EXERCISE

Q1. Select the best answers for the following MCQs.

i. **Which structure enables the programmer to execute a set of instructions repeatedly until a particular condition is met?**

A. Selection
B. Sequence
C. Choice
D. Loop

ii. **Which of the following is also called counter loop?**

A. Do while
B. While
C. For
D. If-else

iii. **Which of the following ends a multiple-statement while loop?**

A. Right bracket
B. Right brace
C. Semicolon
D. Colon

iv. **Which loop is used to execute a set of statements repeatedly for a fixed number of times?**

A. For loop
B. Do while loop
C. While loop
D. Nested loop

v. **Which loop is used when it is required to execute the loop at least once?**

A. For loop
B. Do while loop
C. While loop
D. Nested loop

vi. **Which loop is preferred to use when the number of times the loop will execute is not known in advance?**

A. For loop
B. Do while loop
C. While loop
D. Nested loop

vii. **In which loop condition is placed at the end?**

A. For loop
B. Do while loop
C. While loop
D. Nested loop

viii. **Which statement is used to exit from a loop as soon as certain condition is met?**

A. Break
B. Continue
C. If statement
D. Default

ix. **What is a pass through a loop called?**

A. Execution
B. Iteration
C. Enumeration

D. Culmination

x. Which of the following should be placed at the beginning and the end of a for loop if it consists of more than one statement?

A. ()

B. <>

C. []

D. {}

Answers

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

SHORT QUESTIONS

Q2. Give short answers to the following questions.

i. Differentiate between for loop and while loop.

Ans: For loop:

The **for** statement is used to executed a set of statements repeatedly for a fixed number of times in a program.

While loop:

The while statement is used to implement repetition structure in a program when the number of iterations is not known in advance and the repetition continues until some condition remains true.

ii. Differentiate between while loop and do while loop.

Ans: Difference between While() and Do-While() Loops:

	While() loop	Do-WHILE () Loop
1.	While() loop is pre-tested loop.	Do-while loop is post-tested loop.

2.	The syntax or general form of while() loop is:
	While(condition)
	{
	Statements; //body of loop
	}
		The syntax or general form of do-while() loop is:
		Do{
		Statements; // body of loop.
		}while(condition);

3.	In 'while' loop the controlling condition appears at the start of the loop.	In do-loop the controlling condition appears at the end of the loop.
4.	The iterations do not occur if, the condition at the first iteration appears false	The iteration occurs at least once even if the condition is false at the first iteration.

iii. What will be the output of the following code?

```
Int k;
For(k=1;k<=5;k++)
        Printf("\nI am a student");
Printf("\nGOOD BYE");
```

Ans: I am a student

I am a student

I am a student

I am a student

I am a student

GOOD BYE

iv. What will be the output of the following code?

```
Int n;
For (n=30;n>=10;n-5)
    Printf("\b%d",n);
```

Ans: 30

25

20

15

10

v. Find errors in the following code.

```
Int k,a
A=3;
K=1;
While(k<10):
{
Printf("\n%f\t%f",k,k*a-1);
K=k+2;
}
```

```
Ans:  int a, k;
A=3;
K=1;
While(k<10)
{
        Printf("\n%d\t%d",k,k*a-1);
        K=k+2;
}
```

Output:

```
1       2
2       8
5       14
 7      20
 9      26
```

vi. Convert the following for loop into a while loop.

```
Int k;
For (k=25; k>0; k=k-3)
        Printf("\n%d",k);
Ans:  int k;
      K=25;
     While (k>0)
{
        Printf("\n%d",I);
        K=k-3;
}
```

EXTENSIVE QUESTIONS

Q3. What is looping structure? Explain for loop with examples.

Ans: A loop Structure:

A loop is a structure that enables the programmer to execute the same sequence of statements repeatedly until a particular condition is met.

For loop/ For Statement:

The for is a looping statement which is used to execute a set of statements repeatedly for a fixed number of times. It is also known as counter loop. It has the **general form:**

For (initialization; test condition; increment/decrement)

{

Body of the loop

}

When for statement is executed, a variable (also known as loop variable) is assigned an initial value in the initialization part of the loop, such as k=1 or count=0. The value of the loop variable is checked with the given test condition. The test condition is a relational expression, such as k<10. If the condition is true, the control enters the body of the loop otherwise it will exit the loop.

After the execution of the body of the loop, the control is transferred back to the increment/decrement part of the loop. The loop variable is

incremented or decremented using an assignment statement such as k=k+1. The new value of loop variable is again checked with the test condition. If the condition is satisfied then the body of the loop is again executed. This process goes on till the last condition becomes false.

Body of the loop may have one or more statements. If it contains only a single statement then braces are not needed.

Note: all the three expressions such as initialized, test condition and increment are optional. You can omit any or all in for statement.

Example: For example the following all for statements are valid.

```
For (; ;)

For(int i=1 ; ;)

For(; k<10 ; k++)

For(; ; k++)

For(; x<12 ;)
```

Q4. Explain while and do while loops with examples.

