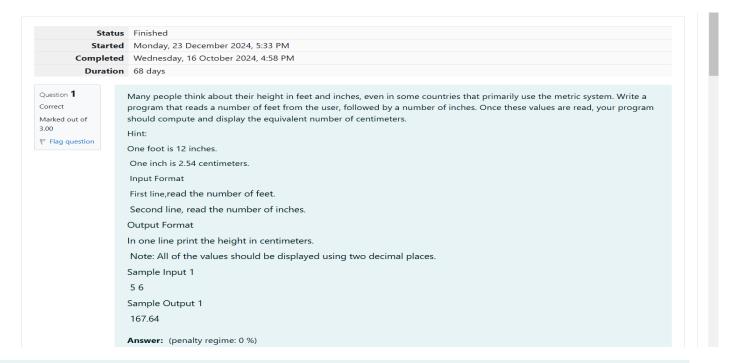
# Week-02-Operators and Expressions, Managing Input and Output Operations

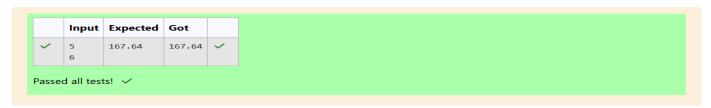
Week-02-01-Practive Session-coding

ROLL NO: 241801294

NAME: THARUN N







Question **2**Correct
Marked out of 5.00

Flag question

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

Input Format

First line, read the first number.

Second line, read the second number.

**Output Format** 

First line, print the sum of a and b

Second line, print the difference when b is subtracted from a

Third line, print the product of a and b

Fourth line, print the quotient when a is divided by b

Fifth line, print the remainder when a is divided by b

Sample

Input 1 100 6

Sample Output

106 94 600 16 4

Answer: (penalty regime: 0 %)

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

```
#include<stdio.h>
    int main()
 2
 3 🔻
    {
4
        int num1,num2;
5
        scanf("%d", &num1);
 6
        scanf("%d", &num2);
7
        int sum = num1 + num2;
8
        int dif = num1 - num2;
int mul = num1 * num2;
9
10
        int div_ = num1 / num2;
11
        int mod = num1 % num2;
        printf("%d\n%d\n%d\n%d\n", sum,dif,mul,div_,mod);
12
13
        return 0;
14
```

	Input	Expected	Got	
~	100	106	106	~
	6	94	94	
		600	600	
		16	16	
		4	4	

Passed all tests! ✓

Question **3**Correct
Marked out of

7.00

▼ Flag question

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

Input Format

Read the number of day old loaves.

**Output Format** 

First line, print Regular price: price

Second line, print Discount: discount

Third line, print Total: total

Note: All of the values should be displayed using two decimal places.

Sample Input 1

10

Sample Output 1

Regular price: 34.90

Discount: 20.94

Total: 13.96

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

```
#include<stdio.h>
2
    int main ()
3
    {
4
        float count,discountprice,regprice,disprice,finalprice;
5
        scanf("%f", &count);
        discountprice = (3.49*(60.0/100.0));
6
        regprice = count*3.49;
7
8
        disprice = count*discountprice;
9
        finalprice = regprice-disprice;
        printf("Regular price: %.2f\nDiscount: %.2f\nTotal: %.2f",regprice,disprice,finalprice);
10
11
        return 0;
12
```

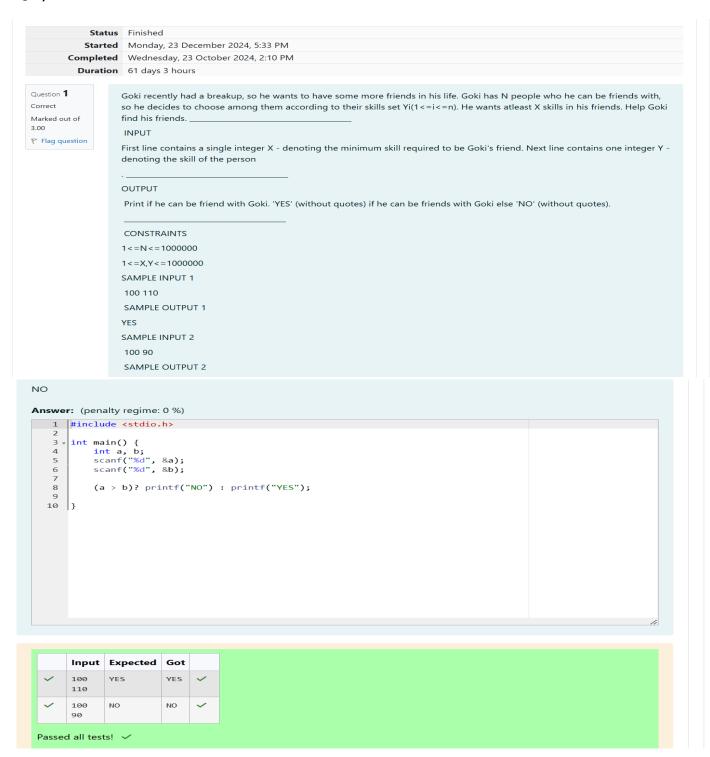
	Input	Expected	Got	
~	10	Regular price: 34.90 Discount: 20.94 Total: 13.96	Regular price: 34.90 Discount: 20.94 Total: 13.96	<b>~</b>

Passed all tests! <

# Week-02-02-Practive Session-coding

ROLL NO: 241801294

NAME: THARUN N



Question **2**Correct
Marked out of 5.00

Flag question

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

Input Format

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

Output Format

Print the number of handshakes.

Constraints

0 < N < 106

SAMPLE INPUT 1

1

SAMPLE OUTPUT

0

SAMPLE INPUT 2

2

SAMPLE OUTPUT 2

1

Explanation Case 1: The lonely board member shakes no hands, hence 0. Case 2: There are 2 board members, 1 handshake takes place.

