CHAPTER 3

INPUT AND OUTPUT HANDLING

SHORT AND LONG QUESTIONS

Q1. Define input functions.

Ans: Input Functions:

In computer programming input means to feed data into a program through an input device. C language provides the **scanf(), getch(), getche(), getchar()** and **gets()** functions to input data.

Q2. Define output functions.

Ans: Output Functions:

In computer programming, output means to display information on screen or print on printer. C language provides the printf(), putchar() and puts() functions to output information on computer screen.

Q3. What is single character output function? Explain with examples.

Ans: Displaying Single Characters

The Putchar() Function:

This function is sued to output a single character to the screen.

Syntax:

```
Putchar(character);
```

Examples 1:

```
Putchar('A')
```

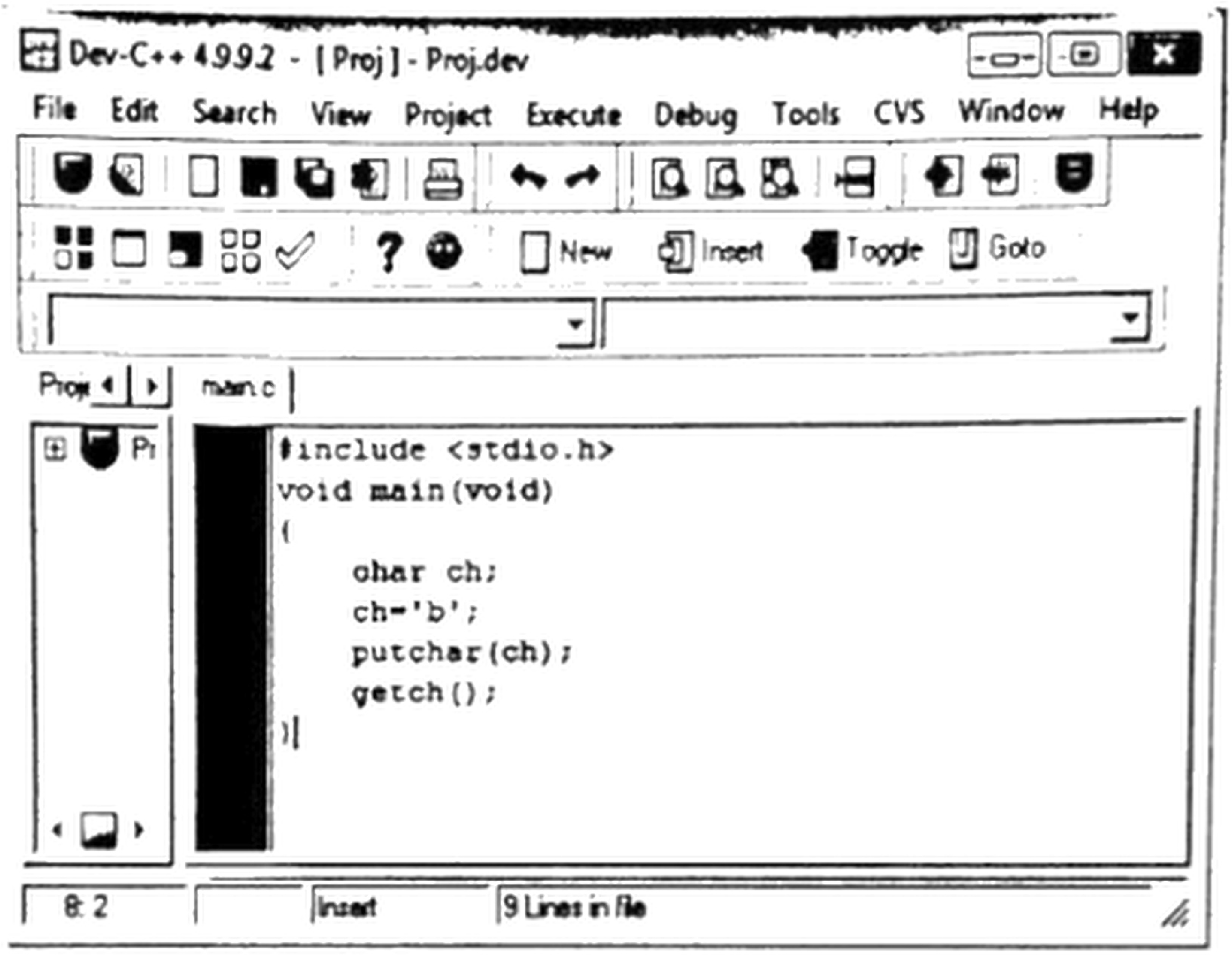
Examples 2:

Char m= ‘P’

Putchar(m)

Program:

The program in Fig prints the character ‘b’ stored in variable ch, on the screen using the putchar() function.


Dev-C++ 4.9.9.2 - [Proj] - Proj.dev
File Edit Search View Project Execute Debug Tools CVS Window Help
New Insert Toggle Goto
Proj
main.c
Pr
8: 2
Insert
9 Lines in file

```
#include <stdio.h>
void main(void)
{
    char ch;
    ch='b';
    putchar(ch);
    getch();
}
```

Program to print a single character

The first statement in the main() function declares the variable ch as type char. Second statement assigns it the letter ‘b’ and the Putchar() function prints it on the screen.

Q4. What is displaying string output function? Explain with example.

Ans: Displaying Strings:

A sequence of characters is known as a string. Strings are very commonly used in programming languages for storing words, name, addresses, sentences etc. String are always enclosed in double quotes as "Computer"

The put() Functions:

The put() function is sued to print a string on the screen.

Syntax:

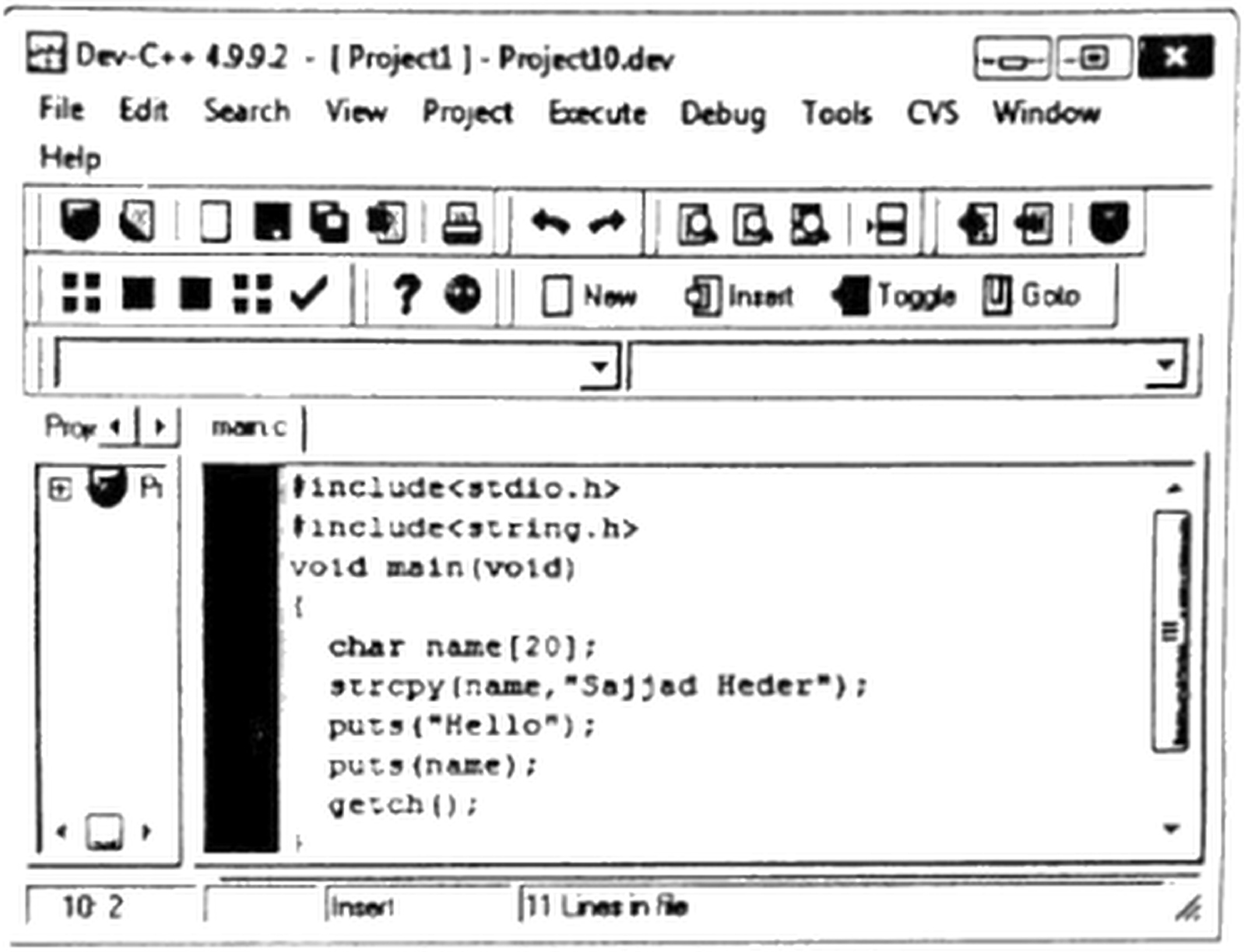
Puts(string);

Examples:

Putchar("AssalamoAllaikum");

Program:

The program in Fig, demonstrates the use of puts() function.


