag = 1;  
15         break;  
16     }  
17 }  
18 if(flag==0)  
19 {  
20     printf("The given number is a prime number\n");  
21 }  
22 else  
23 {  
24     printf("The given number is not a prime number\n");  
25 }  
26 return 0;  
27 }  
28 }
```

	Input	Expected	Got	
✓	5	The given number is a prime number	The given number is a prime number	✓
✓	27	The given number is not a prime number	The given number is not a prime number	✓
✓	121	The given number is not a prime number	The given number is not a prime number	✓
✓	1	The given number is not a prime number	The given number is not a prime number	✓

Passed all tests! ✓

Local and Global Variables Jump to Week-04-01_Practice Session_Coding

9+ 29°C ENG IN 11:40 15-01-2025