Answer: (penalty regime: 0 %)

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

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

int main()

int N;

scanf("%d",&N);

N=N*(N-1)/2;
printf("%d\n", N);

}
```

```
| Input | Expected | Got |
| ✓ | 1 | 0 | 0 | ✓
| ✓ | 2 | 1 | 1 | ✓
```

Passed all tests! 🗸

Question **3**Correct
Marked out of 7.00

Flag question

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

Input:

Three space separated integers.

Output:

Maximum integer value

SAMPLE INPUT

861

SAMPLE OUTPUT

8

Explanation Out of given numbers, 8 is maximum.

Answer: (penalty regime: 0 %)

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

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

	Input	Expected	Got	
~	81 26 15	81	81	~

Passed all tests! <

## **Arithmetic Operators-Optional**

ROLL NO: 241801294

NAME: THARUN N

Q1)

Started Monday, 23 December 2024, 5:33 PM

Completed Wednesday, 23 October 2024, 2:50 PM

**Duration** 61 days 2 hours

Question 1
Correct
Marked out of 1.00
F Flag question

An operator is a special symbol used to manipulate data. The data items that the operators act upon are called operands.

The operator that works on a single operand is called a unary operator and that which works on two operands is known as a binary operator.

**C** provides many types of operators. They are: Arithmetic, Unary, Relational and equality, Logical, Assignment, Conditional, Bitwise and Special operators.

In C, we have 5 arithmetic operators:

#### **Operator Description**

- Used for addition
- Used for subtraction
- \* Used for multiplication
- / Used for division
- % Remainder/Modulus operator for finding remainder

**Arithmetic operators** are applied on **numeric operands**. Thus the operands can be **integers**, **floats** or **characters** (Since a character is internally represented by its numeric code).

The **remainder operator** (%) requires that both the operands be **integers** and the second operand be **non-zero**. Similarly the **division operator** (/) requires that the second operand be **non-zero**.

The format for usage of arithmetic operator is as follows: operand1operatoroperand2

According to the coding conventions in C, a single space should be provided to the left and to the right of an operator.

```
The table given below demonstrates the use of various arithmetic operators using two variables num1 and num2 of
type int with values 10 and 3 respectively:
Expression
              Result
num1 + num2 13
num1 - num2 7
num1 * num2 30
num1 / num2 3
num1 % num21
Read the code given below to understand the usage of arithmetic operators. Retype in the space provided.
int main()
  int num1 = 10, num2 = 3;
  printf("Addition Result = %d\n", (num1 + num2));
  printf("Subtraction Result = %d\n", (num1 - num2));
  printf("Multiplication Result = %d\n", (num1 * num2));
  printf("Division Result = %d\n", (num1 / num2));
  printf("Remainder = %d", (num1 % num2));
  return 0;
Answer: (penalty regime: 0 %)
```

```
Answer: (penalty regime: 0 %)
       #include<stdio.h>
       int main()
    2
   3 ▼
       {
    4
            int num1 = 10, num2 = 3;
            printf("Addition Result = %d\n", (num1 + num2));
    5
    6
            printf("Subtraction Result = %d\n", (num1 - num2));
            printf("Multiplication Result = %d\n", (num1*num2));
printf("Division Result = %d\n", (num1/num2));
    7
    8
   9
            printf("Remainder = %d", (num1%num2));
  10
            return 0;
  11
```

~	Addition Result = 13 Subtraction Result = 7 Multiplication Result = 30	Addition Result = 13 Subtraction Result = 7 Multiplication Result = 30	~
	Division Result = 3 Remainder = 1	Division Result = 3 Remainder = 1	

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

**Division** of one integer by another integer is referred to as **integer** division. This operation always results in an integer with truncated quotient.

If a division operation is carried out with two floating point numbers or with one floating point number and one integer, the result will be a floating point quotient.

The table given below demonstrates the usage of various **arithmetic operators** using two variables num1 and num2 of type float with values 12.5 and 2.0 respectively:

### **Expression Result**

num1 + num2 14.500000 num1 - num2 10.500000 num1 \* num2 25.000000 num1 / num2 6.250000

Note that the **remainder operator** (%) is not applicable for **floating point numbers**.

In the program given below, type the missing code to find the **result** of applying different **arithmetic operators** on **floating point numbers**.

Answer: (penalty regime: 0 %)

```
Reset answer
    #include <stdio.h>
 2
 3
    int main()
 4 ▼ {
 5
         float num1 = 12.5, num2 = 2.0;
 6
         printf("Result of addition = %f\n", (num1 + num2));
 7
         printf("Result of subtraction = %f\n", (num1 - num2));
 8
         printf("Result of multiplication = %f\n", (num1 * num2));
         printf("Result of division = %f\n", (num1 / num2));
 9
 10
         return 0;
 11
    |}
```

Expected Got
Result of addition = 14.500000  Result of subtraction = 10.500000  Result of multiplication = 25.000000  Result of division = 6.2500000  Result of division = 6.2500000  Result of division = 6.2500000

Question **3**Correct
Marked out of 1.00

Flag question

The table given below demonstrates the use of various **arithmetic operators** using two variables c1 and c2 of type char with values 'A' and 'D' respectively:

```
Expression Result
```

```
c1 65
c1 + c2 133
c1 + c2 + 5 138
c1 + c2 + '5' 186
```

In the above examples, the character 'A' is substituted with its ASCII value 65 and 'D' is substituted with 68.

The character '5' is substituted with its ASCII value 53. The integer value 5 is used as it is.

The following table demonstrates the usage of various **arithmetic operators** using two variables a and b of type int with values 11 and -3 respectively:

#### **Expression Result**

```
a + b 8
a - b 14
a * b -33
a / b -3
a % b 2
```

In the program given below, type the missing code to find the **result** of applying different **arithmetic operators** on **char** data type values.

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

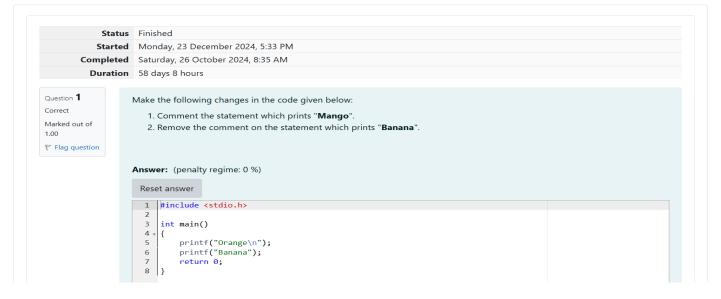
```
Reset answer
      #include <stdio.h>
  2
  3
      int main()
  4 🔻
      {
  5
          char c1 = 'A', c2 = 'D';
  6
         printf("c1 = %d\n", c1);
  7
         printf("c1 + c2 = %d\n", (c1 + c2));
         printf("c1 + c2 + 5 = %d\n", (c1 + c2 + 5));
printf("Result = %d", (c1 + c2 + '5'));
  8
  9
 10
          return 0;
 11 }
```

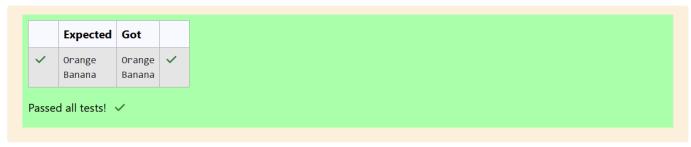
# **Coding-Comment Lines and Tokens-Optional**

ROLL NO: 241801294

## NAME: THARUN N

# Q1)





### Q2)

Question **2**Correct
Marked out of 1.00

F Flag question

As mentioned earlier, a computer program is a collection of instructions or statements.

A C program usually consists of multiple statements.

Each statement is composed of one or more of the  ${\it three}$  given below:

- 1. Comments
- 2. Whitespace characters
- 3. Tokens

In a computer program, a comment is used to mark a section of code as non-executable.

Comments are mainly used for two purposes:

- 1. To mark a section of executable code as non-executable, so that the compiler ignores it during compilation.
- 2. To provide remarks or an explanation on the working of the given section of code in plain English, so that a fellow programmer can read and understand the code.

In C, there are two types of comments:

- 1. **end-of-line comment**: It starts with //. The content that follows the // and continues till the end of that line is a comment. It is also called as **single-line comment**.
- 2. **traditional comment**: It starts with /\* and ends with \*/. The content between /\* and \*/ is the comment. It is also called as **multi-line comment**.

The code given below shows the two types of comments:

