C to C++

Class by CSI-LNMIIT Student Chapter

BASIC INTRODUCTION OF C++

C++ is derived from the C Language, in a broader sense it is a **superset of C**

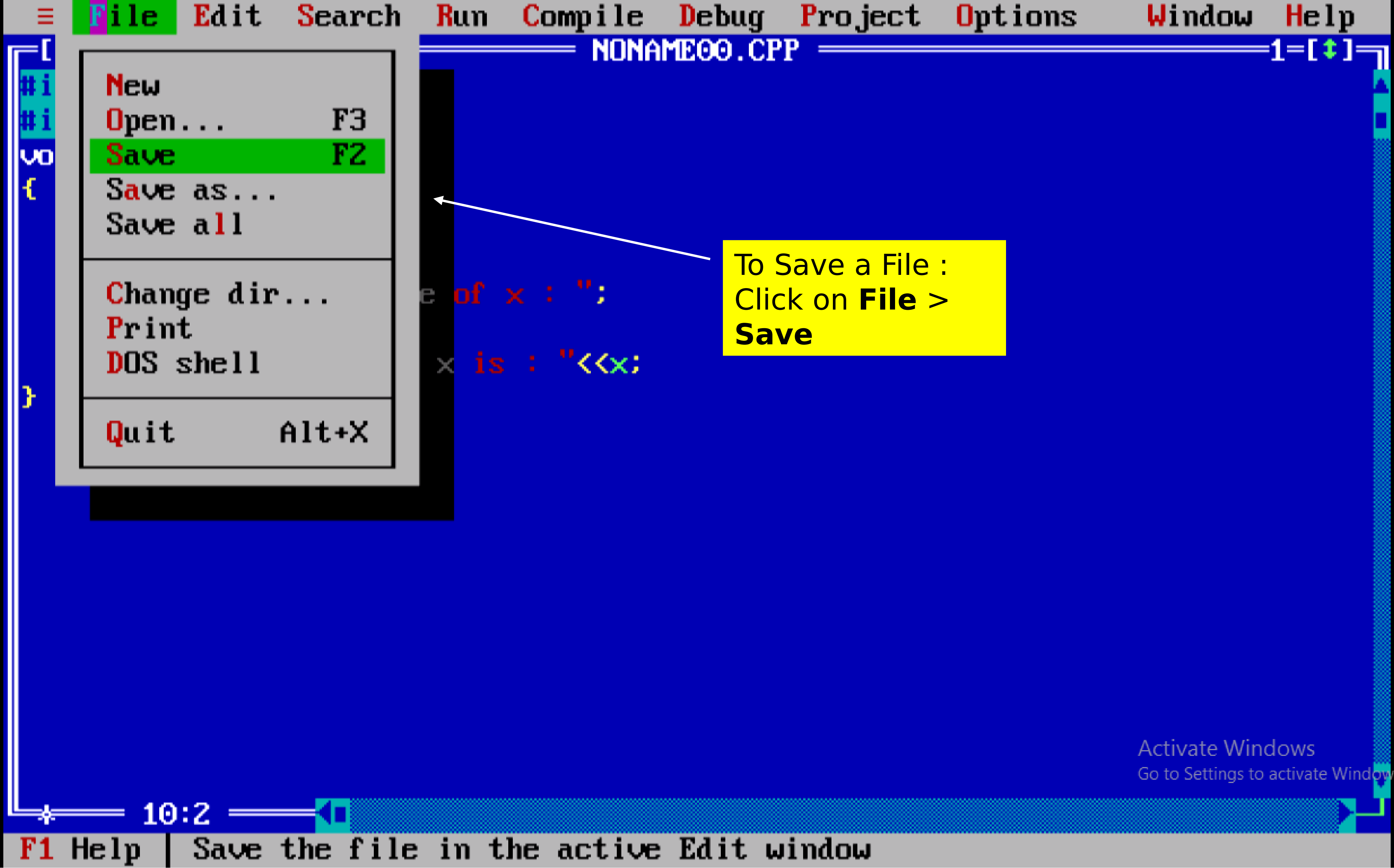
Earlier C++ was known as C with classes.

In C++, the **major change** was the **addition of classes** and a mechanism for inheriting class objects into other classes.

All C operators are valid in C++ and most C Programs can be compiled in a C++ compiler.

C++ expressions are the same as C expressions.