Ans: While loops/The While Statement:

A repetition structure when the number of iterations is known in advance and the repetition continues until test condition remains true.

The **while** statement has the **general form:**

```
While (test condition)
{
   Body of the loop
}
```

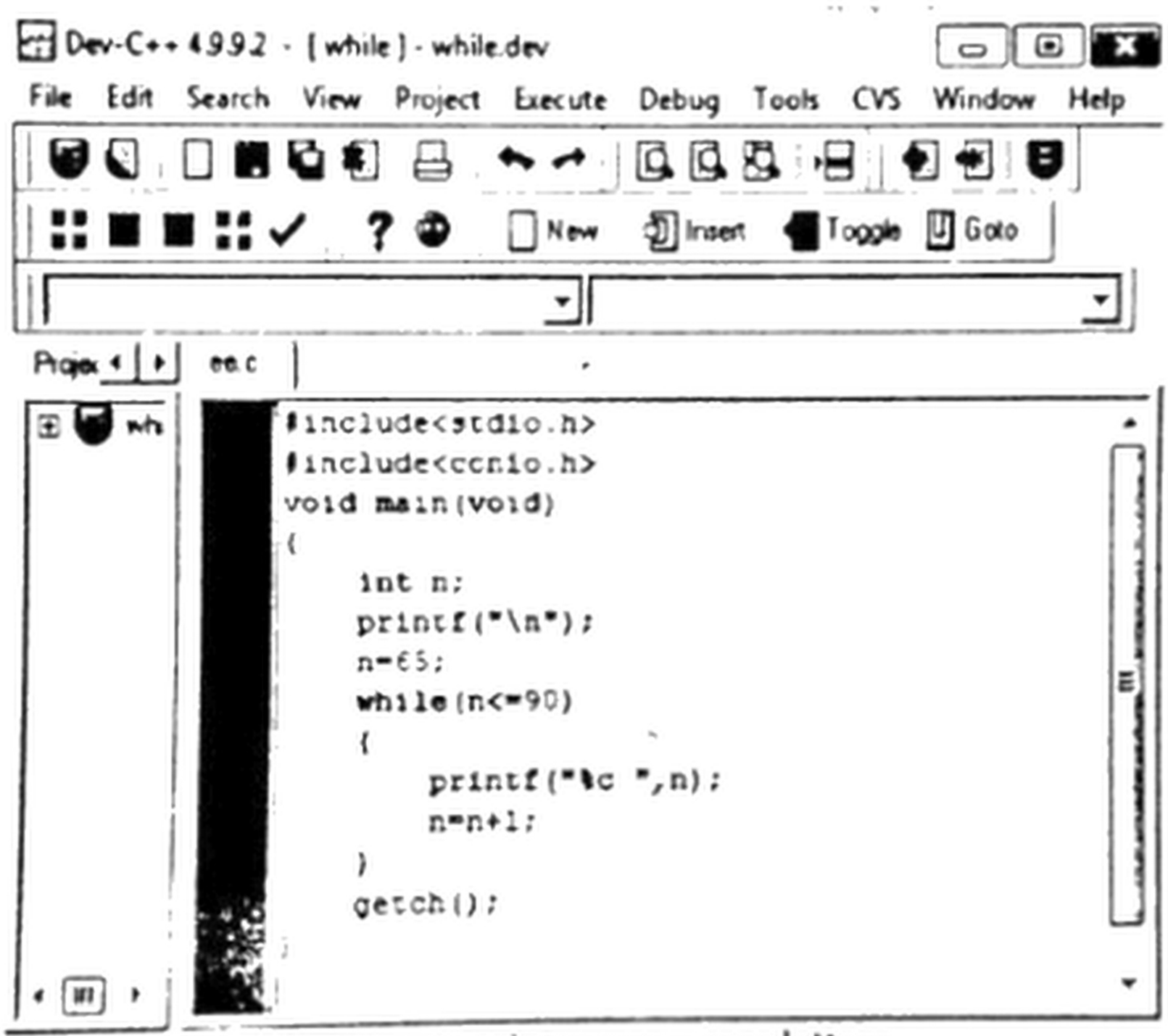
When a while statement is executed, the computer first evaluates the

test condition. If it is true, body of the while loop is executed. After the execution of the body of the loop, the test condition is again evaluated and if it is true, the body of the loop is executed once again. The process continues until the test condition becomes false. When it becomes false, the control is transferred to the first statement following the end of body of loop.

The body of the loop can be a single statement or it can be multiple statements.