```
/*
C programming language was developed by Dennis Ritchie.
This is called a header comment which is used to describe what this program would do. As you can notice the comment is spanning across multiple lines.
*/
#include <stdio.h>
```

```
int main()
{
    int num1 = 10, num2 = 20;
    printf("sum of two numbers = %d", num1 + num2);
    return 0;
}//end of the main() function - this is an example of a end-of-line comment
Read the code given below to understand the different types of comments. Retype in the space provided.

Given below are 3 important points regarding comments:

1. There should not be any space between the two forward slashes in //, i.e., // is incorrect. Similarly, there should not be any space between the slash and star characters in /* and */, i.e., /* and */ are incorrect.

2. Comments do not nest, i.e., /* and */ comment has no special meaning inside a // comment.

3. One should not write comments inside character literals (i.e., characters enclosed between single-quotes). Comments inside String literals (i.e., text enclosed between double-quotes) are treated as part of the String's content.

Content to be reproduced

/*
This is a sample C program
```

```
#include <stdio.h>
int main()
{
    // this is an end of line comment
    printf("I love C Language!");
    return 0;
```

Answer: (penalty regime: 0 %)

developed by REC

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

{
    //this is am end of line comment
    printf("I love C Language!");
    return 0;
}
```

	Expected	Got	
~	I love C Language!	I love C Language!	~
Passec	d all tests! 🗸		

Question **3**Correct
Marked out of 1.00

Flag question

In **C**, the backslash character \ is used to mark an escape sequence. An **Escape Sequence** is an escape character \ followed by a normal character. For example: \n or \t.

The presence of the escape character changes the meaning of the character which follows it. For example, when the string literal "Hello\tWorld" is printed, the result is seen as

```
Hello World
```

In the string literal "Hello\tWorld", \t represents the TAB character.

Similarly, if we want to print a **double quote** inside a double-quoted string literal, we need to escape the **double quote** by using the escape character \. For example :

```
printf("Hello \" (Quote)");
```

The code given above will produce the following output: Hello " (Quote)

Given below are a few points regarding escape sequences:

- Each escape sequence has a unique ASCII value as shown in the table given below.
- Each and every combination of an escape sequence starts with backslash \.
- Although an escape sequence consists of two characters, it represents a single special character in the given context.

Escape sequences and their ASCII codes:

Character	Bell Backspace	Horizontal tab	Vertical tab				Double Quotation	Single Quotation	Question mark	Backslasł	n Null
Escape Sequence	\a \b	\t	\v	\n	\f	\r	\"	\'	\?	\	\0
ASCII value	007 008	009	011	010	012	013	034	039	063	092	000

Read the code given below and retype in the space provided. **Note** the effects of \t and \n in the resulting output when executed successfully.

Content to be reproduced

#include <stdio.h>

```
#include <stdio.h>
int main()
  printf("One Two");
  printf("Three\n");
  printf("Four\nFive\n");
  return 0;
Answer: (penalty regime: 0 %)
      #include<stdio.h>
  2
      int main()
  3 ▼
      {
           printf("One Two");
  4
           printf("Three\n");
  5
```



printf("Four\nFive\n");

return 0;

Passed all tests! <

6 7

8 }

```
Question 4
Correct
Marked out of 1.00
Flag question
```

The code given below contains text that prints "DennisRitchieBrianKernighan".

Make the suggested changes to the code so that it prints "DennisRitchieBrianKernighan" as shown below.

Dennis Ritchie Brian Kernighan

To make the required changes, follow the steps given below to introduce the SPACE character and the \n new line character appropriately:

- 1. Insert a space between "Dennis" and "Ritchie". Make sure that no extra space or any other character apart from space are inserted.
- 2. Insert a \n between "Ritchie" and "Brian" Make sure that no extra space or any other character apart from \n are inserted.
- 3. Insert a space between "Brian" and "Kernighan". Make sure that no extra space or any other character apart from space are inserted.

Answer: (penalty regime: 0 %)

Reset answer

```
Reset answer

1  #include <stdio.h>
2  
3  int main()  
4  v  
5  printf("Dennis Ritchie\nBrian Kernighan");  
6  return 0;  
7  }
```

```
Reset answer

1  #include <stdio.h>
2  int main()
4  r
5  printf("Dennis Ritchie\nBrian Kernighan");
6  return 0;
7 }
```

# **Coding-Variables and Keywords**

ROLL NO: 241801294

NAME: THARUN N

```
Status Finished
           Started Monday, 23 December 2024, 5:33 PM
       Completed Saturday, 26 October 2024, 8:40 AM
          Duration 58 days 8 hours
Question 1
                   Read the code given below to learn naming conventions in identifiers.
Correct
Marked out of
                   For example, consider the program given below:
1.00
                   #include <stdio.h>
Flag question
                   int main()
                           int age = 2; // age is an integer variable
                           int firstNumber = 2; // firstNumber is an integer variable
                           // If there are two or more words in an identifier/variable - User can also use "camel case" style to declare a
                           int second_number = 3; // second_number is an integer variable
                           // Any space cannot be used between two words of an identifier/variable; User can use underscore (_) instead of
                           int _i_am_also_a_valid_identifier = 4; // _i_am_also_a_valid_identifier is an integer variable
                           // An identifier/variable name must be start with an alphabet or underscore (_) only, no other special characte
                           printf("age = %d\n", age);
                           printf("firstNumber = %d\n", firstNumber);
                           printf("second_number = %d\n", second_number);
                           printf("_i_am_also_a_valid_identifier = %d\n", _i_am_also_a_valid_identifier);
```

Fill in the missing code in the below program to print the values of the given variables.