Dev-C++ 4.9.9.2 - [Project1] - Project10.dev
File Edit Search View Project Execute Debug Tools CVS Window Help
New
Insert
Toggle
Goto
Proj
main.c
10: 2
Insert
11 Lines in file

```
#include<stdio.h>
#include<string.h>
void main(void)
{
   char name[20];
   strcpy(name,"Sajjad Heder");
   puts("Hello");
   puts(name);
   getch();
}
```

Use of puts() function in a program

The first statement of the program declares the string variable name and the second statement stores "SajjadHeder" in it. The **strcpy()** function is used for this purpose. The **strcpy()** function stores the string on the right side in the quotation marks to the variable on the left side. The third statement print the text **"Hello"** and the last statement prints the text stored in the variable name.

The **puts()** function is a special-purpose output function. It can only output one string at a time. It is used when the programmer wants to output a single string. It is required to include the **string.h** header file at the beginning of the program when this function is used in a program.

Q5. Explain the displaying of text, variables, constants and expressions using printf() function in C language?

Ans: Displaying of text, variables, constants and expression:

The printf() Function:

The **printf()** function is used to print text and values on the screen in a specified format. It is also called **formatted output function.**

Syntax:

The general syntax of this function is:

Printf(control_string, list of arguments);

The **control_string** consists of text, the formal specifiers and the escape sequence. It is written within double quotes. Text specifies the message that is to be printed along with the values. Format specifier specifies the format according to which a value is to be printed. Escape sequence controls the printing on the output device.

List of arguments consists of a list of variables or arithmetic expression, separated by commas, whose values are to be printed. The values are printed according to the corresponding format specifier. The first format specifier applies to the first argument, the second to the second argument and so on. The arguments in the **printf()** function are optional. When **printf()** function is used to print only a message, then arguments are omitted.

Example:

Following Is an example of **printf()** function.

Printf("\n The value of a is %d and the value of b is %d,", a, b);

Assuming that a has the value of 3 and b as the value of 4, then the following message will be printed by the above statement on a new line.

The value of a is 3 and the value of b is 4.

In the above print statement, \n is an escape sequence and %d is a format specifier. The escape sequence \n is used to start printing on a new line. The format specifier %d is replaced by the values stored in variables a and b in the order in which they are written. The format specifier %d is used for printing values stored in integer variables.

Q6. Briefly explain the scanf() function with the help of program.

Ans: The scanf() Function:

The scanf() function is used to get values into variables from the keyboard during the execution of a program. The value is input into a variable in a specified format.

Its **syntax** is: **scanf(control_string, list of variables);**

The **control_string** in **scanf()** specifies the format specifiers. It is written within double quotes. The **control_string** in **scanf()** function is different from **printf()** function. In **scanf()** function strings cannot be given.

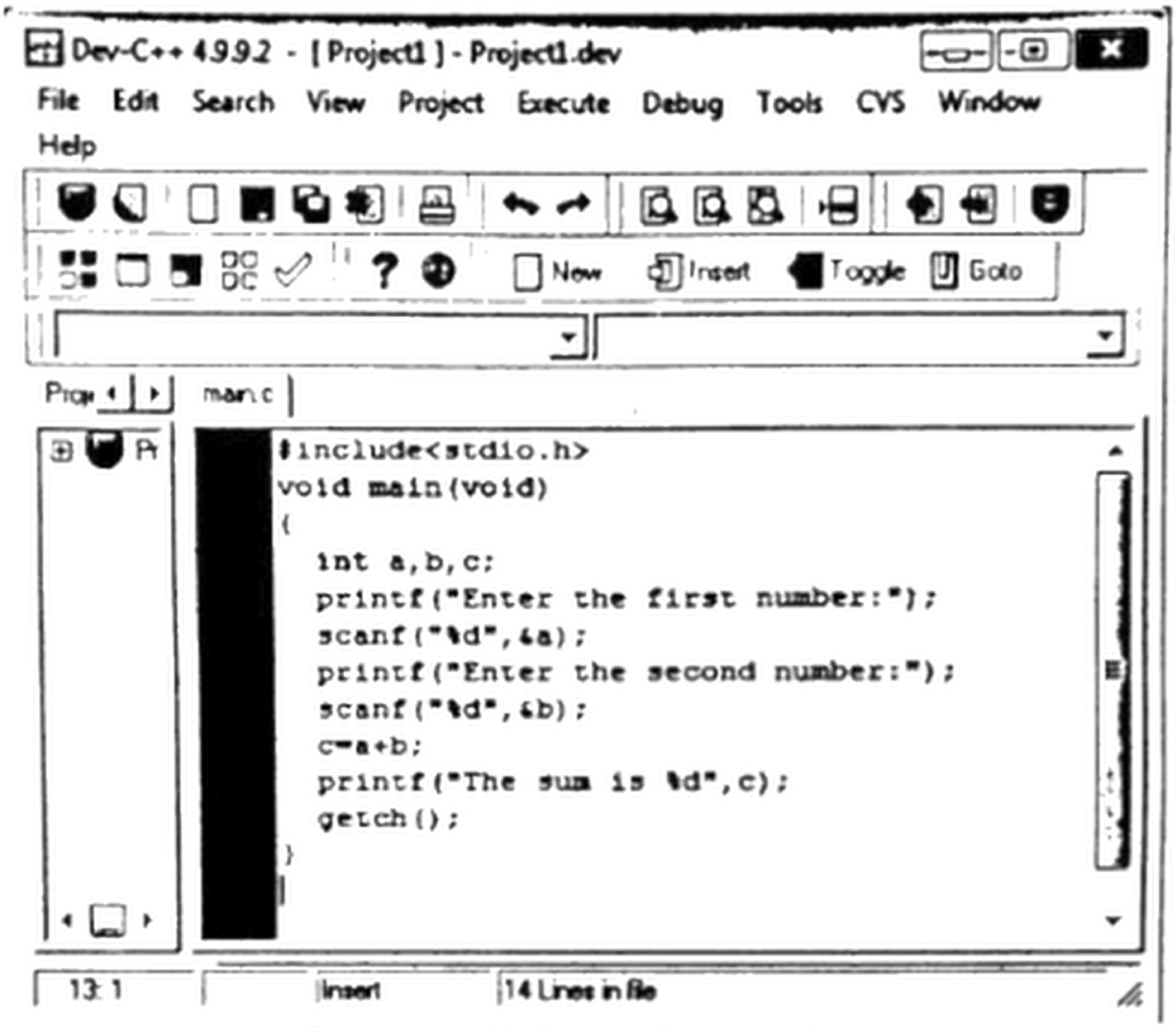
List of variables consists of a list of variables, separated by comms, into which the values are to be entered. In C language, **&** symbol is used in **scanf()** function before the name of the variable to which a value is to be assigned.

For example, to input values into two integer variables, **a** and **b**, the **scanf()** function is written as:

Scanf("%d,%d", &a, &b);

Here is the format specifier %d is used twice in control string for the variables a and b. When this statement is executed, the values may be entered by separating them by comma and the user has to press the Enter key after entering the data.

The program in Fig reads two numbers and print their sum.



```
#include<stdio.h>
void main(void)
{
  int a,b,c;
  printf("Enter the first number:");
  scanf("%d",&a);
  printf("Enter the second number:");
  scanf("%d",&b);
  c=a+b;
  printf("The sum is %d",c);
  getch();
}
```

Program to find sum of two numbers

In the above program, the statement

Inta, b,c;

Is used to declare that **a, b** and **c** are integer variables. Note that the statement ends with a semicolon () as all the C statements do.