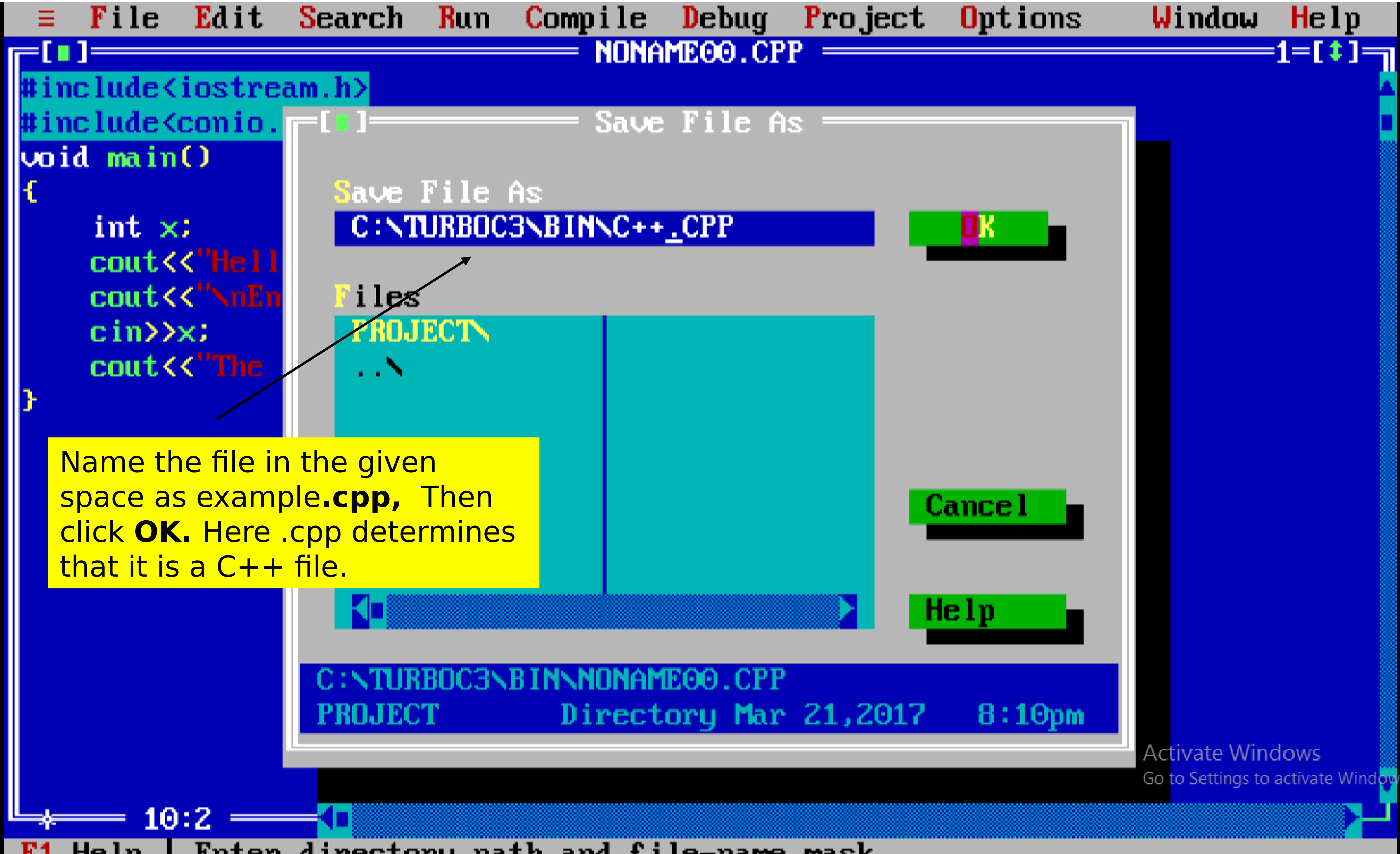
SAVING A FILE IN TURBO C++



```
#i
#i
vo
{
  e of x : ";
  x is : "<<x;
}
```

F1 Help | Save the file in the active Edit window

To Save a File :
Click on **File** >
Save


≡ File Edit Search Run Compile Debug Project Options Window Help
[■]
NONAME00.CPP
1=[↕]
10:2
[■]
OK
Cancel
Help

```
#include<iostream.h>
#include<conio.
void main()
{
      int x;
      cout<<"Hell
      cout<<"\nEn
      cin>>x;
      cout<<"The
}
```

