EXERCISE

Q1. Select the best answers for the following MCQs.

i. **Which function is used to output a single character on the screen?**

A. Printf()
B. Putchar()
C. Puts()
D. Getche()

ii. **Which function reads a single character the instant it is typed, without waiting for the Enter key to be pressed and displays it on the screen?**

A. Scanf()
B. Gets()
C. Getch()
D. Getche()

iii. **Which character terminates a C statement?**

A. Colon
B. Semicolon
C. Period
D. Comma

iv. **Which format specifier is used to print or read a floating-point value in decimal notation?**

A. %d
B. %g
C. %f

D. %e

v. Which escape sequence is used to move cursor to the beginning of current line?

A. \n

B. \r

C. \n

D. \b

vi. Which of the following is an arithmetic operator?

A. %

B. <=

C. &&

D. +=

vii. Which of the following is a logical operator?

A. %

B. <=

C. &&

D. +=

viii. Which statement is equivalent to "k= k + a;" ?

A. K+=a

B. K=+a;

C. K=++a;

D. K=a++;

ix. Which of the following is an increment operator?

A. +

B. +=

C. ++

D. =+

x. Which of the following operator has the highest precedence?

A. &&

B. <=

C. =

D. *

Answers

i. B	ii. D	iii. B	iv.C	v. B
vi.A	vii. C	viii. A	ix.C	x. D

SHORT QUESTIONS

Q2. Give short answers to the following questions.

i. Why format specifier is used? Explain with examples.

Ans: Format Specifiers:

A format specifier is a computer code that 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. A list of commonly used format specifiers is given below

%d decimal integer

%i integer

%ld	long decimal integer
%f	floating-point (decimal notation)
%g	floating-point (exponential notation)
%e	floating point (%f or %g whichever is shorter)
%c	single character
%s	string

ii. Why escape sequence is used? Explain with examples.

Ans: Escape Sequence:

The special characters used in C language to control printing on the output device are called escape sequences. These characters are not printed. These are used inside the control string.

iii. Why escape sequence is used? Explain with examples.

Ans: Escape sequence:

The special characters used in C language to control printing on the output device are called escape sequences. These characters are not printed. These are used inside the control string.

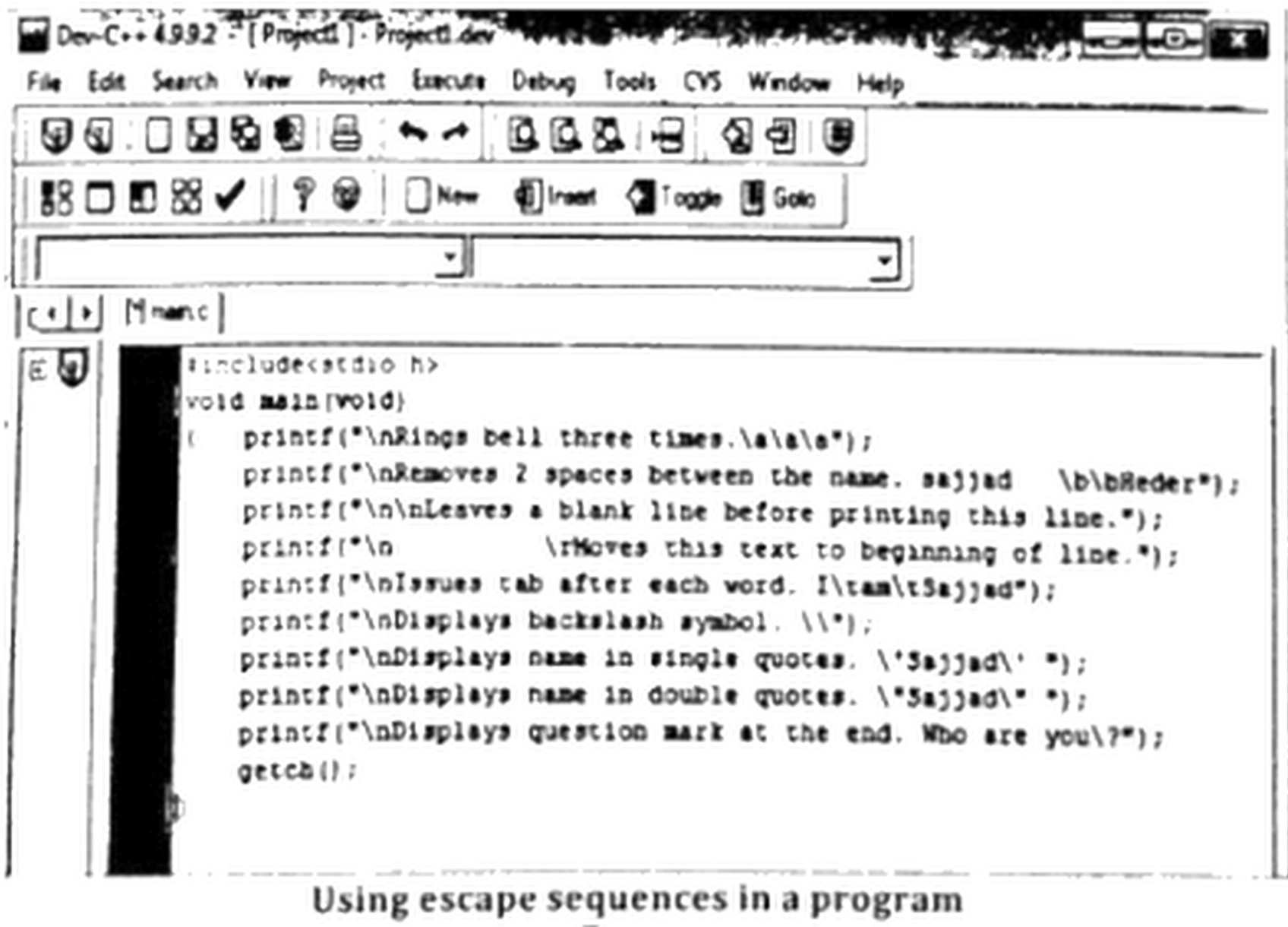
An escape sequence is a combination of a backslash (\) and a code character. The backslash is called the control character. A list of commonly used escape sequence is given in table with their meanings.