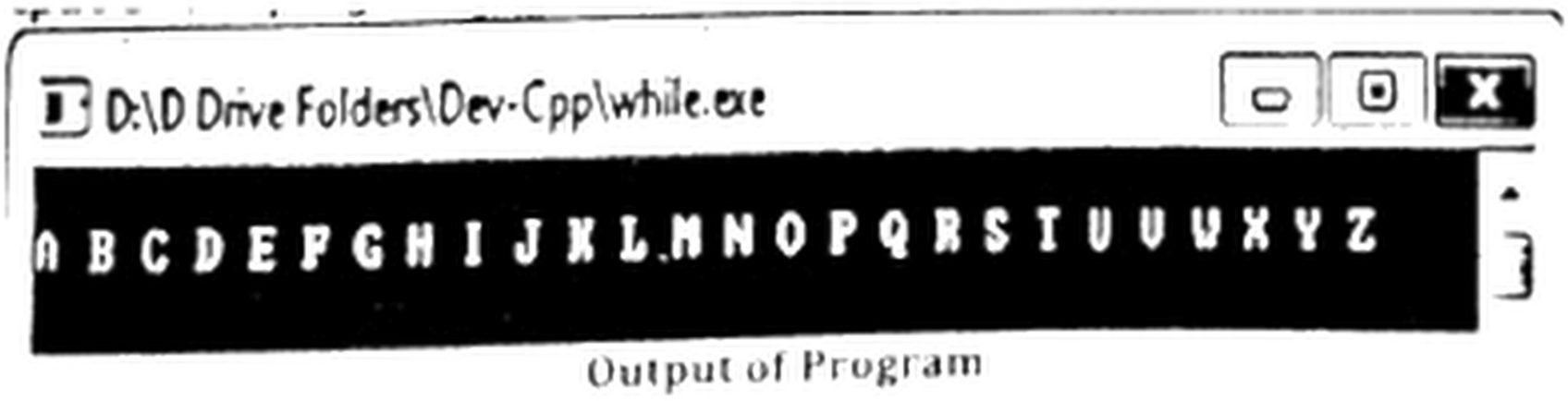
Examples:

Program: The program in Fig prints all the upper case letters on a single line using while loop. The ASCII code for upper case letters are in the range 65 to 90.



Program to print upper – case letters

Output of the program is shown in Fig.



Output of Program

Do-while loops/The Do While Statement:

The do while statement is used to implement loop structure when it is required to execute the loop at least once. The general form of the **do while** loop is given below.

Do

{

Body of the loop

}

While (test condition)

The statement **while (test condition)** is placed at the end of the loop so that the body of the loop is executed at least once whether the condition is true or false. There is a semicolon after the test condition because it is at the end of loop. If the body of loop contains a single statement then braces are not required.

Program: The program in Fig, prints all the lower-case letters of the alphabet using **do while** loop. The ASCII codes for the lower case letters are in the range of 97 to 102.

Program to print lower - case letters

The output of this program is shown in Fig.

Output of Program

- In this program, the integer variable n is initialized to 97 which is the ASCII value of letter a, before entering the loop.
- The expression in the do while statement, n<=122 acts as the test condition. The statements within the loop execute as long as this expression remains true.
- The statement n=n+1 increment the value of n by one with each iteration. Note that the ASCII value of z is 122.

Q5. Explain the purpose of break and continue statements with one example each.

Ans: The Break Statement:

C language provides the break statement to exit from a loop as soon as certain condition occurs. It is used in **for, while** and **do while** statements. Break statement is also used to exit the body of **switch** statement after executing the statements under a **case** and transfers control to the first statement following the end of the **switch** statement.