Save File As

Save File As

C:\TURBOC3\BIN\C++_.CPP

Files

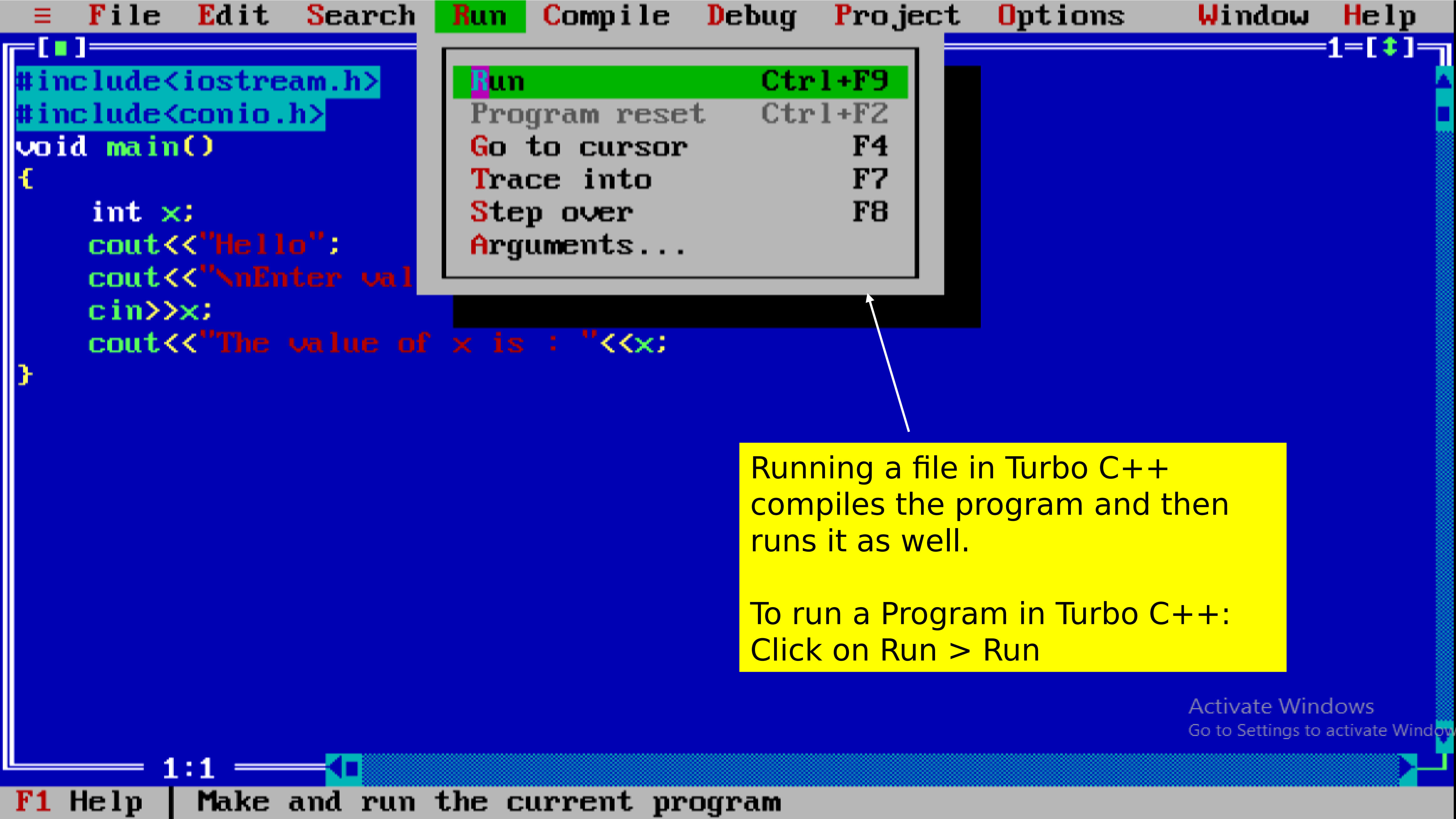
PROJECT\
..\

C:\TURBOC3\BIN\NONAME00.CPP
PROJECT Directory Mar 21,2017 8:10pm

F1 Help | Enter directory path and file-name mask

Name the file in the given space as example**.cpp,** Then click **OK.** Here .cpp determines that it is a C++ file.