Escape sequences in C language:

Escape Sequence	Meaning
\a	Produces alert (bell) sound

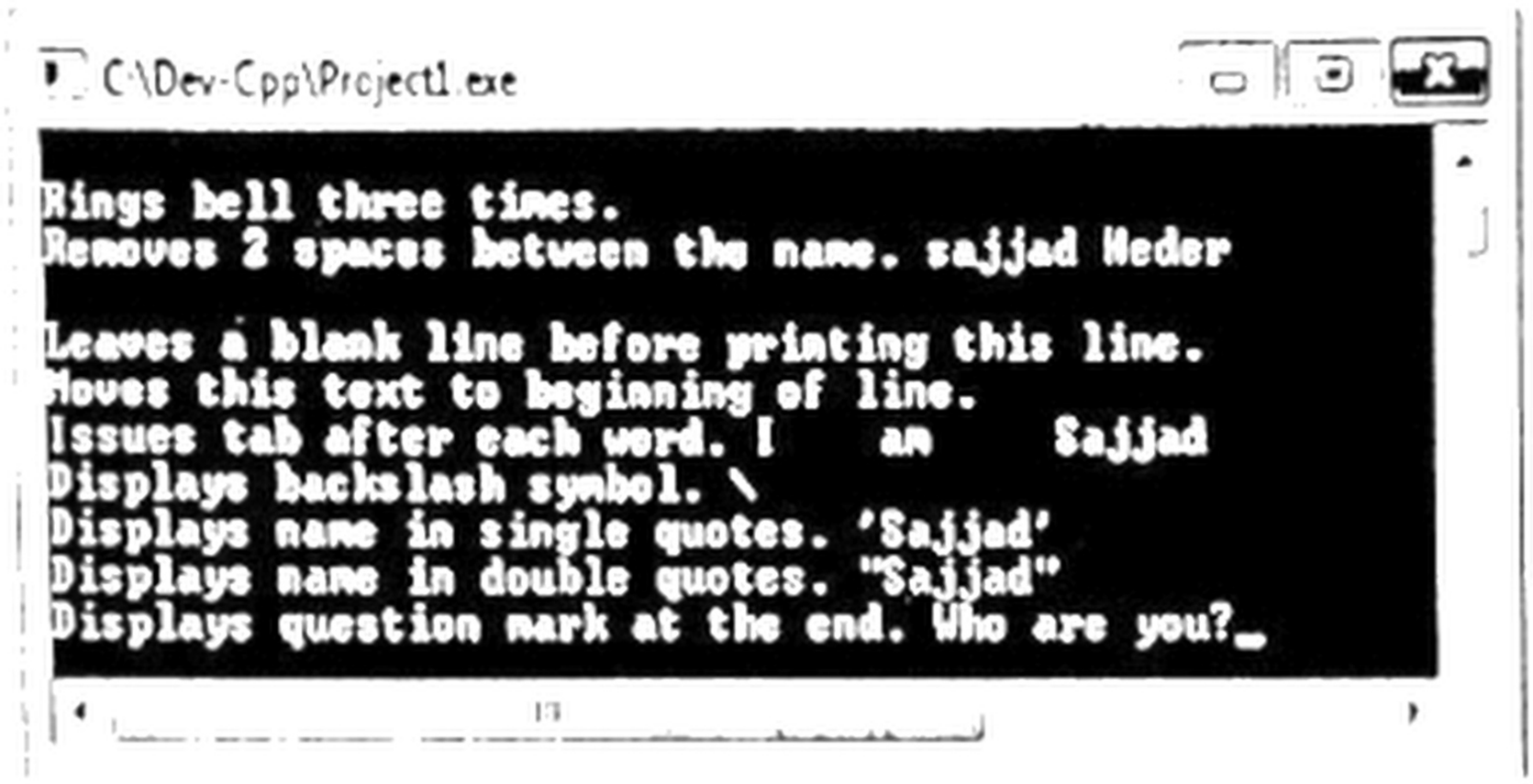
\b	Moves cursor backward by one position
\n	Moves cursor to the beginning of next line
\r	Moves cursor to the beginning of current line
\t	Moves cursor to the nest horizontal tabular position
\ \	Produces a backslash
\'	Produces a single quote
\ ''	Produces a double quote
\?	Produces a question mark