The statement

Printf("Enter the first number:");

Prompts the user for the first number

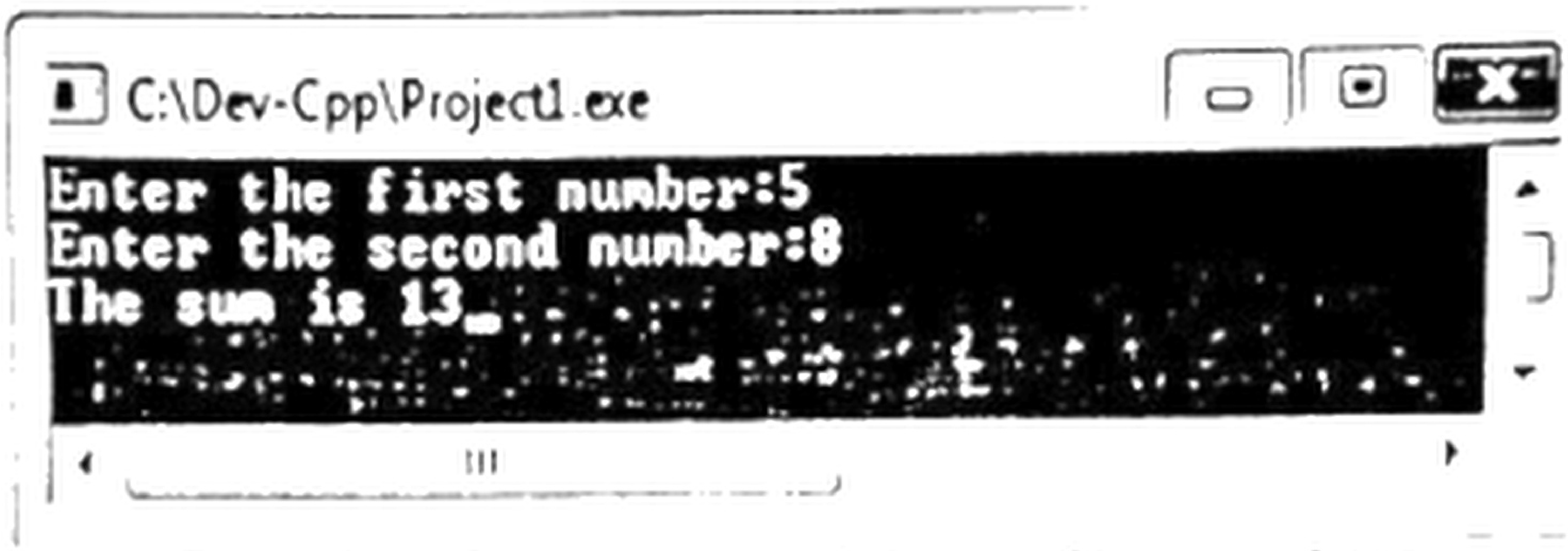
The statement

Scanf("%d, &a);

Reads the value for the variable a and stores it in the computer's memory. In this statement, **%d** format specifier specifies that an integer is to be input.

The next two statements prompt the user to input the second number and store it in variable named b. The expression c=a+b calculates the sum of values stored in variables a and b and stores it in variable c. The last statement print the sum on the screen.

The execution of this program is shown Fig.



```
Enter the first number:5
Enter the second number:8
The sum is 13_
```

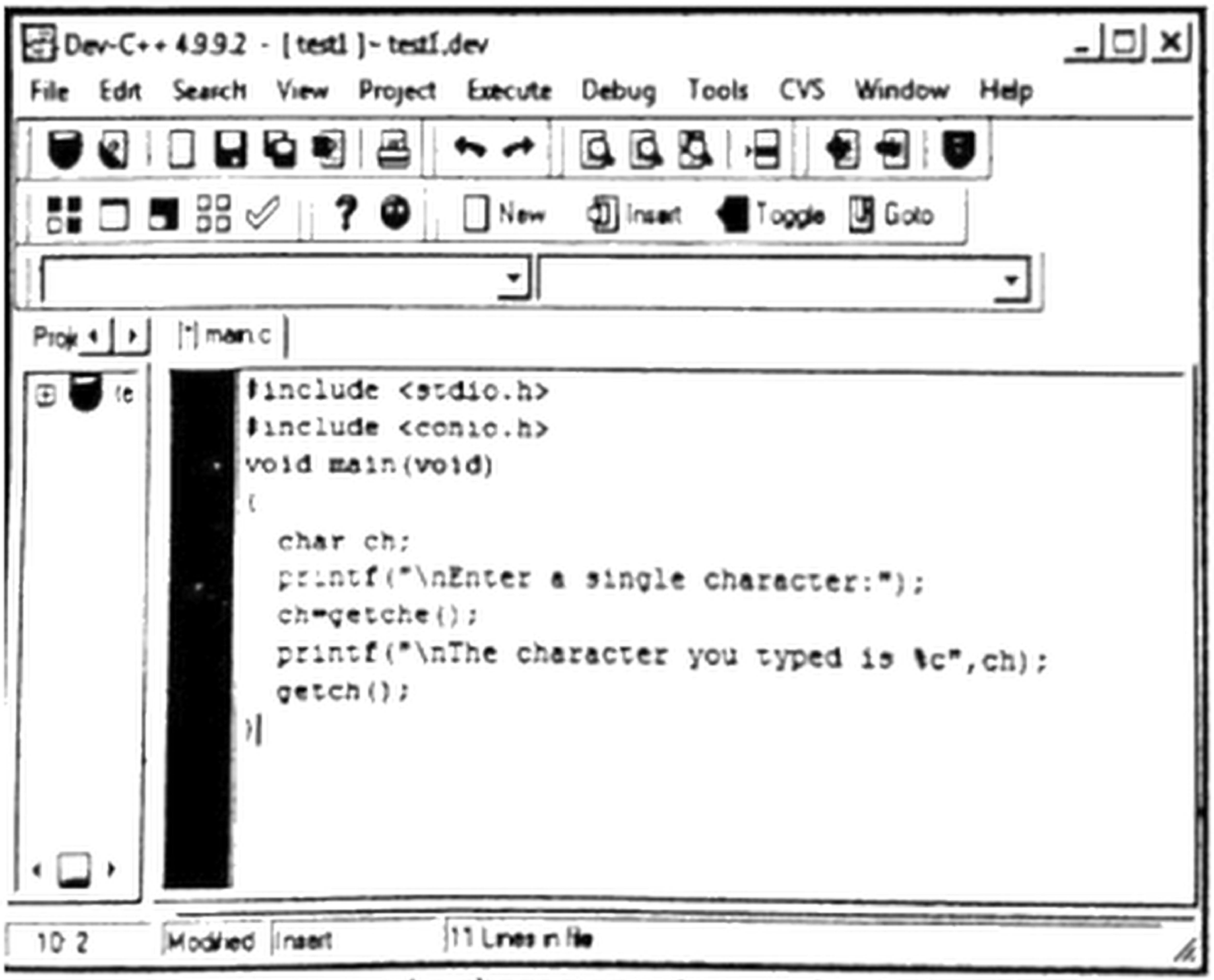
Execution of program to print sum of two numbers

Q7. Describe the getchar() function.

Ans: The getchar() Function:

This input function is used to read a single character from the keyboard. When this function Is used, we have to press the Enter key after typing the character.

The program in Fig, reads a character using **getchar()** function and displays it on the screen.

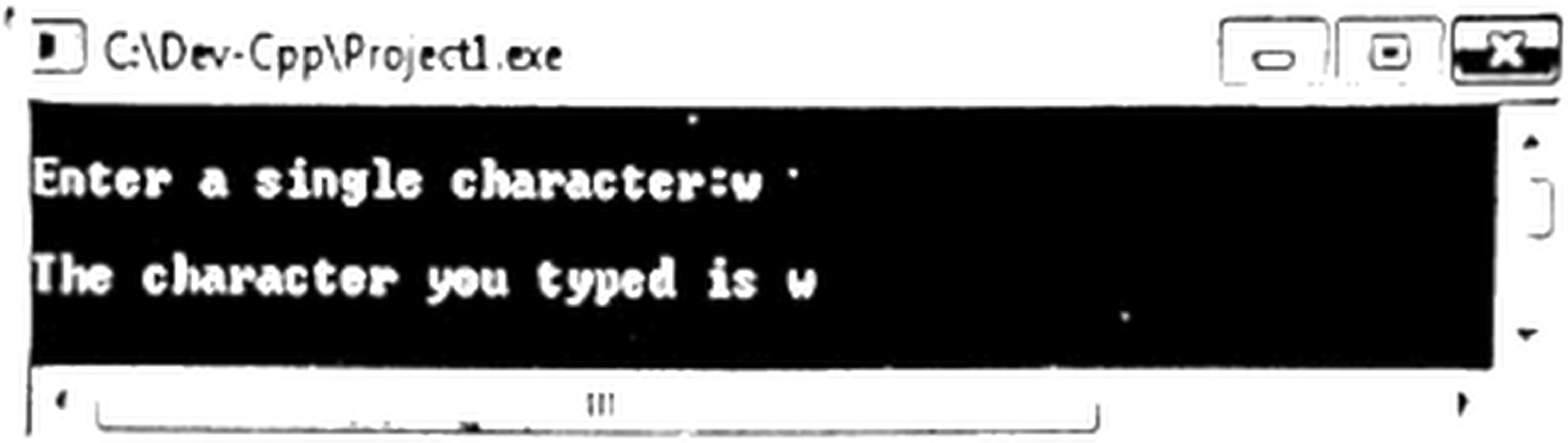


Dev-C++ 4.9.9.2 - [test1] - test1.dev
File Edit Search View Project Execute Debug Tools CVS Window Help
New Insert Toggle Goto
Proje
[*] main.c
10: 2
Modified
Insert
11 Lines in file

```
#include <stdio.h>
#include <conio.h>
void main(void)
{
  char ch;
  printf("\nEnter a single character:");
  ch=getche();
  printf("\nThe character you typed is %c",ch);
  getch();
}
```