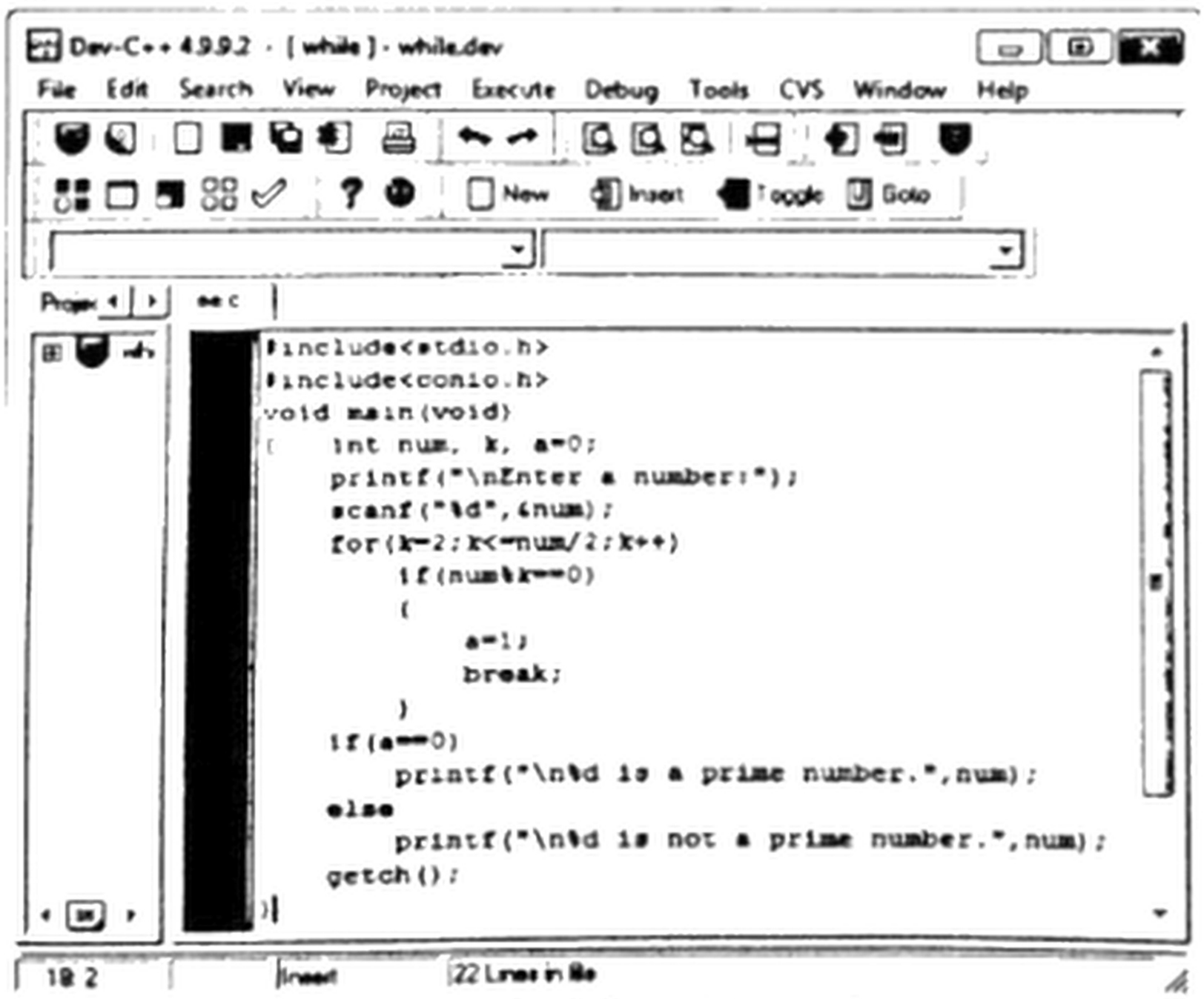
The Continue Statement:

Sometimes during the execution of a loop, when a certain condition occurs after executing a statement, it may be required to skip the remaining statements within the body of the loop and continue for next iteration until the test condition increment. C language provides the continue statement to achieve this task. The continue statement causes the loop to be continued with the next iteration after skipping the remaining statements within the body of the loop.

The general format of continue statement is:

Continue;

Program to check for prime number by using break statement:

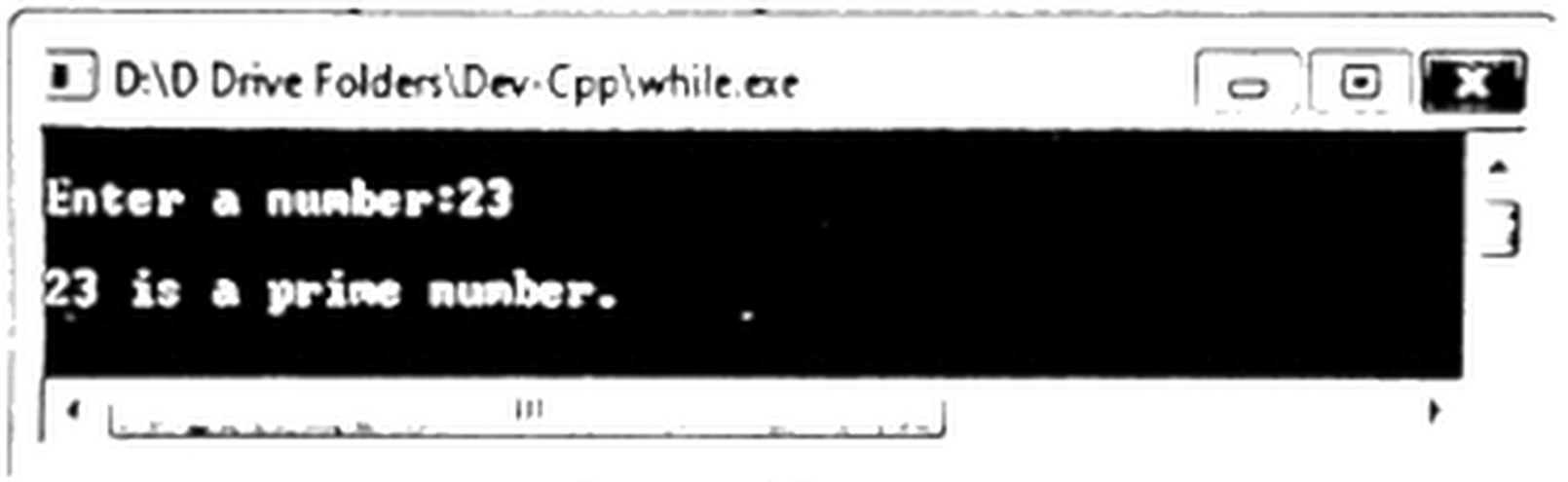


Program to check for prime number

- When the above program is executed, it reads a number and determines whether it is a prime number or not. The integer variable a is assigned the value 0 to assume that the number num is a prime minister.
- The remainder operator (%) is used inside the for loop to check if the number is exactly divisible by any number in the range 2 to num/2.
- If this condition is true, the variable a is assigned the value 1, indicating that num is not a prime number.
- At this point, the break statement is used to exit for loop immediately.