The program in Fig demonstrates the use of escape sequence



Using escape sequences in a program

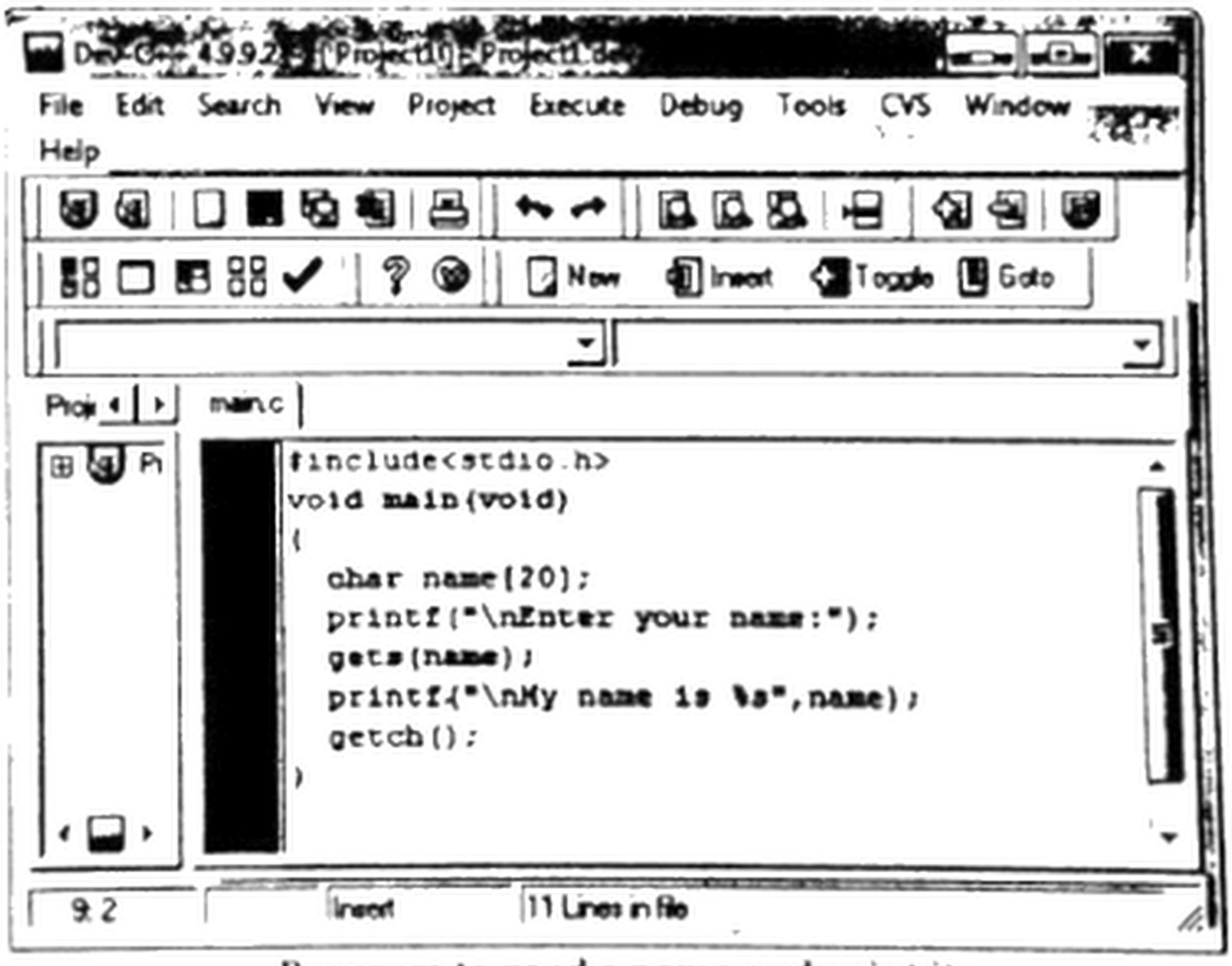
The output of the program is shown in Fig.



iii. What is the purpose of gets() function? Explain with an example.