RUNNING A FILE IN TURBO C++

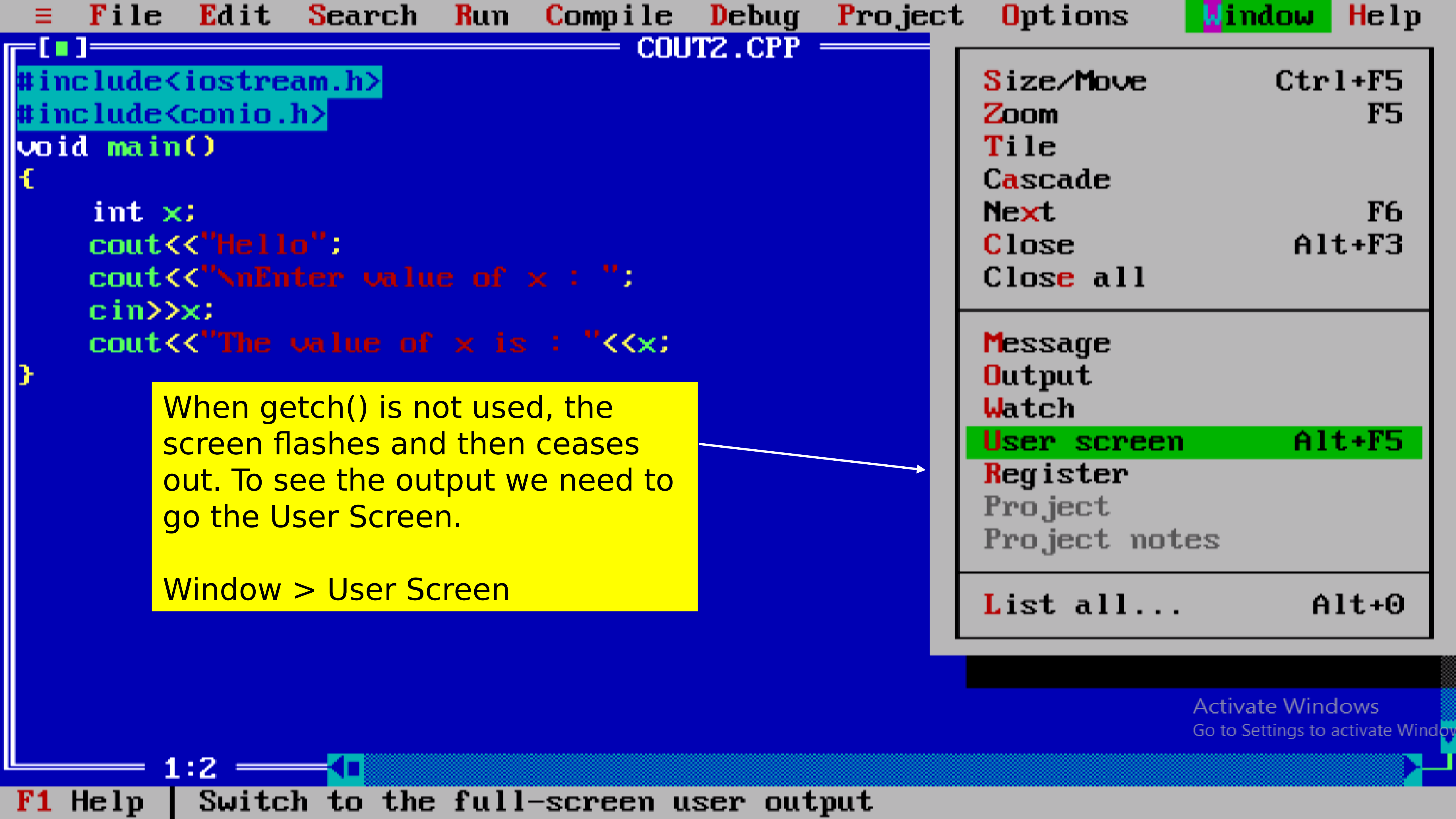


```
#include<iostream.h>
#include<conio.h>
void main()
{
     int x;
     cout<<"Hello";
     cout<<"\nEnter val
     cin>>x;
     cout<<"The value of x is : "<<x;
}
```

F1 Help | Make and run the current program

Running a file in Turbo C++ compiles the program and then runs it as well.

To run a Program in Turbo C++:
Click on Run > Run



```
#include<iostream.h>
#include<conio.h>
void main()
{
	int x;
	cout<<"Hello";
	cout<<"\nEnter value of x : ";
	cin>>x;
	cout<<"The value of x is : "<<x;
}
```

When getch() is not used, the screen flashes and then ceases out. To see the output we need to go the User Screen.

Window > User Screen

F1 Help | Switch to the full-screen user output

cout statement

This command is used to print the output on display screen.