After the execution of the for loop, the value of a is checked, if it is 0 then the message telling the user that num is a prime number is printed otherwise the message, num is not a prime number is printed.

The output of the program is shown in Fig.



Output of Program

Program to read 10 numbers and find the sum of those that are less than 20 by using continue statement:

Program to add numbers less than 20

- When the program is executed, the user enters 10 numbers one by one.
- If a number less than 20 is entered the condition of if statement is false and the number is added in the sum
- If a number is greater than or equal to 20 is entered, the number is not included in the sum as the conditions is true and the last statement of

the loop that adds the number Is skipped using the continue statement.

- After ten iterations the loop terminates and the last segment of the program prints the sum of those numbers that are less than 20.

Execution of Program

Q6. What is nested loop? Give two examples.

Ans: Nested Loops:

In C language it is allowed to nest loops within another loop. Nested loops can be of any kind. For example, the programmer can nest a for loop inside a **while** loop or inside a **do while** loop. Loops can be mixed in any way as required.

Example # 1:

Programs uses nested loop to print the products of numbers as given below.

1 × 1 =1

1 × 2 =2

1 × 3 =3

1 × 4 =4

2 × 1 =2

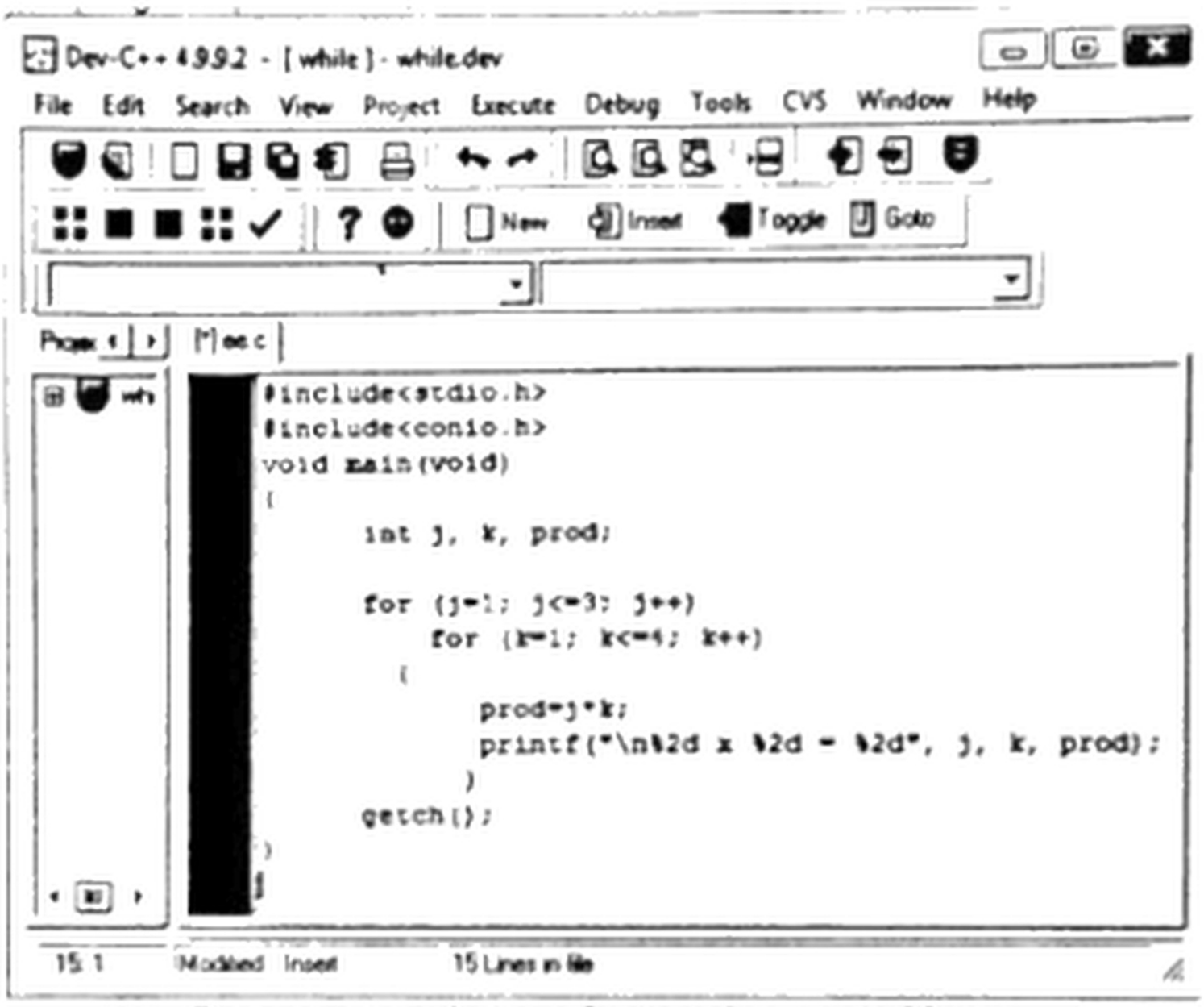
2 × 2 =4

2 × 3 =6

2 × 4 =8

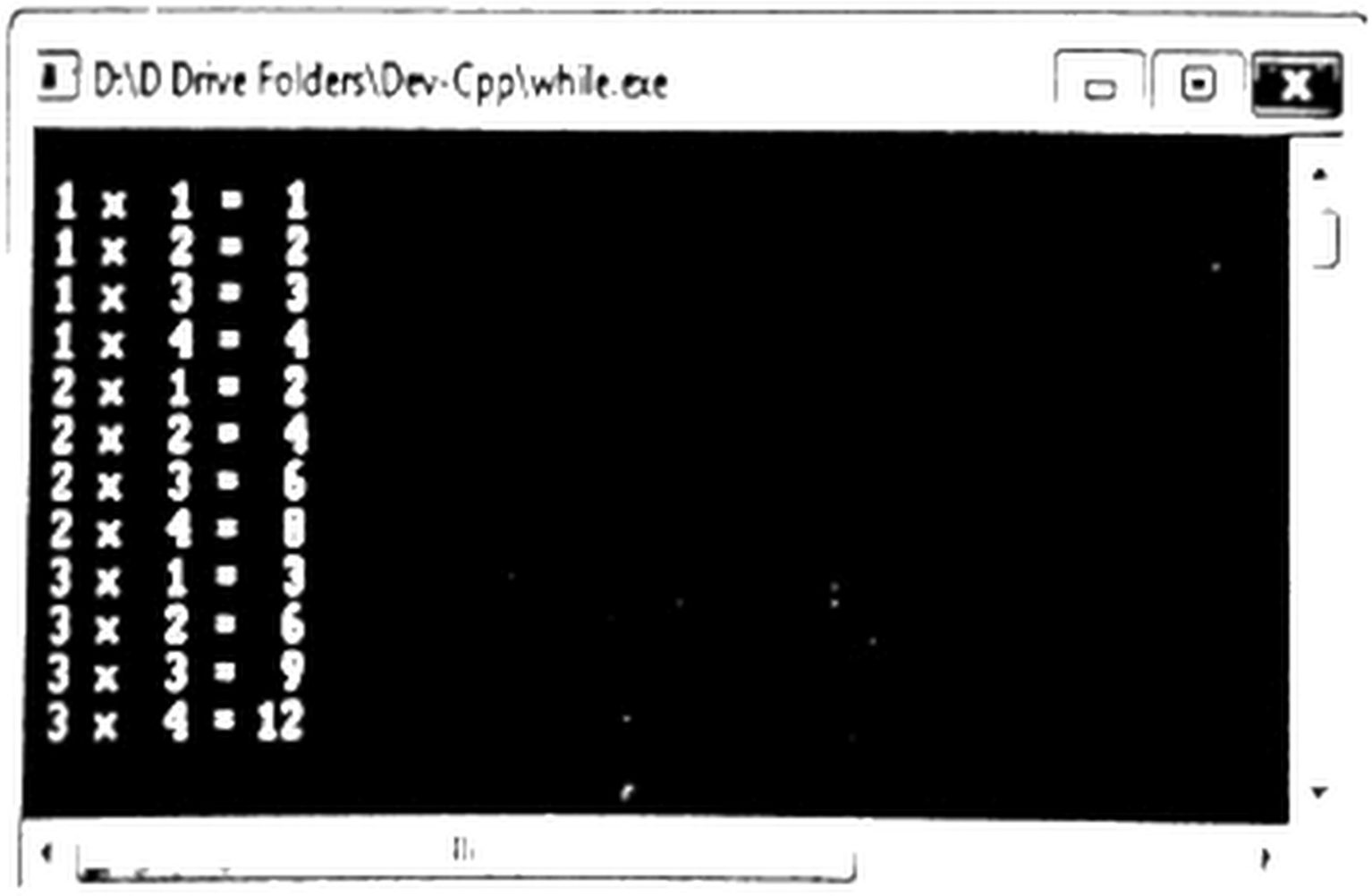
3 × 1 =3

The program prints the products of number



Program to print products using nested loop

The output of the program is shown in Fig.



