Question **1**

Correct

Marked out of 3.00

Flag question

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

Hint:

One foot is 12 inches.

One inch is 2.54 centimeters.

Input Format

First line,read the number of feet.

Second line, read the number of inches.

Output Format

In one line print the height in centimeters.

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

Sample Input 1

5 6

Sample Output 1

One foot is 12 inches.

One inch is 2.54 centimeters.

Input Format

First line,read the number of feet.

Second line, read the number of inches.

Output Format

In one line print the height in centimeters.

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

Sample Input 1

5 6

Sample Output 1

167.64

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>
int main(){
    int ft,inch;
    float cm;
    scanf("%d %d",&ft,&inch);
    cm=(ft*12+inch)*2.54;
    printf("%.2f",cm);
}
```

	Input	Expected	Got	
✓	5 6	167.64	167.64	✓

Passed all tests! ✓

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

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 %)

```
#include <stdio.h>
int main(){
    int a,b,c,d,e,f,k;
    scanf("%d %d",&a,&b);
    c=a+b;
    d=a-b;
    k=a*b;
    e=a/b;
    f=a%b;
    printf("%d\n%d\n%d\n%d\n
}
```

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 %)

```
.h>

,e,f,k;
d",&a,&b);









n%d\n%d\n%d\n%d",c,d,k,e,f);

```

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

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 %)

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

    int day;
    float r,d,t;

    scanf("%d",&day);
    r=day*3.49;
    d=r*60/100;
    t=r-d;


    printf("Regular price: %
    printf("Discount: %.2f\n
    printf("Total: %.2f",t);
}
```

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 %)

```
stdio.h>
{

y;
r,d,t;

"%d",&day);
3.49;
/100;



("Regular price: %.2f\n",r);
("Discount: %.2f\n",d);
("Total: %.2f",t);

```

```
11
12
13 ("Regular price: %.2f\n",r);
14 ("Discount: %.2f\n",d);
15 ("Total: %.2f",t);
16
```

	Input	Expected	Go
✓	10	Regular price: 34.90 Discount: 20.94 Total: 13.96	Reg Dis Tot

Passed all tests! ✓

Finish review

Quiz navigation

1 2 3

Show one page at a time

Finish review

Question **1**

Correct

Marked out of 3.00

Flag question

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

_____

INPUT

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

. _____

OUTPUT

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

_____

CONSTRAINTS

1<=N<=1000000

1<=X,Y<=1000000

SAMPLE INPUT 1

Question **1**

Correct

Marked out of 3.00

Flag question

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

_____

INPUT

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

. _____

OUTPUT

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

_____

CONSTRAINTS

1<=N<=1000000

1<=X,Y<=1000000

SAMPLE INPUT 1

YES

SAMPLE INPUT 2

100 90

SAMPLE OUTPUT 2

NO

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>
int main(){
    int x,y;
    scanf("%d %d",&x,&y);
    if(x<=y){
        printf("YES");

    }
    else{
        printf("NO");
    }

}
```

	Input	Expected	Got	
✓	100 110	YES	YES	✓

	Input	Expected	Got	
✓	100 110	YES	YES	✓
✓	100 90	NO	NO	✓

Passed all tests! ✓

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.

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 %)

```c
#include <stdio.h>
int main(){
    int n;
    scanf("%d",&n);
    printf("%d",n*(n-1)/2);

}
```

	Input	Expected	Got	
✓	1	0	0	✓
✓	2	1	1	✓

Passed all tests! ✓

Question **3**

Incorrect

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

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

8 6 1

SAMPLE OUTPUT

8

Explanation Out of given numbers, 8 is maximum.

Answer: (penalty regime: 0 %)

```
1 #include <stdio.h>
2 int main(){
3     int a,b,c;
4     scanf("%d %d %d",&a,&b,)
5 }
```

Question **1**

Correct

Marked out of 1.00

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, Bi twise 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

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:
operand1`operator`**operand2**

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 % num2	1

Read the code given below to understand the usage of **arithmetic operators**. Retype in the space provided.