Program to read a character using getchar() function

The execution of this program is shown in Fig.



```
Enter a single character:w
The character you typed is w
```

Readingacharacterusinggetchar() function

Q8. What is the statement terminator in C?

OR

Write down the uses of statement terminator in C.

Ans: Statement Terminator:

Semicolon (;) is entered at the end of a statement in C language. It indicates the compiler that the statement ends here. If a statement is not terminated with semicolon, C compiler will give an error message during compilation and the program will not be compiled.

Q9. What are the operators of C Language? List the types of operators.

Ans: operators of C Language:

Expressions consist of constants and variables combined together with operators. An operator is a symbol used to command the computer to perform a certain mathematical or logical operation. Operators are used to operate on data and variables.

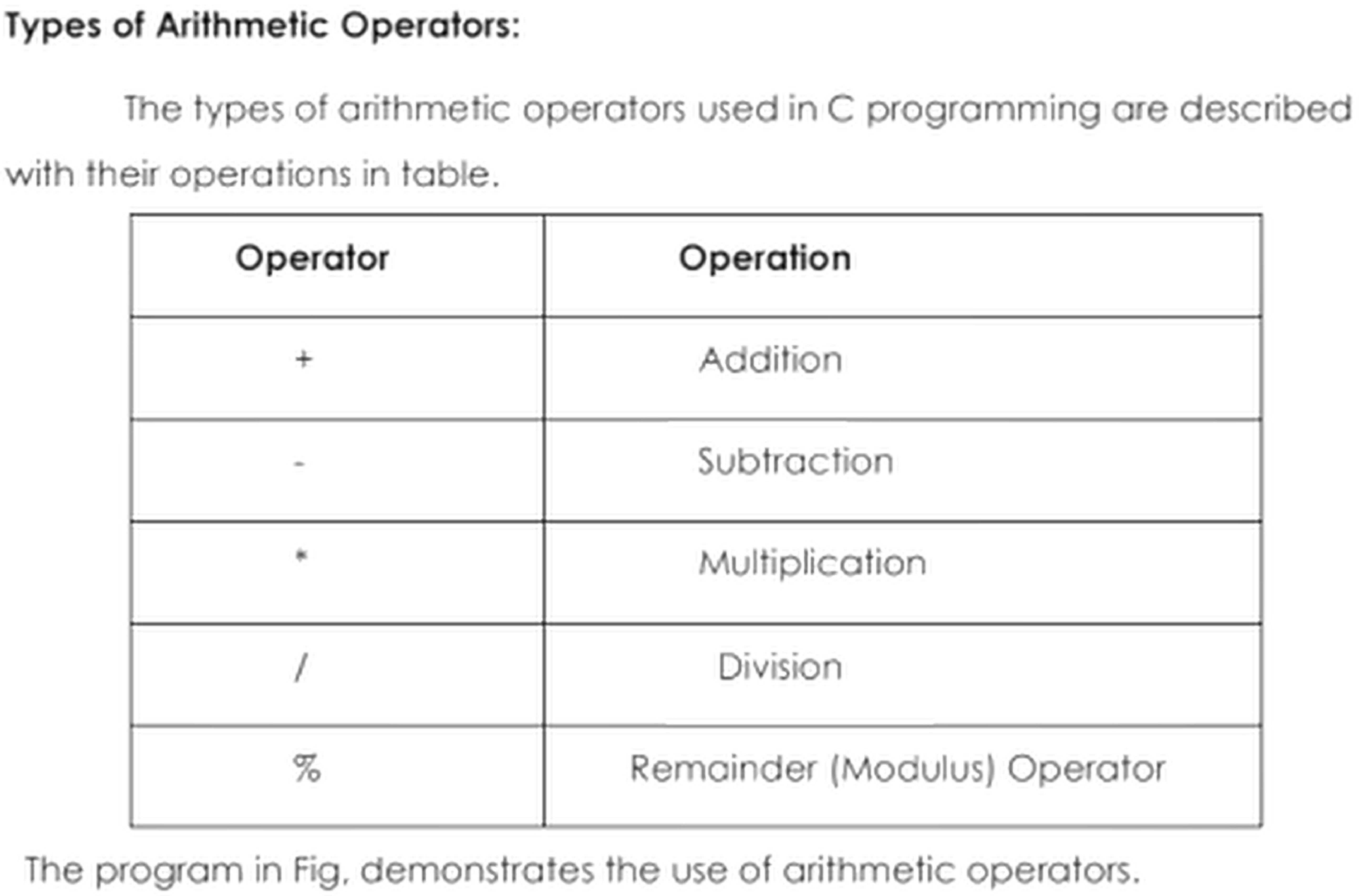
The following types of operators are commonly used in C language.

- Arithmetic operators
- Assignment operators
- Relational operators
- Logical operators
- Increment and decrement operators

Q10. Define arithmetic operators and its types. Also explain how to use them in a program.

Ans: Arithmetic Operators:

Arithmetic operators are used to perform arithmetic operations that include addition, subtraction, multiplication, division and also to find the remainder obtained when an integer is divided by another integer.