Output of Program

- The loop variable j is assigned its initial value 1 and the nested loop

  For (k=1; k<=4; k++)

  {

  Prod=j*k;

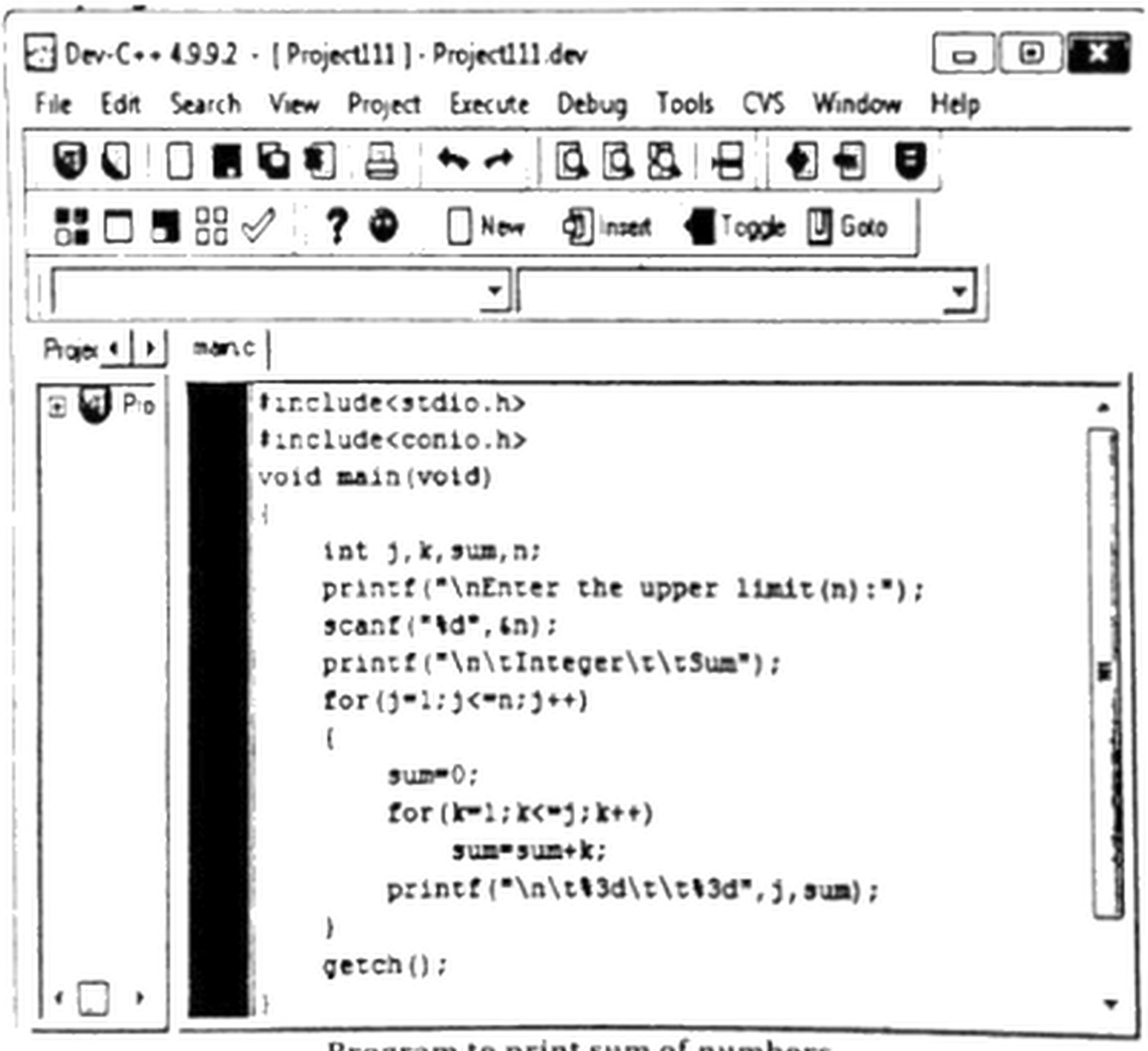
  Printf("\n%2d x %2d = %2d", j, k, prod);

  }

  Is executed. This calculates and displays the first four products 1×1, 1×2, 1×3 and 1×4
- The value of j is then incremented by 1 and the inner loop is executed again. This calculates and displays the next four products, 2×1, 2×2, 2×3 and 2×4.
- Finally, j is incremented to 3, giving the last four products, 3×1, 3×2, 3×3 and 3×4.
- In this program braces must be used in the inner for loop because there are more than one statements to be executed.