```
#include <stdio.h>

int main()

{

   int num1 = 10, num2 = 3;

   printf("Addition Result = %d\n", (num1 +
```

}

Answer: (penalty regime: 0 %)

```
#include <stdio.h>
int main()
{
    int num1=10,num2=3;
    printf("Addition Result
    printf("Subtraction Resu
    printf("Multiplication R
    printf("Division Result
    printf("Remainder = %d",
    return 0;
}
```

	Expected	Go
✓	Addition Result = 13 Subtraction Result = 7 Multiplication Result = 30 Division Result = 3 Remainder = 1	Add Sub Mul Div Rer

Passed all tests! ✓

}

Answer: (penalty regime: 0 %)

```
1
2
3
4
5   = %d\n",(num1+num2));
6  ult = %d\n",(num1-num2));
7  Result = %d\n",(num1*num2));
8   = %d\n",(num1/num2));
9  ,(num1%num2));
10
11
```

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

Passed all tests! ✓

Question **2**

Correct

Marked out of 1.00

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>

int main()
{
    float num1 = 12.5, num2
    printf("Result of additi
    printf("Result of subtra
    printf("Result of multip
    printf("Result of divisi
    return 0;
}
```

	Expected
✓	Result of addition = 14.500000 Result of subtraction = 10.500 Result of multiplication = 25. Result of division = 6.250000

Passed all tests! ✓

Answer: (penalty regime: 0 %)

Reset answer

```
1
2
3
4
5  ;
6  %f\n", (num1 + num2));
7   = %f\n", (num1 - num2));
8  ion = %f\n", (num1 * num2));
9  %f\n", (num1 / num2));
10
11
```

ult of addition = 14.500000 ult of subtraction = 10.500000 ult of multiplication = 25.000000 ult of division = 6.250000	✓

Passed all tests! ✓

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

Answer: (penalty regime: 0 %)

Reset answer

```c
#include <stdio.h>

int main()
{
    char c1 = 'A', c2 = 'D';
    printf("c1 = %d\n", c1);
    printf("c1 + c2 = %d\n",
    printf("c1 + c2 + 5 = %d
    printf("Result = %d", (c
    return 0;
}
```

	Expected	Got
✓	c1 = 65 c1 + c2 = 133 c1 + c2 + 5 = 138 Result = 186	c1 = 65 c1 + c2 = 13 c1 + c2 + 5 Result = 186

Passed all tests! ✓

Answer: (penalty regime: 0 %)

Reset answer

```
1
2
3
4
5  2 = 'D';
6  n", c1);
7  = %d\n", (c1 + c2));
8  + 5 = %d\n", (c1 + c2 + 5));
9   %d", (c1 + c2 + '5'));
10
11
```

ted	Got	
65 c2 = 133 c2 + 5 = 138 t = 186	c1 = 65 c1 + c2 = 133 c1 + c2 + 5 = 138 Result = 186	✓

Passed all tests! ✓

Question **1**

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 %)

space or any other character apart from space are inserted.

Answer: (penalty regime: 0 %)

Reset answer

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

	Expected	Got
✓	Dennis Ritchie Brian Kernighan	Dennis Ritchie Brian Kernigha

Passed all tests! ✓

Question **2**

space or any other character apart from space are inserted.

Answer: (penalty regime: 0 %)

Reset answer

```
1 >
2
3
4
5 s Ritchie\nBrian Kernighan");
6
7
```

xpected	Got	
ennis Ritchie rian Kernighan	Dennis Ritchie Brian Kernighan	✓

Passed all tests! ✓

Question **2**

Question **2**

Correct

Marked out of 1.00

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 **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 //.