Ans: The Gets() Function:

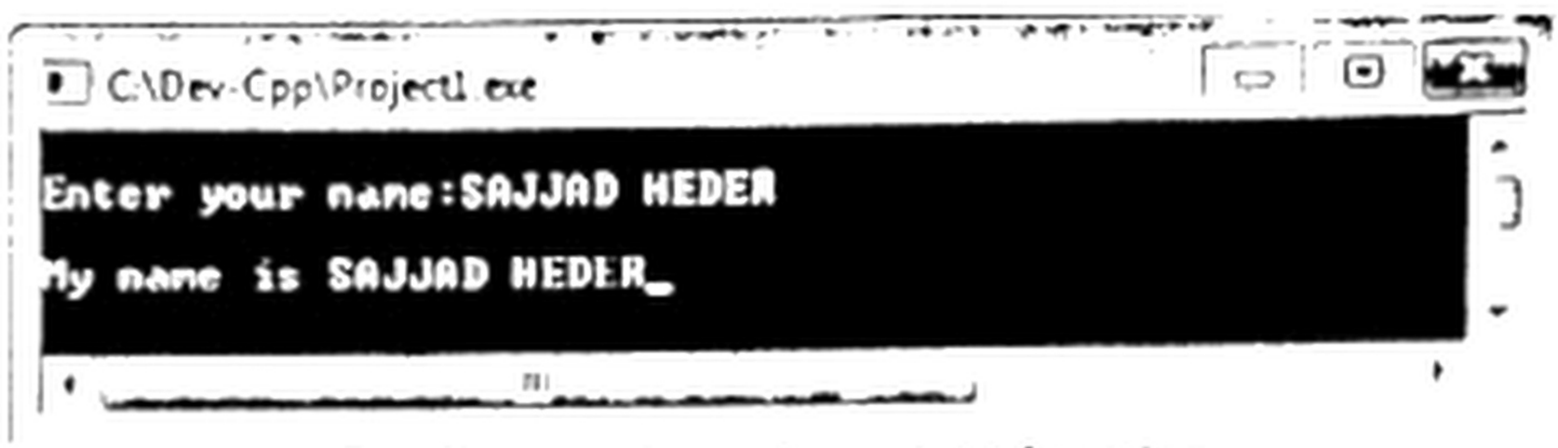
The gets() functions is used to read a string from the keyboard and store it in the variable specified inside the parenthesis. The program in Fig. demonstrates the use of gets() function in a program.



Program to read a name and print it

In this program, the format specifier %s is used to print the string stored in the variable name.

The execution of this program is shown in Fig.



Reading a string using gets( ) function

iv. Differentiate between getch() and getche() functions.

Ans: The Getche() and Getch() Functions:

Sometimes in programming, it is required to read a single character the instant it is typed, without waiting for Enter key to be

pressed. For example, in a game the user might want an object to move each time he presses one of the arrow keys. It would be awkward to press the Enter key each time the user presses an arrow key.

