GE23131-Programming Using C-2024

Status	Finished
Started	Monday, 23 December 2024, 5:33 PM
Completed	Friday, 25 October 2024, 1:09 PM
Duration	59 days 4 hours

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

167.64

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>
int main()
{ float i,cm,f;
scanf("%f %f",&f,&i);
cm=(((f*12)+i)*2.54);
printf("%0.2f",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 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 %)

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

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

Passed all tests! ✓

GE23131-Programming Using 2024

Status	Finished
Started	Monday, 23 December 2024, 5:33 PM
Completed	Wednesday, 6 November 2024, 12:54 PM
Duration	47 days 4 hours

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

100 110

SAMPLE OUTPUT 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	✓
✓	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.

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,a;
scanf("%d",&n);
(a=n*(n-1)/2);
printf("%d",a);
}
```

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

Passed all tests! ✓

GE23131-Programming Using C-2024

Status	Finished
Started	Friday, 27 December 2024, 3:40 PM
Completed	Friday, 27 December 2024, 3:45 PM
Duration	5 mins 7 secs

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, 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 % 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
+ 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 %)

```
#include <stdio.h>

int main() {
    int num1 = 10, num2 =
    
    printf("Addition Resu
    printf("Subtraction R
    printf("Multiplicatio
    printf("Division Resu
    printf("Remainder = %
    
    return 0;
}
```

	Expected
✓	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.**

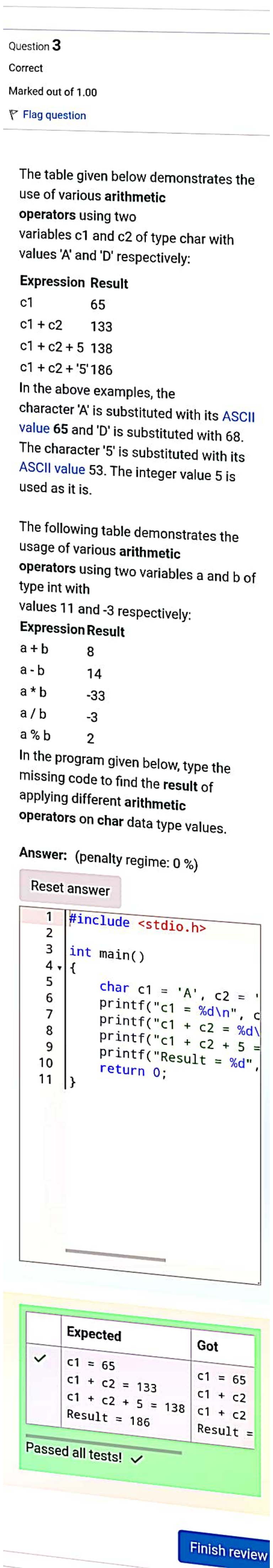
Answer: (penalty regime: 0 %)

Reset answer

```
#include <stdio.h>

int main() {
    float num1 = 12.5, nu
    
    printf("Result of add
    printf("Result of sub
    printf("Result of mul
    printf("Result of div
    
    return 0;
}
```

	Expected
✓	Result of addition = 14.500 Result of subtraction = 10. Result of multiplication = Result of division = 6.2500

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
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

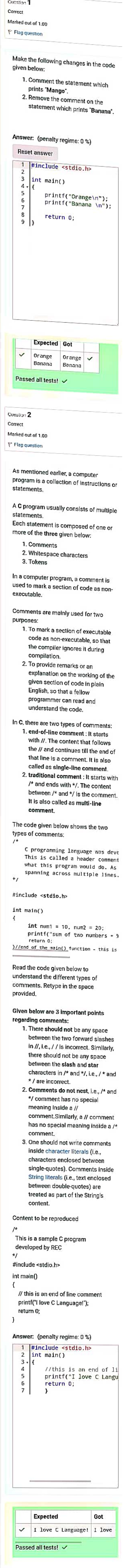
```
#include <stdio.h>

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

	Expected	Got

✓	c1 = 65
	c1 + c2 = 133
	c1 + c2 + 5 = 138
	Result = 186
		c1 = 65
		c1 + c2
		c1 + c2
		Result =

Passed all tests! ✓

Finish review



Question 1

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 \n");

    return 0;
}
```

	Expected	Got	
✓	Orange Banana	Orange Banana	✓

Passed all tests! ✓

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

#include <stdio.h>

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

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.Similarly, a // 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
 developed by REC
*/
#include <stdio.h>

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

Answer: (penalty regime: 0 %)

```
#include <stdio.h>
int main()
{
    //this is an end of li
    printf("I love C Langu
    return 0;
    }
```

	Expected	Got
✓	I love C Language!	I love

Passed all tests! ✓

```
return 0;
}
```

	Expected	Got
✓	One TwoThree	One TwoThree
	Four	Four
	Five	Five

Passed all tests! ✓

Question **4**

Correct

Marked out of 1.00

Flag question

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.
- 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	N lir
Escape Sequence	\a	\b	\t	\v	\r
ASCII value	007	008	009	011	0

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>

int main()
{
   printf("One Two");
   printf("Three\n");
   printf("Four\nFive\n");
   return 0;
}
```