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 develop
    This is called a header comment wh
    what this program would do. As you
    spanning across multiple lines.
*/

#include <stdio.h>

int main()
{
     int num1 = 10, num2 = 20;
     printf("sum of two numbers = %d",
     return 0;
}//end of the main() function - this is an
```

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

```
#include <stdio.h>

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

Answer: (penalty regime: 0 %)

```
1  #include <stdio.h>
2  int main()
3  {
4      //this ia an end of line
5      printf("I love C Language
6      return 0;
7  }
```

	Expected	Got
✓	I love C Language!	I love C La

Passed all tests! ✓

```
#include <stdio.h>

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

Answer: (penalty regime: 0 %)

```
e <stdio.h>
n()

his ia an end of line comment
ntf("I love C Language!");
urn 0;

```

	Got	
C Language!	I love C Language!	✓

Passed 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**\t**World", \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.

```
    printf("Four\nFive\n");
    return 0;
}
```

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>
int main(){
    printf("One Two");
    printf("Three\n");
    printf("Four\nFive\n");
    return 0;
}
```

	Expected	Got	
✓	One TwoThree Four Five	One TwoThree Four Five	✓

Passed all tests! ✓

Question **4**

Correct

Marked out of 1.00

Flag question

Make the following changes in the code given below:

1. Comment the statement which prints "**Mango**".
2. Remove the comment on the statement which prints "**Banana**".

Answer: (penalty regime: 0 %)

Reset answer

```
#include <stdio.h>

int main()
{
    printf("Orange\n");

    printf("Banana");
    return 0;
}
```

Answer: (penalty regime: 0 %)

Reset answer

```c
#include <stdio.h>

int main()
{
    printf("Orange\n");

    printf("Banana");
    return 0;
}
```

	Expected	Got	
✓	Orange Banana	Orange Banana	✓

Passed all tests! ✓

Question **1**

Correct

Marked out of 1.00

Flag question

Read the code given below to learn naming conventions in identifiers.

For example, consider the program given below:

```
#include <stdio.h>

int main()
{
        int age = 2; // age is an inte

        int firstNumber = 2; // firstN

        // If there are two or more wo

        int second_number = 3; // seco

        // Any space cannot be used be

        int _i_am_also_a_valid_identif

        // An identifier/variable name

        printf("age = %d\n", age);
        printf("firstNumber = %d\n", f
        printf("second_number = %d\n",
        printf("_i_am_also_a_valid_ide
        return 0;
}
```

Answer: (penalty regime: 0 %)

Reset answer

```
#include <stdio.h>

int main()
{
    int age = 2;
    int firstNumber = 2;
    int second_number = 3;
    int _i_am_also_a_valid_i
    printf("age = %d\n",age
    printf("firstNumber = %d
    printf("second_number =
    printf("_i_am_also_a_val
    return 0;
}
```

	Expected
✓	age = 2 firstNumber = 2 second_number = 3 _i_am_also_a_valid_identifier

Passed all tests! ✓

Answer: (penalty regime: 0 %)

Reset answer

```
1
2
3
4
5
6
7   ;
8   d_identifier = 4;
9   ge ); // Fill in the missing
10  %d\n",firstNumber ); // Fil
11  = %d\n",second_number ); //
12  valid_identifier = %d\n",_i_
13
14
```

	Expected
✓	age = 2 firstNumber = 2 second_number = 3 _i_am_also_a_valid_identifier

Passed all tests! ✓

Answer: (penalty regime: 0 %)

Reset answer

```
1
2
3
4
5
6
7
8
9   code
10  l in the missing code
11  Fill in the missing code
12  am_also_a_valid_identifier );
13
14
```

	Expected
✓	age = 2 firstNumber = 2 second_number = 3 _i_am_also_a_valid_identifier

Passed all tests! ✓

Identify and correct the error in the code given below.

Answer: (penalty regime: 0 %)

Reset answer

```
1  #include <stdio.h>
2  int main()
3  {
4      printf("Hello, I am learn
5      return 0;
6  }
```

	Expected
✓	Hello, I am learning C Languag

Passed all tests! ✓

Identify and correct the error in the code given below.