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

```
Reset answer
  1 #include <stdio.h>
         int main()
   4 *
                int age = 2;
   6
                int firstNumber = 2;
int second_number = 3;
                int _i_am_also_a_valid_identifier = 4;
   8
               printf("age = %d\n", age);
printf("firstNumber = %d\n", firstNumber); // Fill in the missing code
printf("second_number = %d\n", second_number); // Fill in the missing code
printf("i_am_also_a_valid_identifier = %d\n", _i_am_also_a_valid_identifier); // Fill in the missing
   9
 10
 11
 12
 13
                return 0;
 14 }
```

```
Expected

Got

age = 2
firstNumber = 2
second_number = 3
_i_am_also_a_valid_identifier = 4

Passed all tests! 

Got

age = 2
firstNumber = 2
second_number = 3
_i_am_also_a_valid_identifier = 4

i_am_also_a_valid_identifier = 4
```

# **Coding-Syntax of main() function**

ROLL NO: 241801294

```
Status Finished
            Started Monday, 23 December 2024, 5:33 PM
         Completed Saturday, 26 October 2024, 8:44 AM
           Duration 58 days 8 hours
Question 1
                     In {\bf C} programming language, execution of the code starts with a function called main.
Correct
Marked out of
                     We shall learn more about functions in the later sections. For now, we can safely assume that function is the name given to a set
1.00
                     of one or more executable statements. main() is a user defined function, i.e., a user (a programmer) writes the code for the
Flag question
                     main() function.
                    While executing a C program, the Operating System (OS) only calls the main() function in that program.
                    When the OS executes a program, the program usually returns an integer value 0 if the execution of that program is successful.
                     In C, main() can be written in such a way that it returns a an int.
                     #include <stdio.h>
                     int main()
                         printf("Sample main() function with int as return type!");
                         return 0;// 0 value indicates that the execution is successful
                    If the programmer does not specify any return type, the return type is by default considered as int.
                     The name of the main() function should always be in lowercase, i.e., if a function is written as Main(), it is not the main function
                     which is called by the OS.
                     Read the code given below to familiarize yourself with the syntax of main() function. Retype in the space provided.
```

```
Read the code given below to familiarize yourself with the syntax of main() function. Retype in the space provided.
#include <stdio.h>
int main()
{
  printf("Impossible is nothing!");
  return 0;
}
Answer: (penalty regime: 0 %)
      #include<stdio.h>
      int main()
   2
  3 🔻
      {
           printf("Impossible is nothing!");
  4
   5
           return 0;
   6 }
```

	Expected	Got	
~	Impossible is nothing!	Impossible is nothing!	~
Passed	d all tests! 🗸		

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

```
Expected

Hello, # is a preprocessor in C Hello, # is a preprocessor in C

Passed all tests! ✓
```

## Q3)

Question **3**Correct
Marked out of 1.00

Flag question

6

7 }

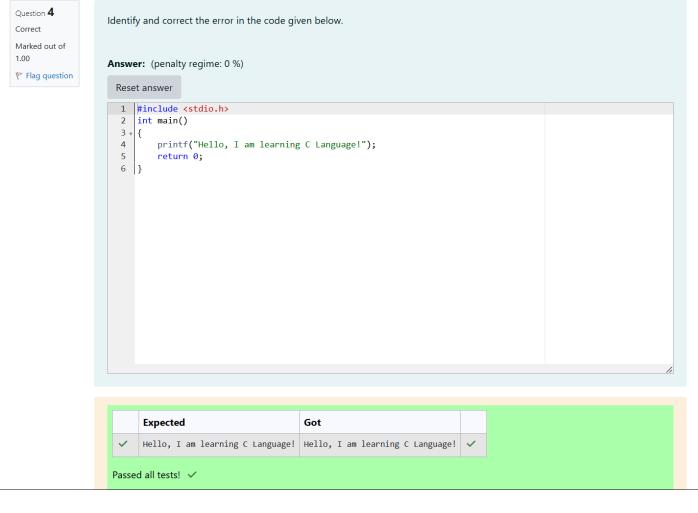
return 0;

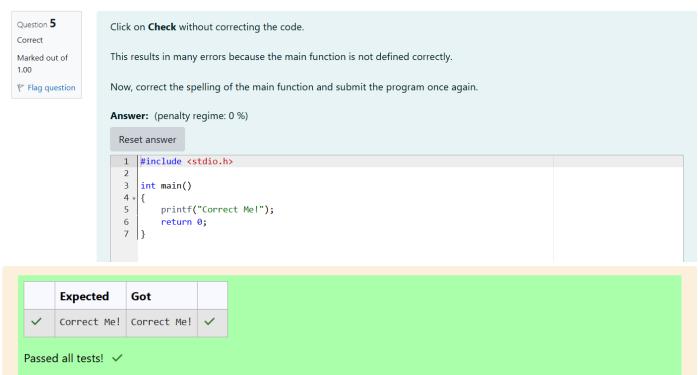
```
Expected

Hello, float data type allocates 4 bytes in memory

Hello, float data type allocates 4 bytes in memory

Passed all tests! ✓
```





# int Data Type

ROLL NO: 241801294

```
Status Finished
            Started Monday, 23 December 2024, 5:33 PM
        Completed Saturday, 26 October 2024, 9:28 AM
          Duration 58 days 8 hours
Question 1
                     In the program given below, we shall learn how to assign values to int data type from binary, octal, hex and character literals.
Correct
                     Read the code given below and retype in the space provided.
Marked out of
1.00
                     #include <stdio.h>

₱ Flag question

                     int main()
                       int binaryThree = 0b11;
                       printf("binaryThree value = %d\n", binaryThree);
                       int octalEight = 010;
                       printf("octalEight value = %d\n", octalEight);
                       int hexTen = 0xA;
                       printf("hexTen value = %d\n", hexTen);
                       int asciiValueOfOne = '1';
                       printf("asciiValueOfOne value = %d\n", asciiValueOfOne);
                       int asciiValueOfA = 'A';
                       printf("asciiValueOfA value = %d\n", asciiValueOfA);
                       return 0:
```

```
Answer: (penalty regime: 0 %)
      #include<stdio.h>
   2
       int main()
   3 ▼ {
   4
           int binaryThree = 0b11;
   5
          printf("binaryThree value = %d\n",binaryThree);
   6
          int octalEight = 010;
   7
          printf("octalEight value = %d\n",octalEight);
   8
           int hexTen = 0xA;
          printf("hexTen value = %d\n",hexTen);
   9
  10
           int asciiValueOfOne = '1';
           printf("asciiValueOfOne value = %d\n",asciiValueOfOne);
  11
  12
           int asciiValueOfA ='A';
          printf("asciiValueOfA value = %d\n",asciiValueOfA);
  13
  14
           return 0;
  15
```

Question Z
Correct
Marked out of 1.00

F Elag question

```
Expected

Got

Given integers are num1 = 15, num2 = 25
Sum of 2 given numbers = 40

Given integers are num1 = 15, num2 = 25
Sum of 2 given numbers = 40

Passed all tests!
```

## Q3)

Question **3**Correct
Marked out of 1.00

F Flag question

To print unsigned values on the console, use %u format character instead of %d in the **printf()** function.