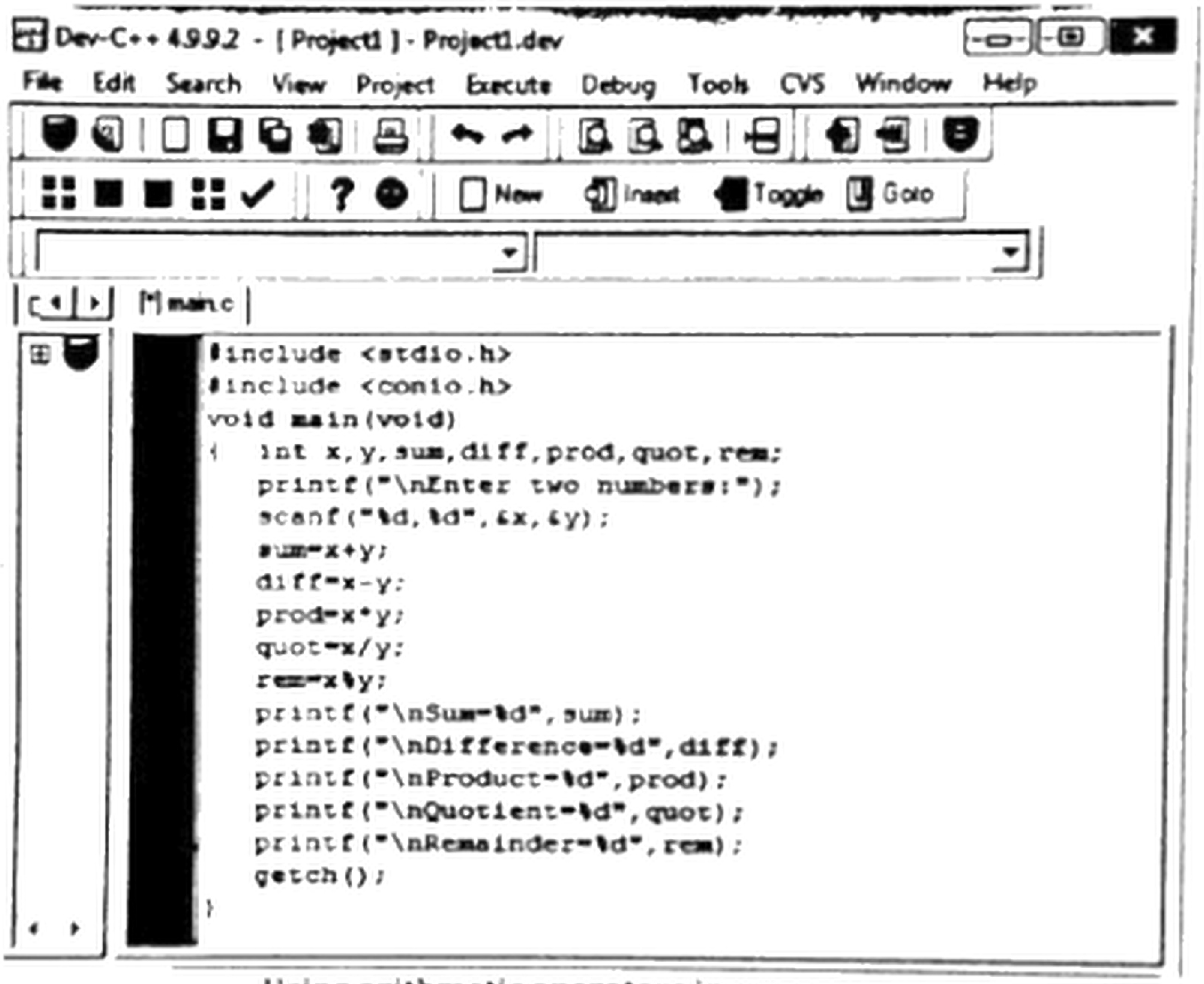


Types of Arithmetic Operators:

The types of arithmetic operators used in C programming are described with their operations in table.

Operator	Operation
+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Remainder (Modulus) Operator

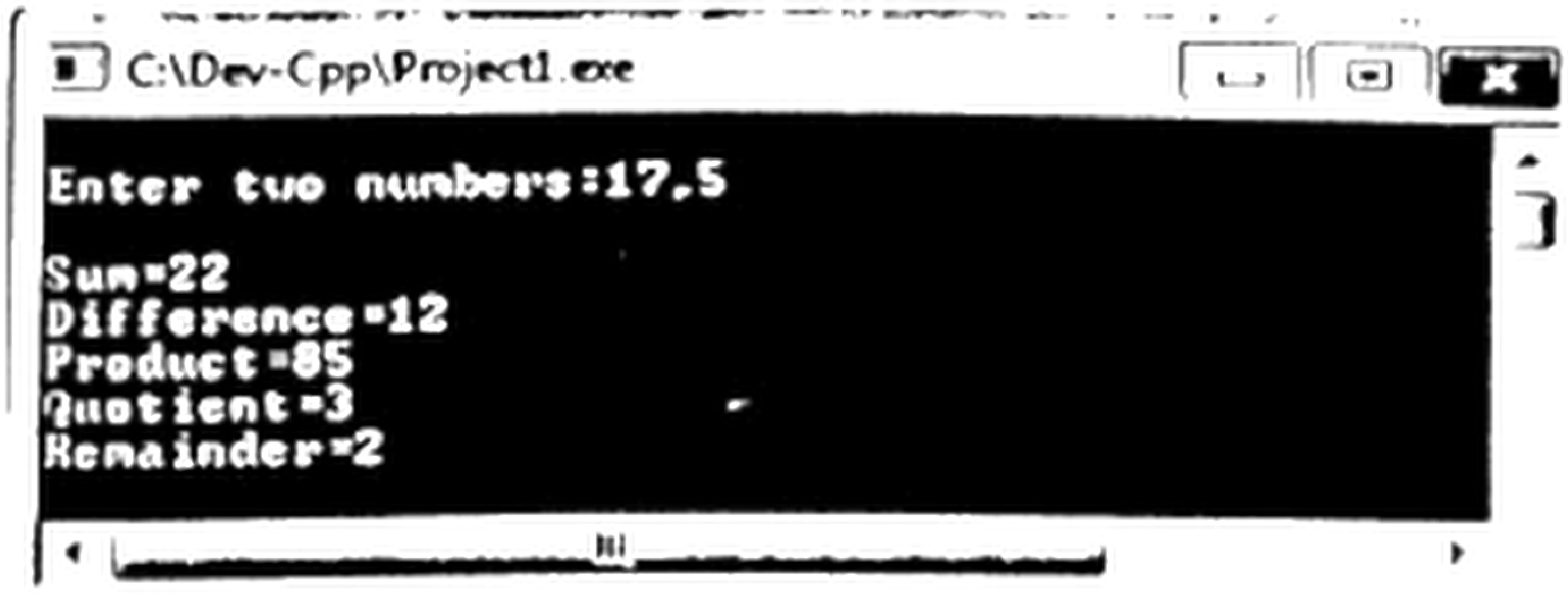
The program in Fig, demonstrates the use of arithmetic operators.


Dev-C++ 4.9.9.2 - [Project1] - Project1.dev
File Edit Search View Project Execute Debug Tools CVS Window Help
New Insert Toggle Goto
main.c

```
#include <stdio.h>
#include <conio.h>
void main(void)
{   int x,y,sum,diff,prod,quot,rem;
    printf("\nEnter two numbers:");
    scanf("%d,%d",&x,&y);
    sum=x+y;
    diff=x-y;
    prod=x*y;
    quot=x/y;
    rem=x%y;
    printf("\nSum=%d",sum);
    printf("\nDifference=%d",diff);
    printf("\nProduct=%d",prod);
    printf("\nQuotient=%d",quot);
    printf("\nRemainder=%d",rem);
    getch();
}
```

Using arithmetic operators in a program

In this program when x/y is performed it will give an integer result because the fractional part is truncated when two operands are of type integer. Moreover, the remainder operator will give the remainder after dividing x by y when x%y is performed. The output of the program is shown in Fig.



```
Enter two numbers:17,5

Sum=22
Difference=12
Product=85
Quotient=3
Remainder=2
```

Use of arithmetic operators to perform calculations

Q11. Explain the use of increment and decrement operators in C with the help of examples.

Ans: Increment and Decrement Operators:

Increment operator is ++ and decrement operator Is --. These are defined in table.

Operator	Definition
++	Increment by 1
--	Decrement by 1

Examples:

++n and n++ are both equivalent to **n=n + 1 (or n+ 1)**

--n and n-- are both equivalent to **n=n – 1 (or n-=1)**

When increment or decrement operator is written before the variable, it is known as prefix and when it is written after the variable, it is known as postfix.

In certain situations, ++n and n++ have different effect. This is because ++n increments n before using its value whereas n++ increments n after it Is used.

As an example, suppose n has the value 3

The statement: **a= ++n;**

Will first increment **n** and then assigns the value 4 to a.

But the statement **a= n++;**

Will first assign the value 3 to a and then increments n to 4. In both cases n has the value 4.

The same rule applies to **–n** and **n—**as well

Q12. Differentiate between assignment operator and equal to operator?

Ans: Difference between Assignment Operator and Equal to Operator:

The assignment operator (=) is used to assign a value to a variable whereas the equal to operator (==) is used to compare two values of same data type.

For example:

A=5;

C=b;

Z=x+y;

In the above statements, variable a is assigned the value 5, c is assigned the value stored in variable b and z is assigned the sum of values stored in variable x and y.

The relational operator (= =) is used to build a condition based on which computer takes some action.

For example:

A= = 1

C= = a+ b

In the first condition, if the value of a is equal to 1 then the condition is true otherwise it is false. In the second condition, the equal to operator is used to check whether the value of c is equal to the sum of a and b. If it is equal then the condition is true otherwise, it is false.

Q13. Differentiate between Unary and Binary Operators.

Ans: Difference between Unary and Binary Operators:

The operators that work with a single operand are known as unary operators whereas operators that work with two operands are known as binary operators. Unary operators are -, ++, -- and the logical operator !(NOT). Binary operators are -, +, *, \, % and logical operators && (AND) and II (OR).

Some examples of unary operators are:

A= -b;

K++;

--x;

Some examples of binary operators are

A= b + c;

Z= x*y;

K=d%e;

Q14. Define and explain the order of precedence of operators.

Ans: Order of Precedence of Operators

Order of precedence of operators is the rule that specifies the order in which operations are to be performed in an expression. The order of precedence is similar to that used in algebraic formulas.

The order of precedence of operators is shown in table. The operator that has the highest precedence is written at the top and the one with the lowest precedence is written at the bottom.