Answer: (penalty regime: 0 %)

Reset answer

```
#include <stdio.h>

int main()
{
    printf("Hello, # is a pre
    return 0;
}
```

	Expected
✓	Hello, # is a preprocessor in

Passed all tests! ✓

Identify and correct the error in the code given below.

Answer: (penalty regime: 0 %)

Reset answer

```
1  #include <stdio.h>
2
3  int main()
4  {
5      printf("Hello, float data
6      return 0;
7  }
```

	Expected
✓	Hello, float data type allocat

Passed all tests! ✓

Flag question

In the program given below, we shall learn how to assign values to int data type from binary, octal, hex and character literals.

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

```
#include <stdio.h>

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 %)

Answer: (penalty regime: 0 %)

```
#include <stdio.h>
int main(){
    int binaryThree = 0b11;
    printf("binaryThree valu
    int octalEight = 010;
    printf("octalEight value
    int hexTen = 0xA;
    printf("hexTen value = %
    int asciiValueOfOne = '1
    printf("asciiValueOfOne
    int asciiValueOfA = 'A';
    printf("asciiValueOfA va

    return 0;
}
```

	Expected	Go
✓	binaryThree value = 3 octalEight value = 8 hexTen value = 10 asciiValueOfOne value = 49 asciiValueOfA value = 65	bir oct hex asc asc

Passed all tests! ✓

Question **2**

Answer: (penalty regime: 0 %)

```
1
2
3
4   %d\n",binaryThree);
5
6  %d\n",octalEight);
7
8  ",hexTen);
9
10 ue = %d\n",asciiValueOfOne);
11
12  = %d\n",asciiValueOfA);
13
14
15
```

	Got	
49 5	binaryThree value = 3 octalEight value = 8 hexTen value = 10 asciiValueOfOne value = 49 asciiValueOfA value = 65	✓

Passed all tests! ✓

Question **2**

Question **2**

Correct

Marked out of 1.00

Flag question

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

Answer: (penalty regime: 0 %)

Reset answer

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int num1 = 15, num2 = 25
6      sum=num1+num2;
7      printf("Given integers a
8      //Write the code to add
9      printf("Sum of 2 given n
10     return 0;
11 }
```

	Expected
✓	Given integers are num1 = 15, Sum of 2 given numbers = 40

Passed all tests! ✓

Question **3**

Incorrect

Marked out of 1.00

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
The final value in num = 4294967295 (i
```

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

Answer: (penalty regime: 0 %)

Question **1**

Correct

Marked out of 1.00

Flag question

Identify and correct the errors in the code given below:

Expected Output:

Given float values are num1 = 5.340000, num2 = 125.789001
The result after dividing in float format = 23.555992
The result after dividing in exponential format = 2.355599e+01

Answer: (penalty regime: 0 %)

Reset answer

```
1  #include <stdio.h>
2
3  int main()
4  {
5      float num1 = 5.34, num2
6      printf("Given float valu
7      result = num2 / num1;
8      printf("The result after
9      printf("The result after
10     return 0;
11 }
```

Reset answer

```
1  .h>
2
3
4
5  = 5.34, num2 = 125.789, resu
6  ven float values are num1 = %
7  um2 / num1;
8  e result after dividing in fl
9  e result after dividing in ex
10
11
```

	Expected
✓	Given float values are num1 = The result after dividing in f The result after dividing in e

Passed all tests! ✓

Question **2**

Correct

Reset answer

```
1
2
3
4
5
6  2 = %f\n", num1, num2);
7
8  rmat = %f\n",result );
9  ial format = %e\n",result );
10
11
```

ɔt

```
ven float values are num1 = 5.340000
e result after dividing in float for
e result after dividing in exponenti
```

Passed all tests! ✓

Question **2**

Correct

given below:

Answer: (penalty regime: 0 %)

Reset answer