Whenever an attempt is made to assign a negative number to an **unsigned int** ( For eg: unsigned int num = -1;) the compiler does not flag it as an **error**. Instead, it will automatically convert the negative number to a positive number as shown below:

```
unsigned int num = -1;
The value stored in num = unsigned int maximum_value + 1 - num;
The final value in num = 4294967295 (in a 32-bit processing system)
```

In the program given below, fill in the missing code to add two integer numbers.

In the program given below, fill in the missing format characters to print signed and unsigned values.

Answer: (penalty regime: 0 %)

## Reset answer

```
#include <stdio.h>
2
3
    int main()
4 *
5
        signed int number1 = -20, number2 = 20;
6
        unsigned int number3 = -1, number4 = 1;
        printf("Given signed values are %d and %u\n", number1, number2); // Fill the correct format charact
7
8
        printf("Given unsigned values are %u and %u\n", number3, number4); // Fill the correct format chara
9
        return 0;
10 }
```

✓ Given signed values are -20 and 20 Given signed values are -20 and 20	
Given unsigned values are 4294967295 and 1 Given unsigned values are 4294967295 and	
ssed all tests! ✓	

```
Question 4
Correct
Marked out of
1.00
Frag question
```

```
Identify the error and correct the code. [Hint: Verify if all variables are declared before they are first used.]
Answer: (penalty regime: 0 %)
 Reset answer
       #include <stdio.h>
    3
       int main()
    4
    5
            int num, number1 = 20, number2 = 30;
    6
           num = number1 - number2;
           printf("The difference of the two given numbers = %d\n", num);
    8
            return 0;
    9
  10
```

```
Expected

The difference of the two given numbers = -10

The difference of the two given numbers = -10

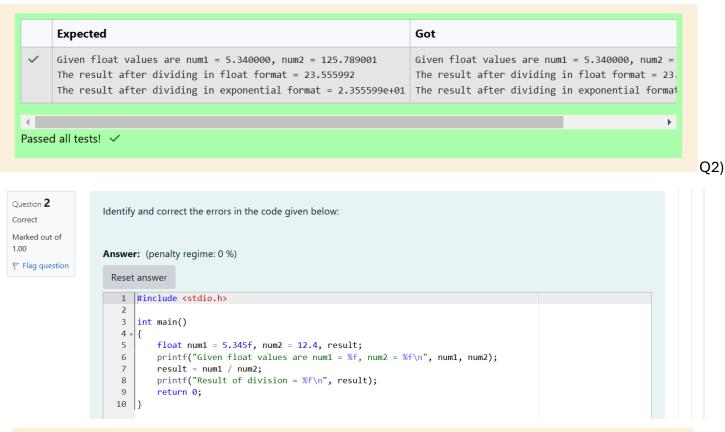
Passed all tests! ✓
```

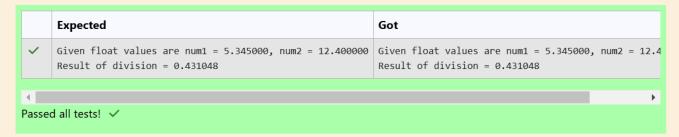
# **float Data Type**

ROLL NO: 241801294

NAME: THARUN N

```
Status Finished
           Started Monday, 23 December 2024, 5:33 PM
       Completed Saturday, 26 October 2024, 9:59 AM
          Duration 58 days 7 hours
                   Identify and correct the errors in the code given below:
Correct
                   Expected Output:
Marked out of
                   Given float values are num1 = 5.340000, num2 = 125.789001
1.00
                   The result after dividing in float format = 23.555992
Flag question
                   The result after dividing in exponential format = 2.355599e+01
                   Answer: (penalty regime: 0 %)
                     Reset answer
                         #include <stdio.h>
                       3
                          int main()
                       4
                          {
                              float num1 = 5.340000, num2 = 125.789001f, result;
                              printf("Given float values are num1 = %.6f, num2 = %.6f\n", num1, num2);
                       6
                              printf("The result after dividing in float format = %.6f\n", result);
                       9
                              printf("The result after dividing in exponential format = %e\n", result);
                      10
                      11 }
```





# **Relational and Equality Operators-Optional**

ROLL NO: 241801294

NAME: THARUN N

Status Finished

Started Monday, 23 December 2024, 5:33 PM

Completed Thursday, 31 October 2024, 5:14 PM

Duration 53 days

Question **1**Correct
Marked out of 1.00

Flag question

Relational and equality operators are used to test or compare two numeric values or numeric expressions.

In C, Relational and equality operators when applied on the operands, produce an integer value which is either 0 or 1 and these are often referred to as logical values. The value 0 represents false and the value 1 represents true.

In  ${\bf C}$ , there are  ${f four}$  relational and  ${f two}$  equality operators as given below:

#### **Operator Description**

- > Checks for greater-than condition
- >= Checks for greater-than-or-equals condition
- < Checks for less-than condition
- <= Checks for less-than-or-equals condition.</p>
- == Checks if two values are equal
- != Checks if two values are unequal

The format for usage of **relational** and **equality operators** is as follows: **operand1**operator**operand2** 

According to the coding conventions in C, a single space should be provided to the left and to the right of the operator.