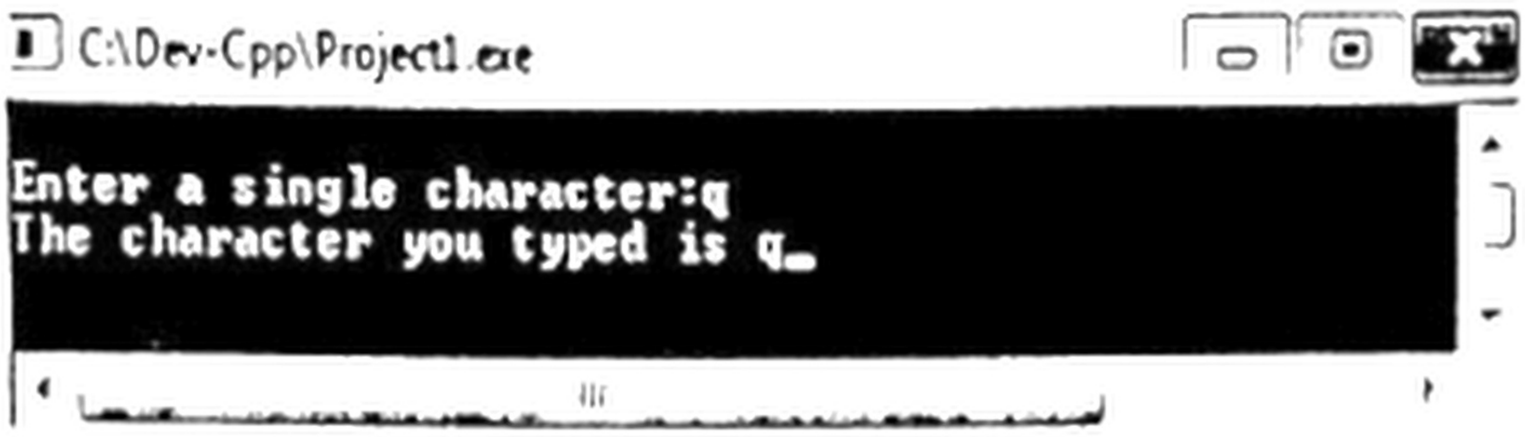
The **getche()** function is used for this purpose. The get means it gets something from an input device. The 'ch' means it gets a character and the 'e' means it echoes (displays) the character to the screen when it is typed.

The program in Fig. reads a single character the instant it is typed and displays it one the screen. The user does not have to press the Enter key after typing the letter.

Program to read a character using getche() function

In this program, the getche() function reads a single character from the keyboard and it is assigned to the variable ch. The format specifier %c is used to print the letter stored in variable ch

The execution of this program is shown in Fig.



Readingacharacterusinggetche() function

The getch() is a similar function to **getche()**, the difference is that it does not display the typed character to the screen. The **getche()** and **getch()** functions require **conio.h** header file, so it must be included in the program as well.

v. Evaluate the following expression.

a) 7+5*(3+4)

b) 100/10/4

c) 50%13%3

d) 30/7*3-6

Ans:

a) 7+5*(3+4)

=7+5*7

=7+35

=42

b) 100/10/4

=10/4

=2.5

c) 50%13%3

=11%3

=2

d) 30/7*3-6

=4.28*3-6

=12.8-6

=6.8

vi. What will be the output of the following program?

```
#include <stdio.h>
Void main(void)
{
  Int z,y,z1,z2,z3,z4;
  X=17;
  Y=5;
  Z1=x/y
  Printf("\nz1=%d",z1);
Z2=x%y;
Printf("\nz2=%d:,z2);
Z3=++x;
Printf("\nz3=%d",z3);
Z4=y++;
  Printf("\nz4=%d",z4);
}
```

Ans: Output:

Z1 = 3

Z2 = 2

Z3 = 18

Z4 = 5

vii. What will be the output of the following program?

```
#include <stdio.h>
Void main(void)
{
        Int b;
        Float a,c,d,e,f;
        A=14.84;
        B=7;
        C=a-b;
        Printf("\nc=%f",c);
        D=a/b;
        Printf("\nd=%f",d);
        E=a-b*3
        Printf("\ne=%f",e);
        F=(a+b)/2;
        Printf(;\nf=%f",f0;}
```

Ans : Output:

C=7.840000

D= 2.120000

E= 6.160000

F=10.920000

EXTENSIVE QUESTIONS

Q3. Describe how basic and compound assignment operators are used?

Ans: Assignment Operators:

Assignment operators are used to assign values to variables used in computer programs. C language provides three types of assignment operators. There are basic assignment operator, compound assignment operators and increment/decrement operators.

Basic Assignment Operator:

The basic assignment operator is =. This is used to assign value of an expression to a variable. It has the general form:

Variable=expression

Where expression may be a constant, another variable to which a value has previously been assigned or a formula to be evaluated. For example:

Sum= a + b;

Compound Assignment Operators:

In addition to =, there are number of assignment operators unique to C. These include +=, -=, /= and %=. Suppose op represents an arithmetic operator. Then, the compound assignment operator has the following general form to assign value of an expression to a variable.

Variable op= expression

For example, consider the following statement:

Sum= sum + n;

This assignment statement could be written using a compound assignment operator as:

Sum += n;

The effect is exactly the same but the expression is more compact.

Some more examples are:

Sum-= n	is equivalent to	sum= sum – n
Prod * = n	is equivalent to	prod = prod *n
A / = b	is equivalent to	a = a / b
A % = b	is equivalent to	a = a % b

Q4. Describe the functions of the following operators?

i) Relational operators

ii) Logical operators

iii) Conditional operator

Ans: i) Relational operators:

Relational operators are used to compare two values of the same type. These are used in expressions when a decision is to be based on a condition. After evaluation of a relational expression, the result produced is either True or False. Relational operators are used in programming for decision making.