```
1  #include <stdio.h>
2
3  int main()
4  {
5      float num1 = 5.345, num2
6      printf("Given float valu
7      result = num1 / num2;
8      printf("Result of divisi
9      return 0;
10 }
```

	Expected
✓	Given float values are num1 = Result of division = 0.431048

Passed all tests! ✓

given below:

Answer: (penalty regime: 0 %)

Reset answer

```
1
2
3
4
5
6   , num2 = %f\n", num1, num2);
7
8   lt);
9
10
```

Got

```
Given float values are num1 = 5.345
Result of division = 0.431048
```

Passed all tests! ✓

given below:

Answer: (penalty regime: 0 %)

Reset answer

```
1
2
3
4
5
6   , num2 = %f\n", num1, num2);
7
8   lt);
9
10
```

1 = 5.345000, num2 = 12.400000 048	✓

Passed all tests! ✓

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>
int main(){
    int num1=7;
    float num2=5.5;
    char c='w';
    printf("Result1 = %d\n",
    printf("Result2 = %d\n",
    printf("Result3 = %d\n",
    printf("Result4 = %d\n",
    printf("Result5 = %d\n",
}
```

	Expected	Got	
✓	Result1 = 1 Result2 = 0 Result3 = 1 Result4 = 1 Result5 = 0	Result1 = 1 Result2 = 0 Result3 = 1 Result4 = 1 Result5 = 0	✓

Passed all tests! ✓

Answer: (penalty regime: 0 %)

```
1
2
3
4
5
6   %d\n",(num1>5));
7   %d\n",((num1+num2)<=10));
8   %d\n",(c==119));
9   %d\n",(c!='p'));
10  %d\n",(c>=10*(num1+num2)));
11
```

	Expected	Got	
✓	Result1 = 1 Result2 = 0 Result3 = 1 Result4 = 1 Result5 = 0	Result1 = 1 Result2 = 0 Result3 = 1 Result4 = 1 Result5 = 0	✓

Passed all tests! ✓

```c
#include <stdio.h>
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 = 17 x = 17 x++ = 17 x = 18 --x = 17 x = 17 x-- = 17 x = 16	+x = 16 -x = -16 x = 16 ++x = 17 x = 17 x++ = 17 x = 18 --x = 17 x = 17 x-- = 17 x = 16	✓

Passed all tests! ✓

```c
clude <stdio.h>
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 = 17 x = 17 x++ = 17 x = 18 --x = 17 x = 17 x-- = 17 x = 16	+x = 16 -x = -16 x = 16 ++x = 17 x = 17 x++ = 17 x = 18 --x = 17 x = 17 x-- = 17 x = 16	✓

Passed all tests! ✓

}

Answer: (penalty regime: 0 %)

```c
#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 =
}
```

	Expected	Got
✓	x = 39 y = 24 z = 63	x = 39 y

Passed all tests! ✓

}

Answer: (penalty regime: 0 %)

```
dio.h>

,y=39,z=45;



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

```

	Got	
24 z = 63	x = 39 y = 24 z = 63	✓

Passed all tests! ✓

Answer: (penalty regime: 0 %)

```c
#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",x);
}
```

	Expected	Got	
✓	x = 20 y = 36 z = 2 x = 6	x = 20 y = 36 z = 2 x = 6	✓

Passed all tests! ✓

Answer: (penalty regime: 0 %)

```
#include <stdio.h>
int main(){
    int m=75,pm=50;
    (m>pm)?printf("Passed C e

}
```

	Expected	Got
✓	Passed C exam.	Passed C exam.

Passed all tests! ✓

Answer: (penalty regime: 0 %)

```
1
2
3
4  .");:printf("Failed C exam.");
5
6
```

	Expected	Got
✓	Passed C exam.	Passed C exam.

Passed all tests! ✓

given numbers using **ternary operator**.