The table given below demonstrates the use of various **relational and equality operators** using variables int num1 = 7;, float num2 = 5.5;, char ch = 'w':

Expression	Interpreta	ition Result Value
(num1 > 5)	true	1
((num1 + num2) <= 10)	false	0

```
Expression
                            Interpretation Result Value
(num1 > 5)
                            true
                                           1
((num1 + num2) <= 10)
                            false
                                           0
(ch == 119)
                                           1
                            true
(ch != 'p')
                                           1
                            true
(ch >= 10 * (num1 + num2)) false
                                           0
Read the code given below and retype in the space provided.
#include <stdio.h>
int main()
{
  int num1 = 7;
  float num2 = 5.5;
  char ch = 'w';
  printf("Result1 = %d\n", (num1 > 5));
  printf("Result2 = %d\n", ((num1 + num2) <= 10));
  printf("Result3 = %d\n", (ch == 119));
  printf("Result4 = %d\n", (ch != 'p'));
  printf("Result5 = %d", (ch >= 10 * (num1 + num2)));
  return 0;
```

```
Answer: (penalty regime: 0 %)
   1 #include<stdio.h>
   2
       int main()
   3 ₹ {
   4
            int num1 = 7;
   5
           float num2 = 5.5;
           char ch = 'w';
   6
           printf("Result1 = %d\n", (num1 > 5));
   7
   8
           printf("Result2 = %d\n", ((num1 + num2) <=10));
           printf("Result3 = %d\n", (ch == 119));
   9
           printf("Result4 = %d\n", (ch != 'p'));
printf("Result5 = %d", (ch >=10* (num1 + num2)));
  10
  11
  12
            return 0;
  13 }
```

```
Expected

Result1 = 1
Result2 = 0
Result3 = 1
Result4 = 1
Result4 = 1
Result5 = 0

Result5 = 0

Result5 = 0

Result5 = 0
```

Logical

## **Operators-Optional**

ROLL NO: 241801294

NAME: THARUN N

Q1)



Question 1
Correct
Marked out of 1.00

Flag question

Logical operators are used to perform logical operations on the given expressions.

An expression containing a logical operator returns either 0 (or) 1 depending on the evaluation of the expression to either false or true respectively.

Note: In C, false is represented as 0 (zero) and all non-zero values can be treated as true.

Given below are the **three** logical operators in **C**:

### **Operator Description Meaning**

&& logical AND It returns true when both conditions are true, else, it returns false
|| logical OR | It returns true if atleast one of the conditions is true

! logical NOT It returns true when the given expression is false and returns false when the given expression is true

According to the coding conventions in C, a single space should be provided to the left and to the right of the operator.

The below table demonstrates the use of various **relational and equality operators** using variables int num1 =  $7_{ii}$  float num2 =  $5.5_{ij}$  char ch = 'w':

# Expression Interpretation Result Value (num1 >= 6) && (ch == 'w') true 1

(num2 < 11) && (num1 > 100) false 0

```
    Expression
    Interpretation Result Value

    (num1 >= 6) && (ch == 'w')
    true
    1

    (num2 < 11) && (num1 > 100)
    false
    0

    (ch != 'p') || ((num1 + num2) <= 10) true</td>
    1

    !(num1 > (num2 + 1))
    false
    0

    !(num1 <= 3)</td>
    true
    1
```

Read the code given below and retype in the space provided.

```
#include <stdio.h>

int main()
{
    int num1 = 7;
    float num2 = 5.5;
    char ch = 'w';
    printf("Result1 = %d\n", ((num1 >= 6) && (ch == 'w')));
    printf("Result2 = %d\n", ((num2 < 11) && (num1 > 100)));
    printf("Result3 = %d\n", ((ch != 'p') || ((num1 + num2) <= 10)));
    printf("Result4 = %d\n", !(num1 > (num2 + 1)));
    printf("Result5 = %d\n", !(num1 <= 3));
    return 0;
```

```
Answer: (penalty regime: 0 %)
   1 #include<stdio.h>
   2
      int main()
   3 ₹ {
          int num1 = 7;
   4
   5
          float num2 = 5.5;
          char ch = 'w';
   6
   7
          printf("Result1 = %d\n", ((num1 >=6) && (ch == 'w')));
          printf("Result2 = %d\n", ((num2 <11) && (num1 >=10)));
   8
   9
          printf("Result3 = %d\n", ((ch != 'p') || ((num1 + num2) <= 10)));</pre>
          printf("Result4 = %d\n", !(num1 > (num2 + 1)));
  10
          printf("Result5 = %d\n", !(num1 <= 3));</pre>
  11
  12
          return 0;
  13
      }
  14
```

```
Expected Got

Result1 = 1 Result1 = 1 
Result2 = 0 Result2 = 0
Result3 = 1 Result3 = 1
Result4 = 0 Result4 = 0
Result5 = 1 Result5 = 1

Passed all tests!
```

## **Unary Operators-Optional**

ROLL NO: 241801294

NAME: THARUN N

Q1

```
Status Finished
            Started Monday, 23 December 2024, 5:33 PM
        Completed Thursday, 31 October 2024, 5:46 PM
           Duration 52 days 23 hours
Question 1
                     Read the code given below to understand the working of unary operators. Retype in the space provided.
                     #include <stdio.h>
Marked out of
1.00

▼ Flag question

                     int main()
                       int x = 16;
                       printf("+x = %d\n", (+x));
                       printf("-x = %d\n", (-x));
                       printf("x = %d\n", x);
                       printf("++x = %d\n", (++x));
                       printf("x = %d\n", x);
                       printf("x++ = %d\n", (x++));
                       printf("x = %d\n", x);
                       printf("--x = %d\n", (--x));
                       printf("x = %d\n", x);
                       printf("x-- = %d\n", (x--));
                       printf("x = %d", x);
                       return 0;
```

```
Expected Got
     +x = 16
               +x = 16
      -x = -16 -x = -16
     x = 16
               x = 16
     ++x = 17 ++x = 17
     x = 17
              x = 17
     x = 18 x = 18 --x = 17
     x = 17 x = 17
x - = 17 x - = 17
     x = 16
              x = 16
Passed all tests! <
```

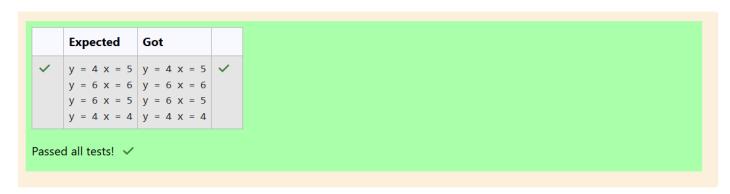
## Q2)

```
Question 2
Correct
Marked out of
1.00
```

Flag question

```
Read the code given below to understand the working of increment and decrement operators. Retype in the space provided.
#include <stdio.h>
int main()
  int x = 4, y;
  y = x + +;
  printf("y = %d x = %d\n", y, x);
  y = ++x;
  printf("y = %d x = %d\n", y, x);
  y = x--;
  printf("y = %d x = %d\n", y, x);
  y = --x;
  printf("y = %d x = %d\n", y, x);
  return 0;
```

```
#include<stdio.h>
2
   int main()
3 ₹ {
4
      int x = 4, y;
5
     y = x++;
     printf("y = %d x = %d\n", y, x);
6
7
     y = ++x;
     printf("y = %d x = %d\n", y, x);
8
9
      y= x--;
      printf("y = %d x = %d\n", y, x);
10
11
      y = --x;
      printf("y = %d x = %d\n", y, x);
12
13
14
       return 0;
15 }
```



# **Assignment Operator-Optional**

ROLL NO: 241801294

NAME: THARUN N

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

Completed Thursday, 31 October 2024, 6:02 PM

Duration 52 days 23 hours
```

```
Question 1
Correct
```

Marked out of 1.00

```
Flag question
```

```
Read the code given below to understand the usage of the assignment operator. Retype in the space provided.