Answer: (penalty regime: 0 %)

```
#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

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 %)

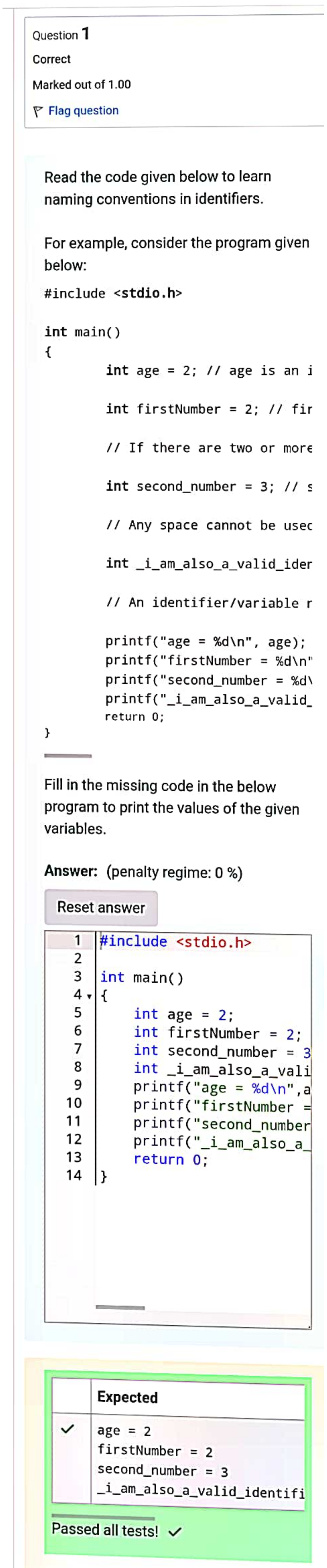
Reset answer

```
#include <stdio.h>

int main()
{
    printf("Dennis Ritchie
    return 0;
}
```

	Expected	Got
✓	Dennis Ritchie Brian Kernighan	Dennis Rit Brian Kern

Passed all tests! ✓

Finish review



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 i

        int firstNumber = 2; // fir

        // If there are two or more

        int second_number = 3; // s

        // Any space cannot be usec

        int _i_am_also_a_valid_ider

        // An identifier/variable r

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

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

Answer: (penalty regime: 0 %)

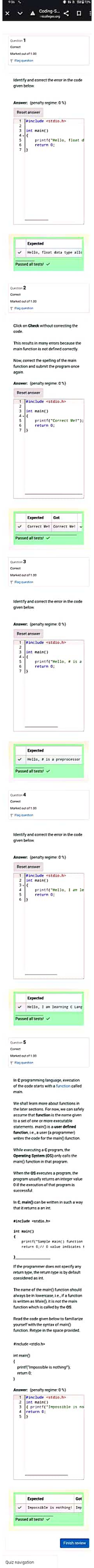
Reset answer

```
#include <stdio.h>

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

	Expected

✓	age = 2
	firstNumber = 2
	second_number = 3
	_i_am_also_a_valid_identifi

Passed all tests! ✓



Question **1**

Correct

Marked out of 1.00

Flag question

Identify and correct the error in the code given below.

Answer: (penalty regime: 0 %)

Reset answer

```
#include <stdio.h>

int main()
{
    printf("Hello, float d
    return 0;
}
```

	Expected
✓	`Hello, float data type allc`

Passed all tests! ✓

Question **2**

Correct

Marked out of 1.00

Flag question

Click on **Check** without correcting the code.

This results in many errors because the main function is not defined correctly.

Now, correct the spelling of the main function and submit the program once again.

Answer: (penalty regime: 0 %)

Reset answer

```
#include <stdio.h>

int main()
{
    printf("Correct Me!");
    return 0;
}
```

	Expected	Got	
✓	`Correct Me!`	`Correct Me!`	✓

Passed all tests! ✓

Question **3**

Correct

Marked out of 1.00

Flag question

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
    return 0;
}
```

	Expected
✓	`Hello, # is a preprocessor`

Passed all tests! ✓

Question **4**

Correct

Marked out of 1.00

Flag question

Identify and correct the error in the code given below.

Answer: (penalty regime: 0 %)

Reset answer

```
#include <stdio.h>
int main()
{
    printf("Hello, I am le
    return 0;
}
```

	Expected
✓	`Hello, I am learning C Lang`

Passed all tests! ✓

Question **5**

Correct

Marked out of 1.00

Flag question

In **C** programming language, execution of the code starts with a function called main.

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 of one or more executable statements. main() is a **user defined function**, i.e., a user (a programmer) writes the code for the 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
    return 0;// 0 value indicates !

}
```

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.

```
#include <stdio.h>

int main()
{
  printf("Impossible is nothing!");
  return 0;
}
```

Answer: (penalty regime: 0 %)

```
#include <stdio.h>
int main()
{ printf("Impossible is no
return 0;
}
```

	Expected	Got
✓	`Impossible is nothing!`	`Imp`

Passed all tests! ✓

Finish review

Quiz navigation

Question **1**

Correct

Marked out of 1.00

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 %)

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

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

Passed all tests! ✓

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

```
#include <stdio.h>

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

	Expected
✓	Given integers are num1 = 1 Sum of 2 given numbers = 40

Passed all tests! ✓