Precedence	**Operator**	**Description**

1.	++, --	Increment and Decrement (Prefix or Postfix)
2.	*, /, %	Multiplication, Division and Remainder
3.	+, -	Addition and Subtraction
4.	<, <=, >, >=	Relational Operators
5.	= =, !=	Equal to and Not Equal to
6.	!	Logical NOT
7.	&&	Logical AND
8.	||	Logical OR
9.	=, *=, /=, +=, -=	Assignment Operators

As shown int able, the increment/decrement operators are performed first, then arithmetic operators, then relational operators, after that logical operators and the end assignment operators.

If an expression contains two or more operators of the same precedence then the operations are performed from left to right in the order in which they appear.

To illustrate this, consider the expression

7* 10 – 5 % 3 * 4 + 9 / 3

The leftmost multiplication is performed first, giving the intermediate result.

70 – 5 % 3 * 4 + 9 / 3

Multiplication is performed next, giving

70 – 8 + 9 / 3

The last high-priority operation , division is performed next, It gives

70 – 8 + 3

Next the low-priority operations are performed in the order in which they occur from left to right. The subtraction Is thus performed first giving.

62 + 3

And then addition is carried out, giving the final result 65

The standard order of evaluation can be modified by using brackets to enclose part of an expression. The part of the expression within brackets is first evaluated in the standard manner and then the result is combined to evaluate the complete expression.

For example in the expression **a/ (b + c)**

B+c will be perfumed first and then a will be divided by the sum of b and c.

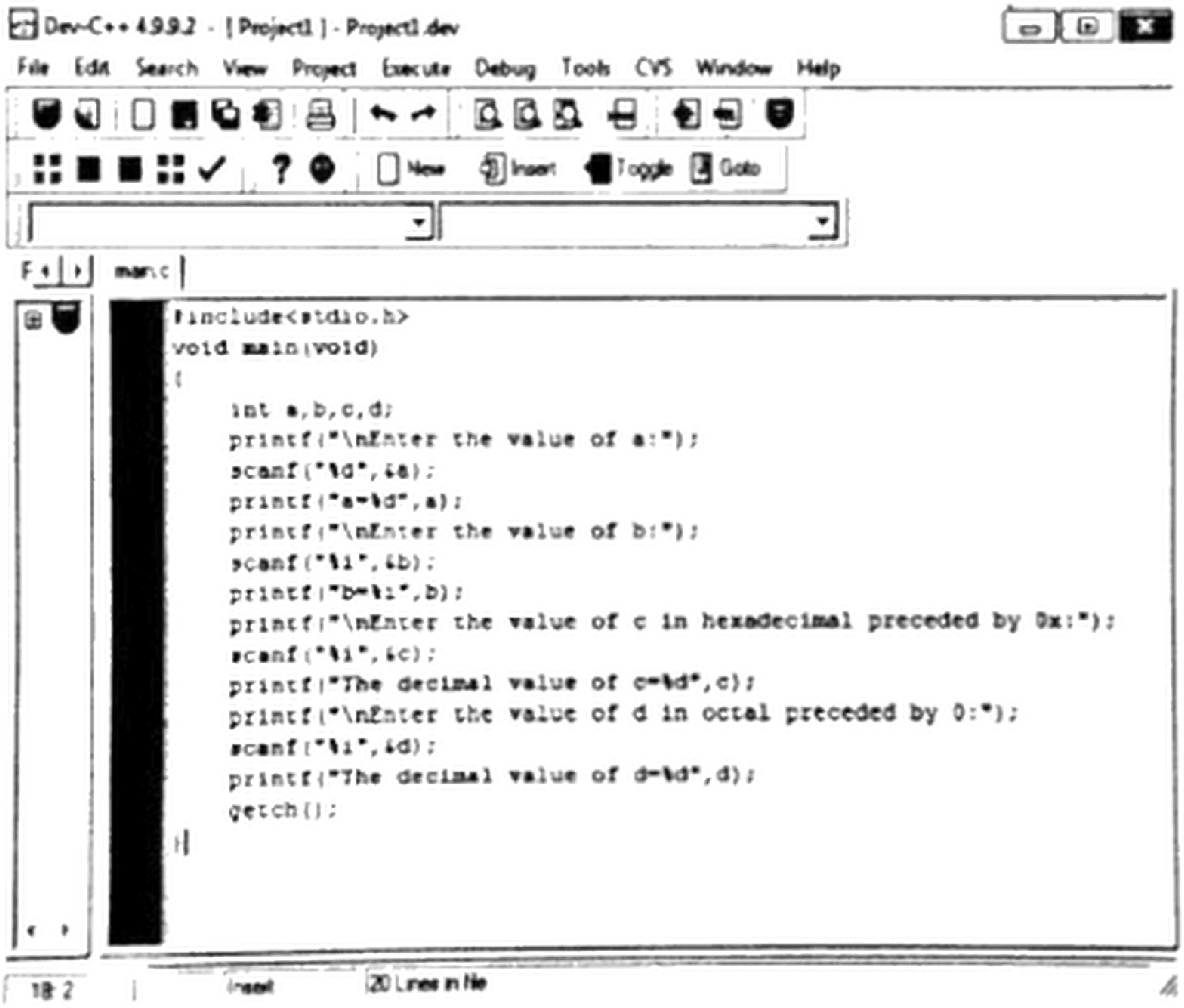
If the brackets are nested, that is, if one set of brackets is contained within another, the computations in the innermost brackets are performed first.

Q15. Why Integer Format Specifier is used? Explain with examples.

Ans: Integer Format Specifier (%d, %ld and %i):

The format specifier %d is used to read or print a decimal integer and the format specifier %id is used with long integers.

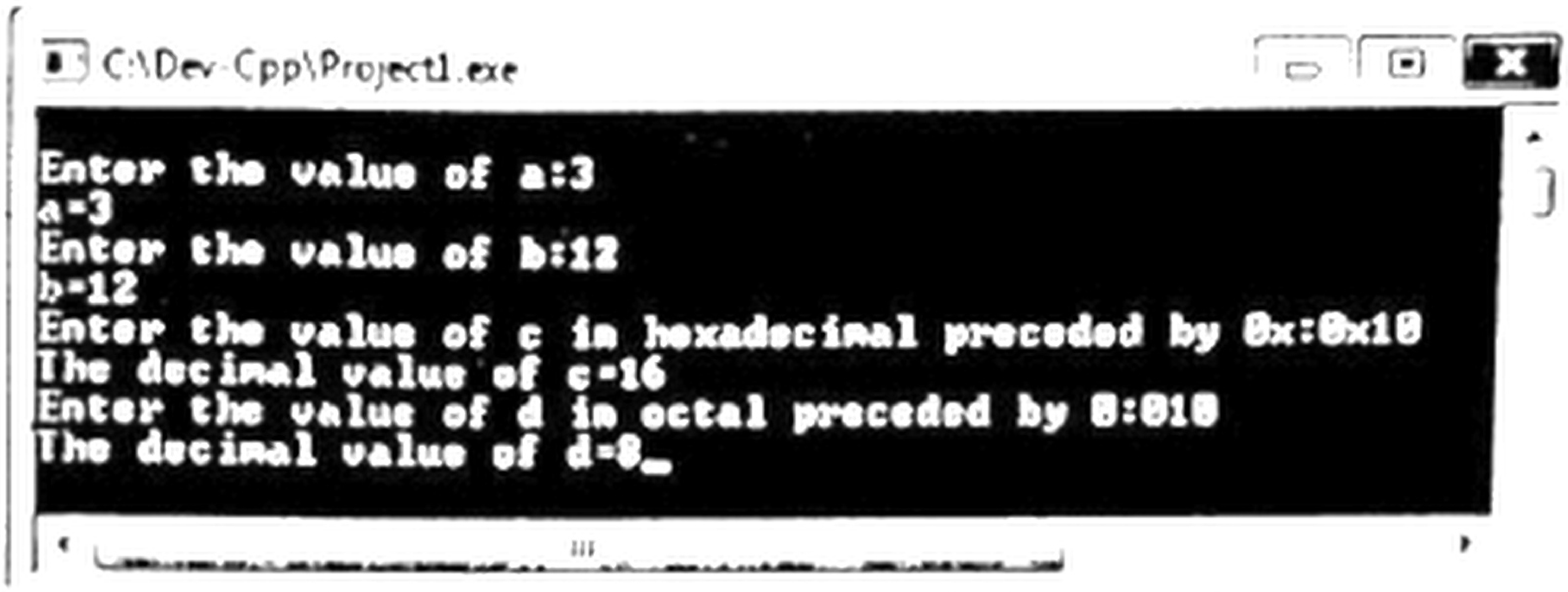
The program in Fig, demonstrates the use of integer format specifiers.