Types of Relational Operators:

Six types of relational operators are available in C language. These are described in table

Operator	Definition
= =	Equal to
! =	Not equal to
<	Less than
>	Greater than

< =	Less than or equal to
> =	Greater than or equal to

The following are some examples of relational operators

c>=a+b

x<5.3

n==20

count!=10

If a has the value 25, b has the value 10 and c has the value 28 then first expression is false since 28 is not greater than or equal to 35.

In the second expression, if x has the value 4.5 the expression is true because x is less than 5.3

In the third expression, if n is equal to 20 then the expression will be true. For any value other than 20, the expression will be false.

In the last expression if count is any number other than 10 then the expression will be true. It will only be false when count is equal to 10.

ii) Logical operators:

Logical operators are used for building compound conditions. We have seen before that a single condition is built using a relational operator in an expression. If we need to build more than one condition for some action to take place in programming, then we have to form compound condition.

Types of Logical Operators:

There are three types of logical operators. These are described in table.

Operator	Definition
&&	AND
||	OR
!	NOT

Logical AND (&&) Operator:

It is used to form compound condition in which two relational expressions are evaluated. One relational expression is to the left and the other to the right of the operator. If both of the relational expressions (conditions) are true then the compound condition is considered true otherwise it is false.

Syntax:

Expression 1 && Expression 2

Truth table for AND operator is shown here under:

Expression-1	Expression-2	Expression1 && Expression-2
True	True	True
True	False	False
False	True	False
False	False	False

Example:

Consider the following compound condition:

(a>=1) && (a<=10)

This expression is considered true of both conditions (a>=1) and (a<=10) are true, that is, if the value of a is between 1 to 10.

In the next example, the compound condition is considered true if the value stored in a is greater than the value stored in b and the value stored in c is equal to 15.

(a>b) && (c= =15)

The following compound condition will check whether the character stored in variable **ch** is a lowercase letter or not.

(ch>='a') && (ch<='z')

Logical OR (II) Operator:

Logical OR operator is also used to form a compound condition. Just like the logical AND operator, one relational expression is to the left and the other to the right of the OR operator. The compound condition is true if either of the conditions is true or both conditions are true. It is considered false only if both of the countries are false.

Syntax:

Expression 1 II Expression 2

Truth table for AND operator is shown here under

Expression-1	Expression-2	Expression-1 II Expression-2
True	True	True
True	False	True
False	True	True

False	False	False

Example:

(n<10)II(n>25)

Suppose the value of n is 5, then the expression will be considered true because one of the two conditions is true. If the value of n is 28 then also the compound condition will be true. If the value of n is 12 then the expression will be false since both conditions are false.

The next compound condition will be true if a is greater than b or c is equal to 10. It will also be true if both conditions are true, that is, a is greater than b and c is equal to 10. It will only be false if a is not greater than b and at the same time c is not equal to 10.

(a<b) II (C=10)

Logical OR condition is used when we wish to perform an operation if one of two conditions is true or both of the countries are true.

Logical NOT (!) Operator

The logical NOT operator is used with a single expression (condition) and evaluates to true if the expression is false and evaluates to false if the expression is true. In other words, it reverses the result of a single expression.

Syntax:

!Expression1

Truth table for AND operator is shown here under:

Expression	!Expression
True	False

False	True

For example, the expression:

!(a>b)

Will be true if a is not less than b. In other words, the condition will be true if a is greater than or equal to b. The same condition can also be written as given below which is easy to understand

(a>=b)

i. Conditional operator/ Conditional (Ternary) Operator:

A conditional operator is a decision-making operator. It has the following form.

Condition?expression1 : expression2:

When this statement is executed the condition is evaluated. If it is true, the entrie conditional expression takes on the value of expression1. If it is false, the conditional expression takes on the value of expression2. The entire conditional expression takes on a value and can therefore be used in an assignment statement. Consider the following example.