Answer: (penalty regime: 0 %)

Reset answer

```c
#include <stdio.h>

int main()
{
    int num1 = 20, num2 = 25,
    large = (num1>num2)?num1:
    printf("Largest number =
    return 0;
}
```

	Expected	Got
✓	Largest number = 25	Largest nu

Passed all tests! ✓

given numbers using **ternary operator**.

Answer: (penalty regime: 0 %)

Reset answer

```
1
2
3
4
5  arge;
6  m2; // Write the correct code
7  ", large);
8
9
```

	Got	
umber = 25	Largest number = 25	✓

Passed all tests! ✓

20	1000

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>
int main(){
    int l,b,a,p;
    scanf("%d %d",&l,&b);
    p=2*(l+b);
    a=l*b;
    printf("%d\n",p);
    printf("%d",a);
}
```

	Input	Expected	Got	
✓	50 20	140 1000	140 1000	✓

Passed all tests! ✓

8	4

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>
int main(){
    int p,t;
    scanf("%d %d",&p,&t);
    printf("%d\n",p/t);
    printf("%d",p%t);

}
```

	Input	Expected	Got	
✓	60 8	7 4	7 4	✓

Passed all tests! ✓

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>
int main(){
    int w,x,y,p;
    scanf("%d %d %d",&w,&x,&y
    p = (w*x)-(w*y)-100;
    printf("%d",p);
}
```

	Input	Expected	Got	
✓	1000 2 1	900	900	✓

Passed all tests! ✓

1000 2 1	900

Answer: (penalty regime: 0 %)

```
1  nclude <stdio.h>
2  t main(){
3    int w,x,y,p;
4    scanf("%d %d %d",&w,&x,&y);
5    p = (w*x)-(w*y)-100;
6    printf("%d",p);
7
```

	Input	Expected	Got	
✓	1000 2 1	900	900	✓

Passed all tests! ✓

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>
int main(){
    int x,y;
    scanf("%d",&x);
    y = (x%10)+((x/10)%10);
    printf("%d",y);
}
```

	Input	Expected	Got	
✓	87	15	15	✓
✓	54	9	9	✓

Passed all tests! ✓

In the program given below, we shall learn how to assign values to int data type from binary, octal, hex and character literals.

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

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

```c
#include <stdio.h>
int main(){
    int binaryThree = 0b11;
    printf("binaryThree valu
    int octalEight = 010;
    printf("octalEight value
    int hexTen = 0xA;
    printf("hexTen value = %
    int asciiValueOfOne = '1
    printf("asciiValueOfOne
    int asciiValueOfA = 'A';
    printf("asciiValueOfA va
    return 0;



}
```

	Expected	Go
✓	binaryThree value = 3 octalEight value = 8 hexTen value = 10 asciiValueOfOne value = 49 asciiValueOfA value = 65	bir oct hex asc asc

Passed all tests! ✓

```
1
2
3
4   %d\n",binaryThree);
5
6  %d\n",octalEight);
7
8  ",hexTen);
9
10 ue = %d\n",asciiValueOfOne);
11
12  = %d\n",asciiValueOfA);
13
14
15
16
17
18
19
20
21
22
```

	Got	
49 5	binaryThree value = 3 octalEight value = 8 hexTen value = 10 asciiValueOfOne value = 49 asciiValueOfA value = 65	✓

Passed all tests! ✓

Answer: (penalty regime: 0 %)

Reset answer

```c
#include <stdio.h>

int main()
{
    int num1 = 15, num2 = 25
    printf("Given integers a
    sum = num1+num2;
    //Write the code to add
    printf("Sum of 2 given n
    return 0;
}
```

	Expected
✓	Given integers are num1 = 15, Sum of 2 given numbers = 40

Passed all tests! ✓

Answer: (penalty regime: 0 %)

Reset answer

```
1
2
3
4
5
6   , num1, num2);
7
8   e result in the variable sum
9
10
11
```

	Expected
✓	Given integers are num1 = 15, Sum of 2 given numbers = 40

Passed all tests! ✓