Dev-C++ 4.9.9.2 - [Project1] - Project1.dev
File Edit Search View Project Execute Debug Tools CVS Window Help
New Insert Toggle Goto
main.c
18: 2
Insert
20 Lines in file

```
#include<stdio.h>
void main(void)
{
     int a,b,c,d;
     printf("\nEnter the value of a:");
     scanf("%d",&a);
     printf("a=%d",a);
     printf("\nEnter the value of b:");
     scanf("%i",&b);
     printf("b=%i",b);
     printf("\nEnter the value of c in hexadecimal preceded by 0x:");
     scanf("%i",&c);
     printf("The decimal value of c=%d",c);
     printf("\nEnter the value of d in octal preceded by 0:");
     scanf("%i",&d);
     printf("The decimal value of d=%d",d);
     getch();
}
```

Program that uses integer format specifiers

The format specifier %d and %i are same when they are used with the printf() function but different when used with scanf() functions. For printf() function, both %d and %i are used for decimal integer as shown in program.



```
Enter the value of a:3
a=3
Enter the value of b:12
b=12
Enter the value of c in hexadecimal preceded by 0x:0x10
The decimal value of c=16
Enter the value of d in octal preceded by 0:010
The decimal value of d=8
```

The execution of the program is shown in Fig

Q16. Why Floating-point Format Specifiers is used? Explain with examples.

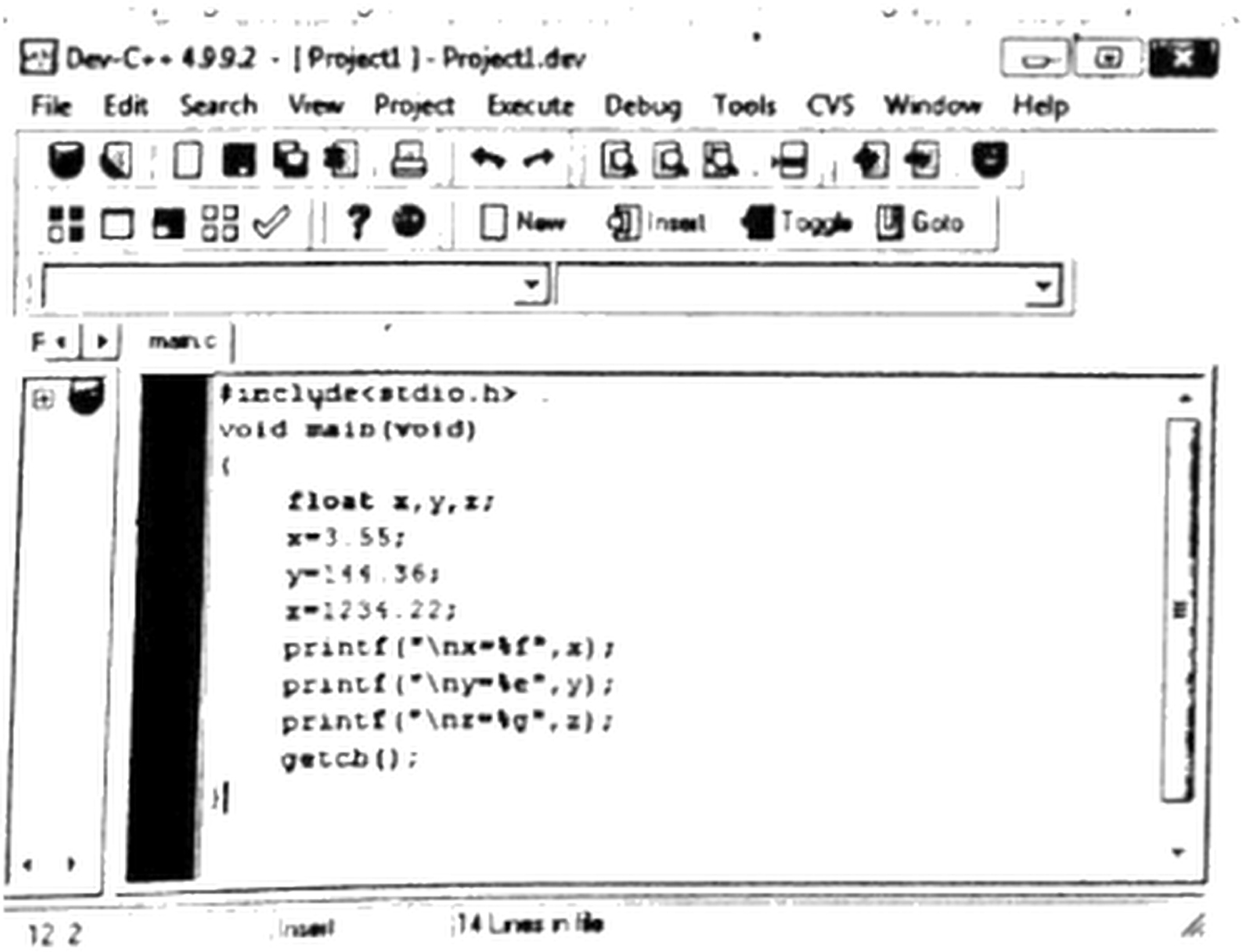
Ans: Floating-point Format Specifiers (%f, %e and %g)

The format specifier %f is used to read and print floating-point numbers in decimal notation with a precision of 6 digits after the decimal point.

The format specifier %e is used to read and print floating-point numbers in exponential notation.

The formal specifier %g is used to print floating-point numbers in decimal or exponential notation whichever is shorter.

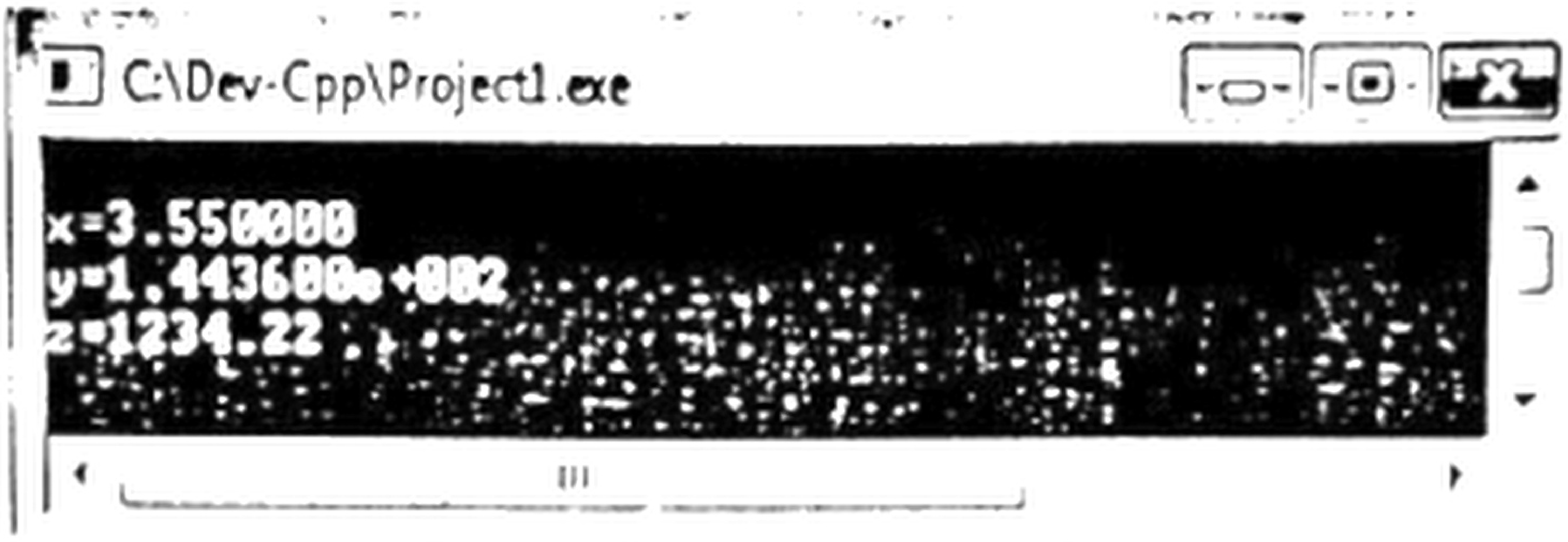
The program in Fig. demonstrates the use of floating-point format specifiers.