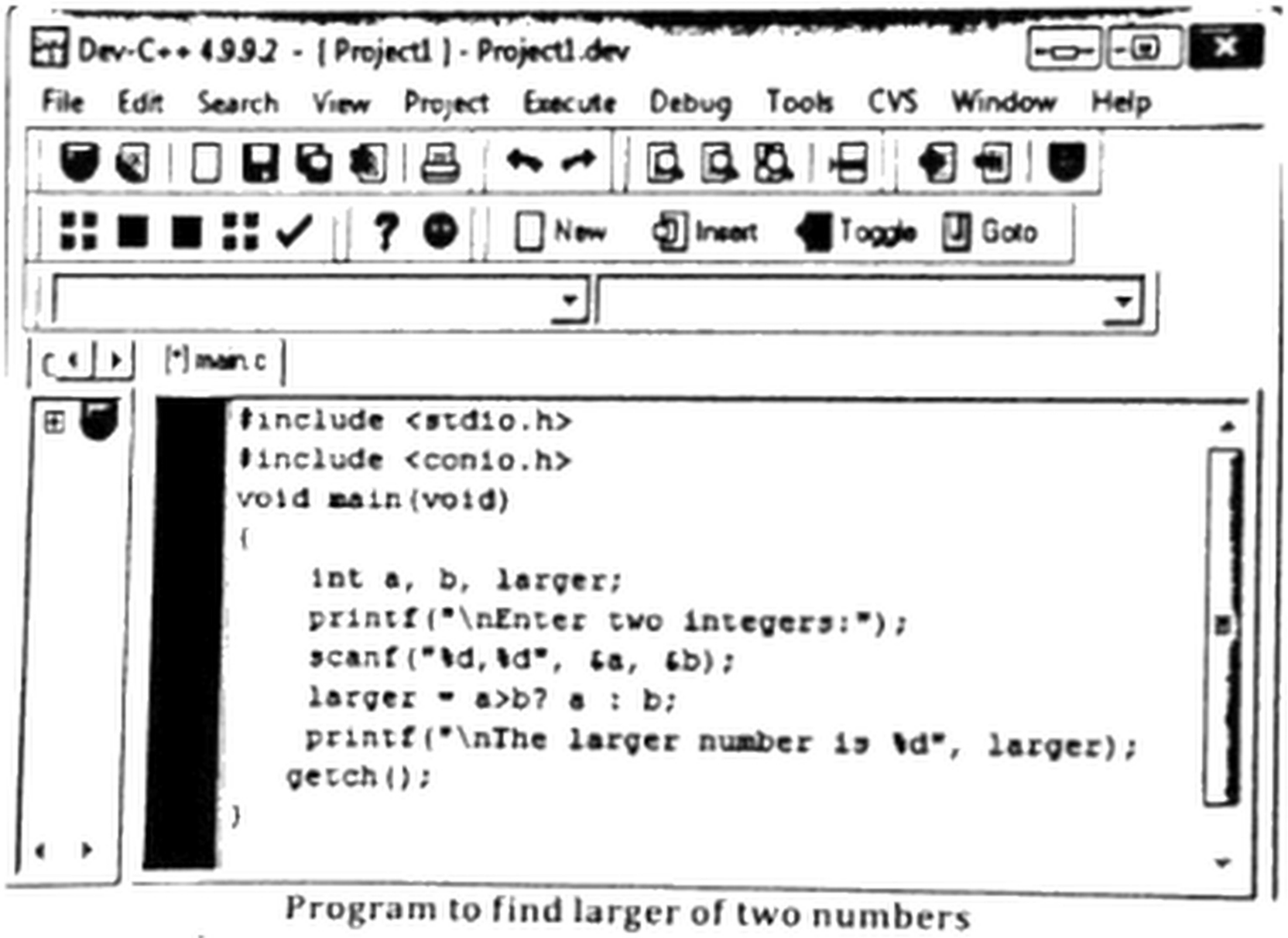
A= (k>15)? X*y : x+y;

This statement will assign the product of x and y to the variable a, if k is greater than 15, otherwise a will be assigned the sum of x and y. This expression is equivalent to the following if-else statement which will be explained in the next unit.

```
If (k>15)
        A= x*y;
Else
        A= x+y;
```