#include <stdio.h>

int main()
{

int x = 24, y = 39, z = 45;

z = x + y;

y = z - y;

x = z - y;

printf("x = %d y = %d z = %d", x, y, z);

return 0;
}

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

Expected Got
x = 39 y = 24 z = 63 x = 39 y = 24 z = 63
ed all tests! 🗸

Question  ${\bf 2}$ 

Correct
Marked out of
1.00

▼ Flag question

```
#include <stdio.h>

int main()
{
    int x = 2, y = 18, z = 12;
        x += y;
    printf("x = %d\n", x);
    y *= 2;
    printf("y = %d\n", y);
    z /= 5;
    printf("z = %d\n", z);
    x %= 7;
    printf("x = %d\n", x);
    return 0;
}
```

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

```
1 #include<stdio.h>
2 int main()
3 ₹ {
4
      int x = 2, y = 18, z = 12;
     x += y;
5
     printf("x = %d\n", x);
6
7
     y *=2;
     printf("y = %d\n", y);
8
9
     z /=5;
10
     printf("z = %d\n", z);
11
      x %=7;
      printf("x = %d", x);
12
13
      return 0;
14 }
```

	Expected	Got	
~	x = 20 y = 36 z = 2 x = 6	x = 20 y = 36 z = 2 x = 6	<b>~</b>

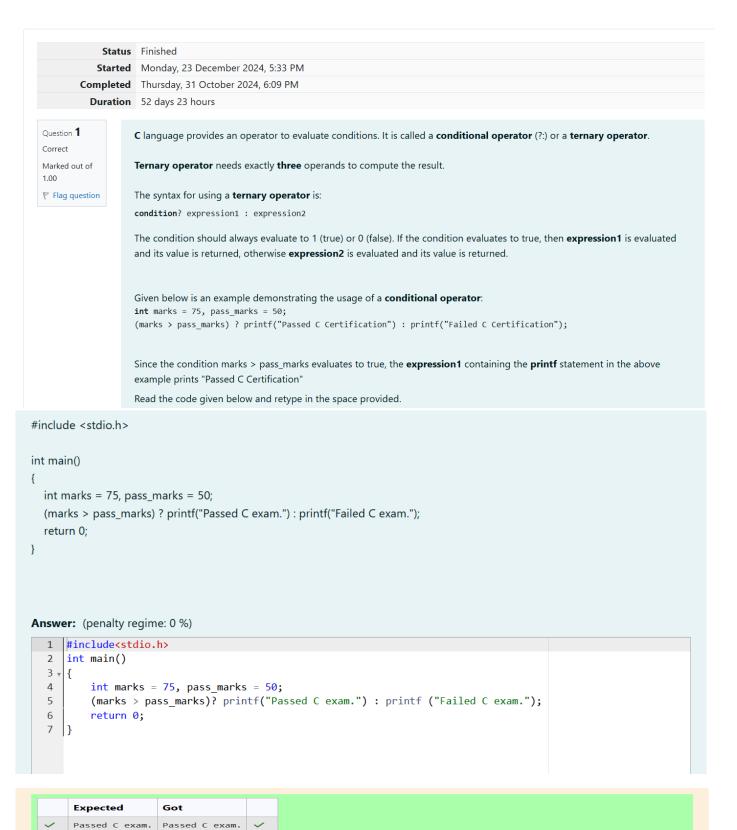
Passed all tests! <

## **Ternary Operator-Optional**

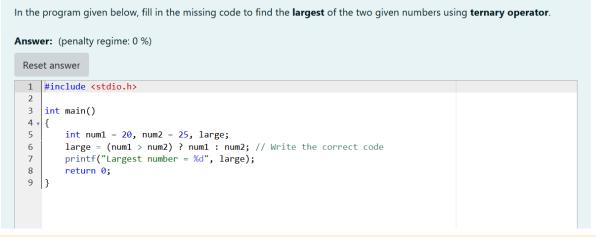
ROLL NO: 241801294

NAME: THARUN N

Passed all tests! 🗸







```
Expected

Largest number = 25 Largest number = 25 

Passed all tests! 

Got

V

Passed all tests!
```

## **Cricket Stadium**

ROLL NO: 241801294

NAME: THARUN N

Q1)



Question 1
Correct
Marked out of
1.00
F Flag question

There was a large ground in center of the city which is rectangular in shape. The Corporation decides to build a Cricket stadium in the area for school and college students, But the area was used as a car parking zone. In order to protect the land from using as an unauthorized parking zone, the corporation wanted to protect the stadium by building a fence. In order to help the workers to build a fence, they planned to place a thick rope around the ground. They wanted to buy only the exact length of the rope that is needed. They also wanted to cover the entire ground with a carpet during rainy season. They wanted to buy only the exact quantity of carpet that is needed. They requested your help. Can you please help them by writing a program to find the exact length of the rope and the exact quantity of carpet that is required?

Input format:

Input consists of 2 integers. The first integer corresponds to the length of the ground and the second integer corresponds to the breadth of the ground.

**Output Format:** 

Output Consists of two integers. The first integer corresponds to the length. The second integer corresponds to the quantity of carpet required.

Sample Input:

50

20

Sample Output:

140

1000

### For example:

Input	Result
50	140
20	1000

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

```
1 #include<stdio.h>
2 int main()
3 v
4     int s1, s2;
5     scanf("%d", &s1);
6     scanf("%d", &s2);
7     printf("%d\n%d", ((s1*2) + (s2*2)), (s1*s2));
8
9 }
```

## **Sports Day Celebration**

ROLL NO: 241801294

NAME: THARUN N

Q1)



Question 1
Correct
Marked out of 1.00

F Flag question

Training for sports day has begun and the physical education teacher has decided to conduct some team games. The teacher wants to split the students in higher secondary into equal sized teams. In some cases, there may be some students who are left out from the teams and he wanted to use the left out students to assist him in conducting the team games. For instance, if there are 50 students in a class and if the class has to be divided into 7 equal sized teams, 7 students will be there in each team and 1 student will be left out. That 1 student will assist the PET. With this idea in mind, the PET wants your help to automate this team splitting task. Can you please help him out?