Dev-C++ 4.9.9.2 - [Project1] - Project1.dev
File Edit Search View Project Execute Debug Tools CVS Window Help
New Insert Toggle Goto
main.c
12 2
Insert
14 Lines in file

```
#include<stdio.h>
void main(void)
{
    float x,y,z;
    x=3.55;
    y=144.36;
    z=1234.22;
    printf("\nx=%f",x);
    printf("\ny=%e",y);
    printf("\nz=%g",z);
    getch();
}
```

Usingfloating - pointformatspecifiersinaprogram

The execution of the program is shown in Fig.



```
x=3.550000
y=1.443600e+002
z=1234.22
```

Printingfloating - pointnumbers

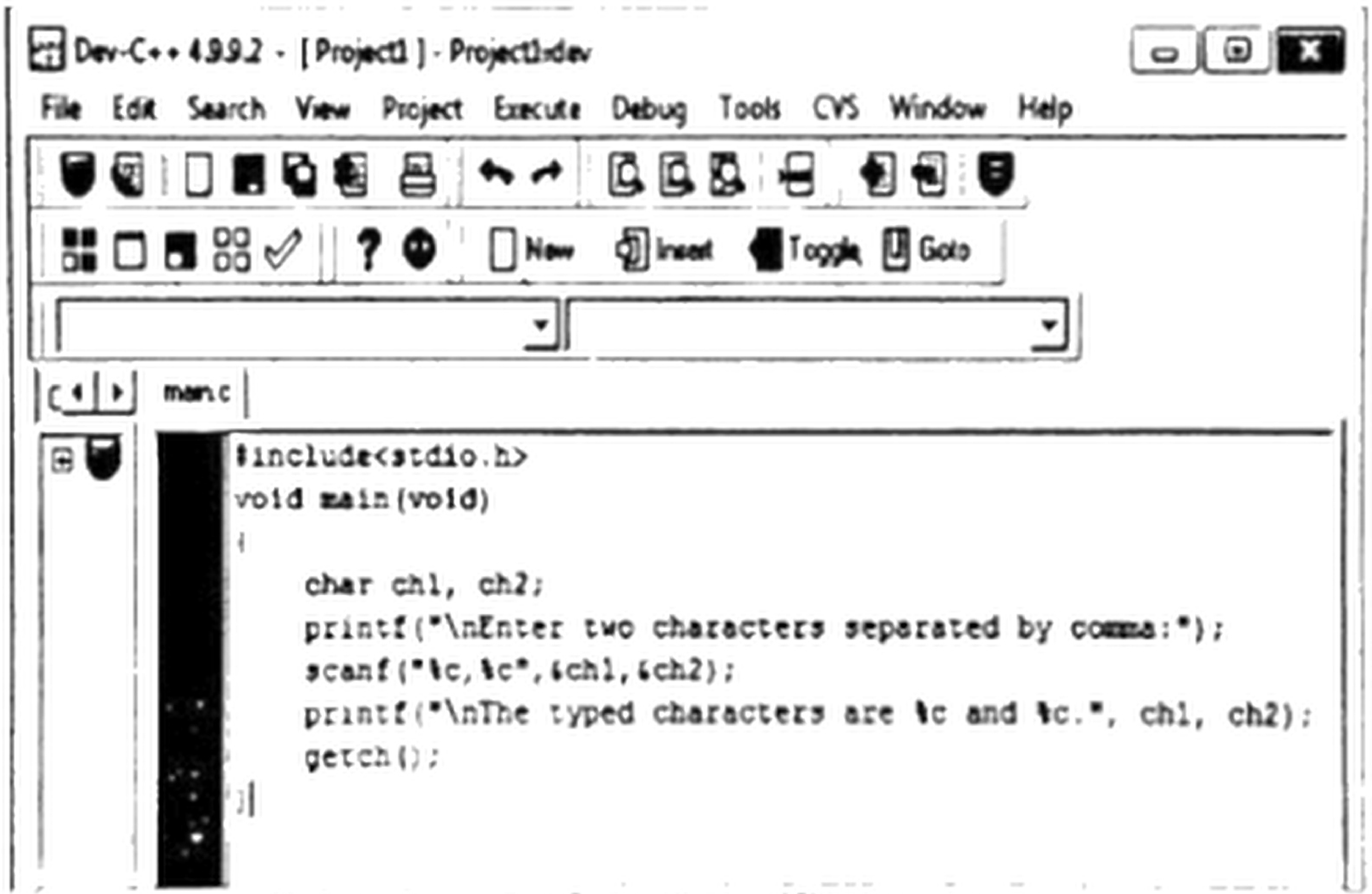
The value of variable z is printed as a floating point number because it is shorter than exponential notation.

Q17. Why Character Formal Specifier is used? Explain with examples.

Ans: The Character Format Specifier (%c):

The character formal specifier, % is used to read or print a single character.

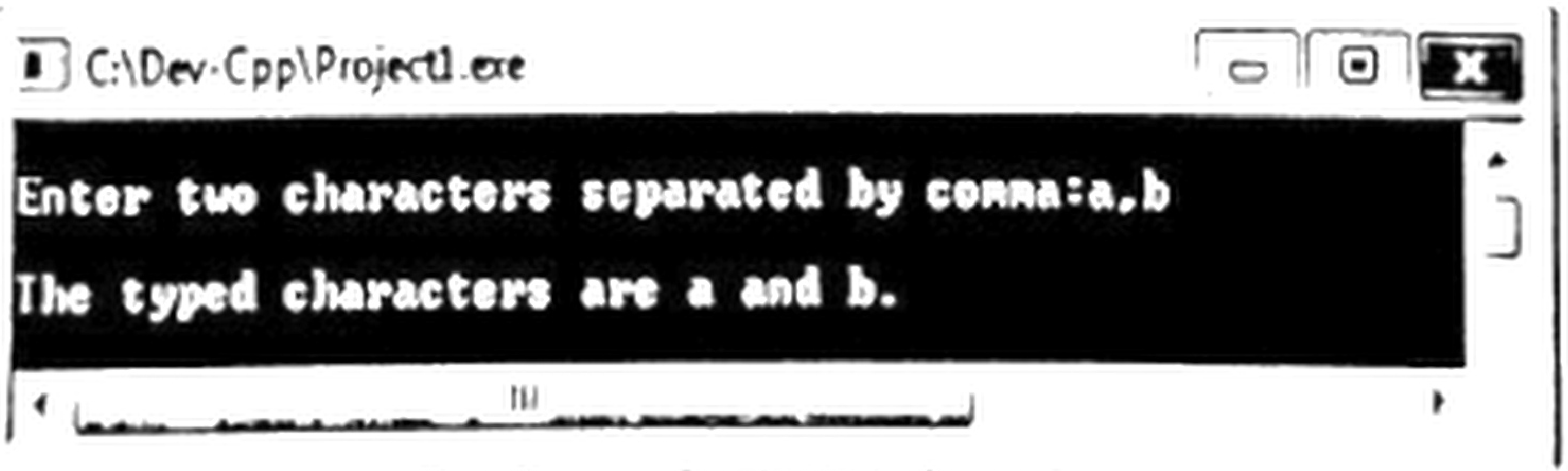
The program in Fig, demonstrates the use of % c format specifier.


Dev-C++ 4.9.9.2 - [Project1] - Project1.dev
File Edit Search View Project Execute Debug Tools CVS Window Help
New Insert Toggle Goto
main.c

```
#include<stdio.h>
void main(void)
{
    char ch1, ch2;
    printf("\nEnter two characters separated by comma:");
    scanf("%c,%c",&ch1,&ch2);
    printf("\nThe typed characters are %c and %c.", ch1, ch2);
    getch();
}
```

Using character format specifier in a program

The execution of the program is shown in Fig.



```
Enter two characters separated by comma:a,b
The typed characters are a and b.
```

Reading and printing characters

KEY POINTS

- The **printf()** functions is used to print text and values on the screen in a specified format.
- The **Putchar()** function is used to output a single character to the screen.
- The **Puts()** function is used to print a string on the screen.
- The **scanf()** function is used to get values into variables from the keyboard during execution of a program
- The **getche()** and getch() functions are used to read a single character from the keyboards, the instant it is typed without waiting for the Enter to be pressed. The getche() function displays the typed character on the screen whereas getch() function does not display it.
- The **getchar()** function is used to read a single character from the keyboard.
- The **gets()** function is used to read a character string from the keyboard.
- A **semicolon** is entered at the end of each C statement to terminate it.
- A **format specifier** tells about the data type, field-width and the format according to which a value is to be printed or read from an input device.
- **Escape sequence** is a combination of backslash (\) and a code character to control printing of data on the screen.
- **Assignment operator** is used to assign a value to a variable.
- **Relational operator** is used to compare two values of the same type. It is used in an expression when a decision is to be based on a condition in a program.
- **Logical operator** is used for building compound condition.

- The **order of precedence** of operators is the rule that specifies the order in which operations are to be performed in an expression.