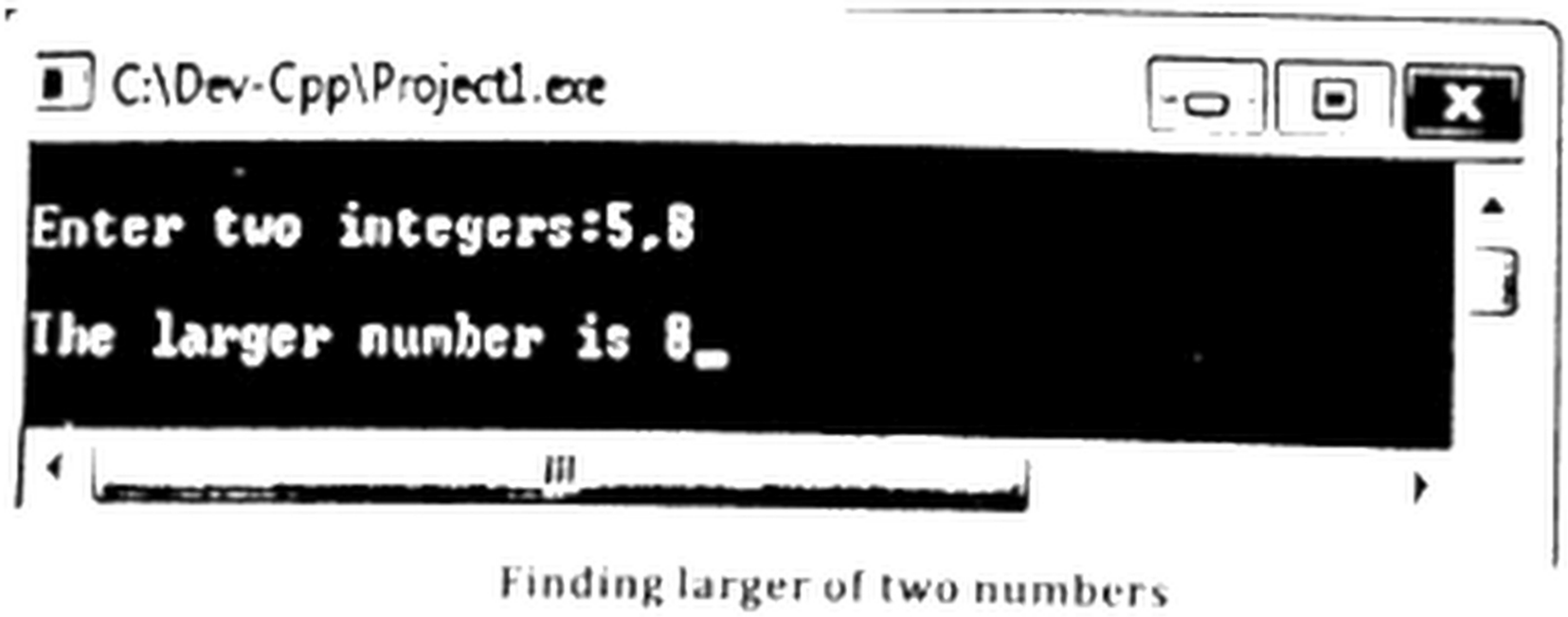
Some programmers may prefer to use the above if else statement rather than using the conditional operator because it is easy to understand.

The program in Fig, demonstrate the use of conditional operator for finding the larger of two numbers.



Program to find larger of two numbers

The execution of the program is shown in Fig.



Finding larger of two numbers

Q5. Write a program that reads three numbers and prints their sum, product and average.

Ans: #include <stdio.h>

Int main()

{

Int num1, num2, num3, sum, prod, avg;

Printf("Enter the three numbers separated by ','\n");

Scanf("%d, %d, %d", &num1, &num2, &num3);

Sum= num1+ num2 + num3;

Prod=num1 * num2 * num3;

Avg=sum/3

Printf("the sum of the three numbers is : %d\n", sum);

Printf("the prod of the three numbers is : %d\n", prod);

Printf("the average of the three numbers is : %d\m", avg);

}

Q6. Write a program that reads the length and width of a rectangle and prints its area.

Ans: #include <stdio.h>

Main()

{

Float length, width, area;

```
Printf("Enter length of rectangle: ");

Scanf("%f",&length);

Printf("Enter width of rectangle:");

Scanf("%f",&width);

Area= length * width;

  Printf("Area of rectangle = %f sq.units" , area);

}
```

Q7. Write a program that reads the length of one side of a cube and prints its volume.

Ans:

```
#include <stdio.h>

#include <conio.h>

Main()

{

    Float side, volume;

    Printf("Enter length of any side of cube\n");

    Scanf("%f, &side);

    Volume = side*side*side;

    Printf ("Volume of Cube : %0.4f\n", volume);

}
```

Q8. Write a program that reads temperature in Celsius , converts it into Fahrenheit and prints on the screen.

```
Ans: #include <stdio.h>

#include <conio.h>

Main()

{

   Float farhren, Celsius;

   Printf("Enter the temperature in Celsius\n");

   Scanf("%f", &Celsius);

   Fahren = (9.0/5.0) * Celsius + 32;

   Printf("%.2fc is equal to %.2F\n", Celsius, fahren);

}
```

Q9. Write a program that reads name and address of a person and prints it on the screen using gets() and puts() functions.

```
Ans:  #include <stdio.h>

#include <string.h>

Int main()

{

  Char name [50], add[50];

  Printf("enter your name:");

  Gets(name);

  Printf("Enter your address:");

Gets(add);

Printf("your name is:");
```

```
    Puts(name);
    Printf("your address is:");
    Puts(add);
}
```