#### INPUT FORMAT

Input consists of 2 integers. The first integer corresponds to the number of students in the class and the second integer corresponds to the number of teams.

### **OUTPUT FORMAT:**

The output consists of two integers. The first integer corresponds to the number of students in each team and the second integer corresponds to the students who are left out.

### SAMPLE INPUT:

60

8

SAMPLE OUTPUT:

7

## For example:

Input	Result
60	7
8	4

## Answer: (penalty regime: 0 %)

```
1  #include<stdio.h>
2  int main()
3  * 
4     int p, t;
5     scanf("%d", &p);
6     scanf("%d", &t);
7     printf("%d\n%d", (p/t), (p%t));
8     }
```

	Input	Expected	Got	
~	60 8	7	7	<b>~</b>

Passed all tests! ✓

## **The Newspaper Agency**

ROLL NO: 241801294

NAME: THARUN N

Q1)



Question 1
Correct
Marked out of 1.00
F Flag question

Each Sunday, a newspaper agency sells w copies of a special edition newspaper for Rs.x per copy. The cost to the agency of each newspaper is Rs.y. 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 which it obtains only on Sundays. Can you please help them out by writing a program to compute the profit if w, x, and y are given?

INPUT FORMAT:

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

OUTPUT FORMAT:

The output consists of a single integer which corresponds to the profit obtained by the newspaper agency.

SAMPLE INPUT:

1000

2

SAMPLE OUTPUT:

900

#### For example:

Input	Result
1000	900
2	
1	

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

```
1
   #include<stdio.h>
2
    int main()
3 ▼ {
4
        int w, x, y;
        scanf("%d", &w);
5
        scanf("%d", &x);
6
        scanf("%d", &y);
7
8
9
        int t = (w * (x - y)) - 100;
        printf("%d", t);
10
11
```

	Input	Expected	Got	
<b>~</b>	1000 2 1	900	900	~

Passed all tests! <

## The Chronicles of Narnia

ROLL NO: 241801294

NAME: THARUN N

Q1)

**Duration** 52 days 19 hours

Question 1

Marked out of

Flag question

Correct

1.00

Status Finished Started Monday, 23 December 2024, 5:33 PM Completed Thursday, 31 October 2024, 9:35 PM

> Four kids Peter, Susan, Edmond and Lucy travel through a wardrobe to the land of Narnia. Narnia is a fantasy world of magic with mythical beasts and talking animals. While exploring the land of narnia Lucy found Mr. Tumnus the two legged stag , and she followed it, down a narrow path .She and Mr.Tumnus became friends and he offered a cup of coffee to Lucy in his small hut.It was time for Lucy to return to her family and so she bid good bye to Mr.Tumnus and while leaving Mr.Tumnus told that it is quite difficult to find the route back as it was already dark. He told her to see the trees while returning back and said that the first tree with two digits number will help her find the way and the way to go back to her home is the sum of digits of the tree and that numbered way will lead her to the tree next to the wardrobe where she can find the others. Lucy was already confused, so please help her in finding the route to her home....

Input Format:

Input consists of an integer corresponding to the 2-digit number.

Output Format:

Output consists of an integer corresponding to the sum of its digits.

**SAMPLE INPUT:** 

SAMPLE OUTPUT:

15

For example:

Input	Result
87	15

Answer: (penalty regime: 0 %)

```
#include<stdio.h>
 2
    int main()
 3 ,
    {
 4
         int n;
 5
        scanf("%d", &n);
 6
        int p = 0;
        while (n != 0) {
 7
 8
            p += (n\%10);
             n = n/10;
9
10
        printf("%d", p);
11
12
```

	Input	Expected	Got	
~	87	15	15	~
~	54	9	9	~

Passed all tests! <

## int Data Type

ROLL NO: 241801294

NAME: THARUN N

```
Status Finished
            Started Monday, 23 December 2024, 5:33 PM
        Completed Thursday, 31 October 2024, 9:52 PM
          Duration 52 days 19 hours
Question 1
                     In the program given below, we shall learn how to assign values to int data type from binary, octal, hex and character literals.
Correct
                     Read the code given below and retype in the space provided.
Marked out of
1.00
                     #include <stdio.h>
Flag question
                     int main()
                       int binaryThree = 0b11;
                       printf("binaryThree value = %d\n", binaryThree);
                       int octalEight = 010;
                       printf("octalEight value = %d\n", octalEight);
                       int hexTen = 0xA;
                       printf("hexTen value = %d\n", hexTen);
                       int asciiValueOfOne = '1';
                       printf("asciiValueOfOne value = %d\n", asciiValueOfOne);
                       int asciiValueOfA = 'A';
                       printf("asciiValueOfA value = %d\n", asciiValueOfA);
                       return 0;
```

```
Answer: (penalty regime: 0 %)
      #include<stdio.h>
   2
       int main()
   3 ,
       {
   4
           int binaryThree = 0b11;
   5
           printf("binaryThree value = %d\n", binaryThree);
           int octalEight =010;
   6
   7
           printf("octalEight value = %d\n", octalEight);
   8
           int hexTen = 0xA;
   9
           printf("hexTen value = %d\n", hexTen);
  10
           int asciiValueOfOne = '1';
  11
           printf("asciiValueOfOne value = %d\n", asciiValueOfOne);
  12
           int asciiValueOfA = 'A';
           printf("asciiValueOfA value = %d\n", asciiValueOfA);
  13
  14
```

```
Expected

Sot

binaryThree value = 3
octalEight value = 8
hexTen value = 10
asciiValueOfOne value = 49
asciiValueOfA value = 65

Passed all tests! 

binaryThree value = 3
octalEight value = 8
hexTen value = 10
asciiValueOfOne value = 49
asciiValueOfA value = 65

Passed all tests! 

binaryThree value = 3
octalEight value = 8
hexTen value = 10
asciiValueOfOne value = 49
asciiValueOfA value = 65

Passed all tests!
```

Correct

Marked out of
1.00

▼ Flag question

Question  ${\bf 2}$ 

In the program given below, fill in the missing code to add two integer numbers.

Answer: (penalty regime: 0 %)

Reset answer

	Expected	Got	
~	Given integers are num1 = 15, num2 = 25 Sum of 2 given numbers = 40	Given integers are num1 = 15, num2 = 25 Sum of 2 given numbers = 40	~

Passed all tests! ✓