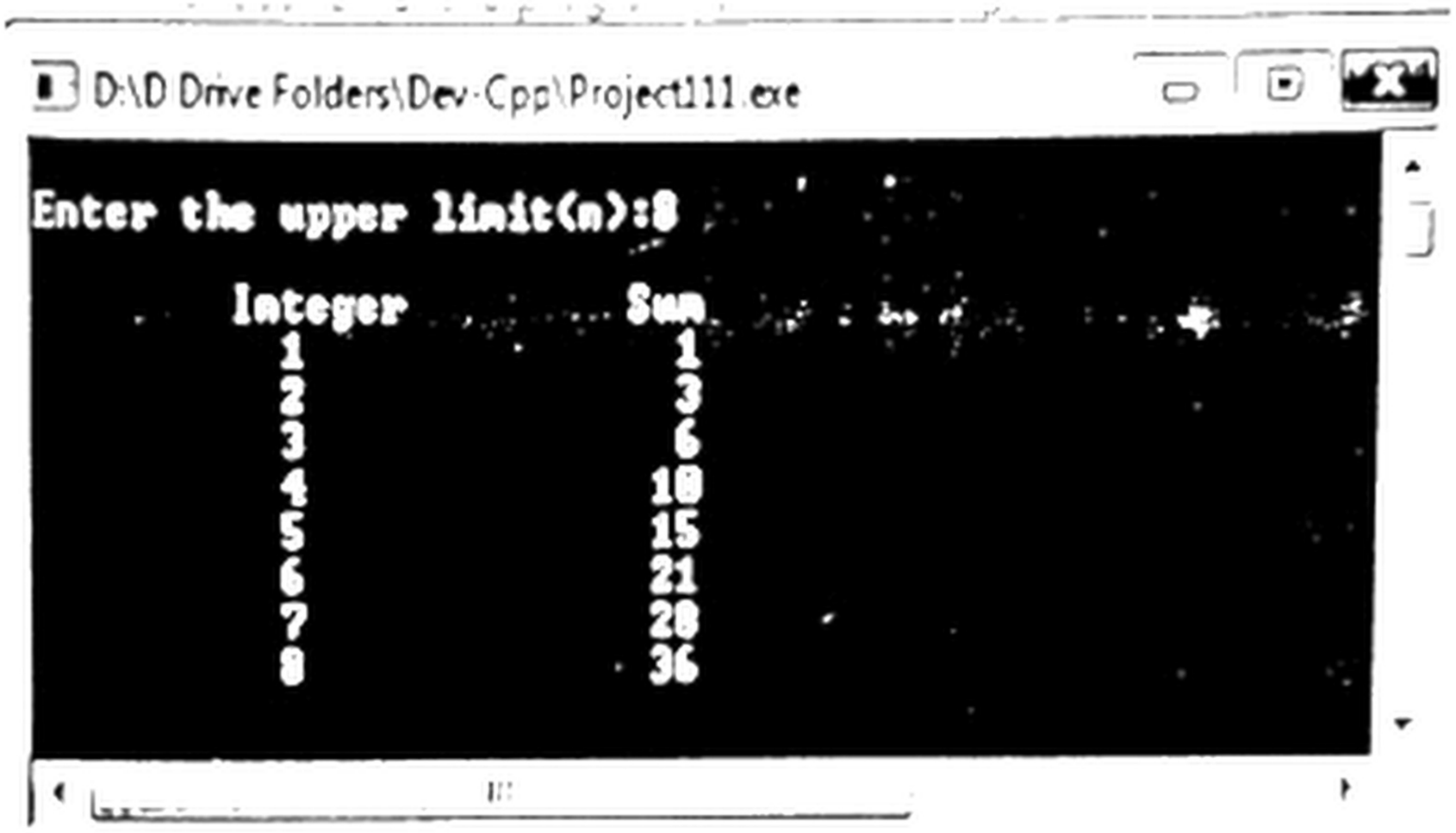
Example # 2:

Program uses nested loop to calculate the sum of the integers for each integer from 1 to n, where n is the value that the user of the program enters.



Program to print sum of numbers

The execution of the program is shown in Fig



Execution of Program

Lab Activities

1. Write a program to print the sum of even numbers from 1 to 50.

   Sum= 2+4+6+.....+50

2. Write a program to print the given sequence of numbers using a loop on a single line.

   5 10 15 20 25 25 30 35 40 45 50

3. Write a program to print the sum of squares of all the numbers from 1 to 10 using a loop.

   Sum =

4. Write a program that prints all the upper case letters in reverse order on a single line using a loop.

   Z X Y W U V T S R Q P O N M L K J I H G F E D C B A

5. Write a program that reads a number and print its table using while loop.

6. A class of 15 students takes an examination in which marks range from 1 to 100. Write a program to find and print the average marks.
7. A class of 20 students takes an examination in which marks range from 1 to 100. Write a program to print the number of students passed and failed. Passing marks are 33.
8. Write a program that prints a line of 60 asterisks (*) using do while loop.
9. Write a program that prints table of squares of all the numbers from 1 to 10 as shown below.

Number	Square
1	1
2	4
3	9
4	16
5	25
6	36
7	49
8	64
9	81
10	100