Syntax of cout Statement :

For formatted output operations, cout is used together with the **insertion operator**, which is written as **<<** (i.e., two "less than" signs).

```
cout << "Output sentence";       // prints Output sentence on screen
cout << 120;                     // prints number 120 on screen
cout << x;                        // prints the value of x on screen
cout << "Hello";                 // prints Hello
cout << Hello;                   // prints the content of variable Hello
```

Multiple insertion operations (<<) may be chained in a single statement, this is know as **Cascading of Operators**.

```
cout << "This " << " is a " << "single C++ statement";
```

The output of above syntax:

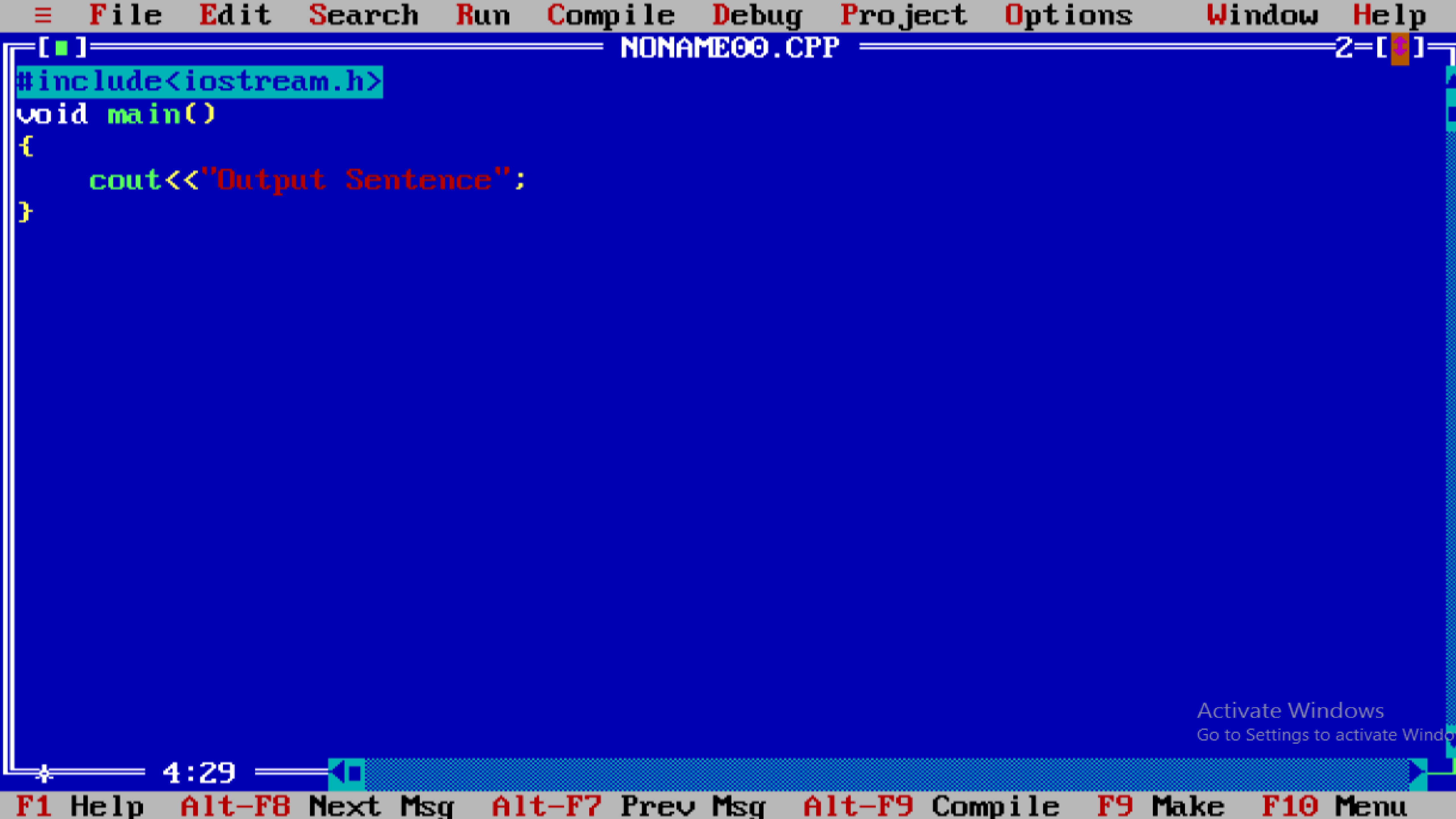
```
This is a single C++ statement
```

```
cout << "I am " << age << " years old and my zipcode is " << zipcode;
```

Assuming the age variable contains the value 24 and the zipcode variable contains 90064, the output of the previous statement would be:

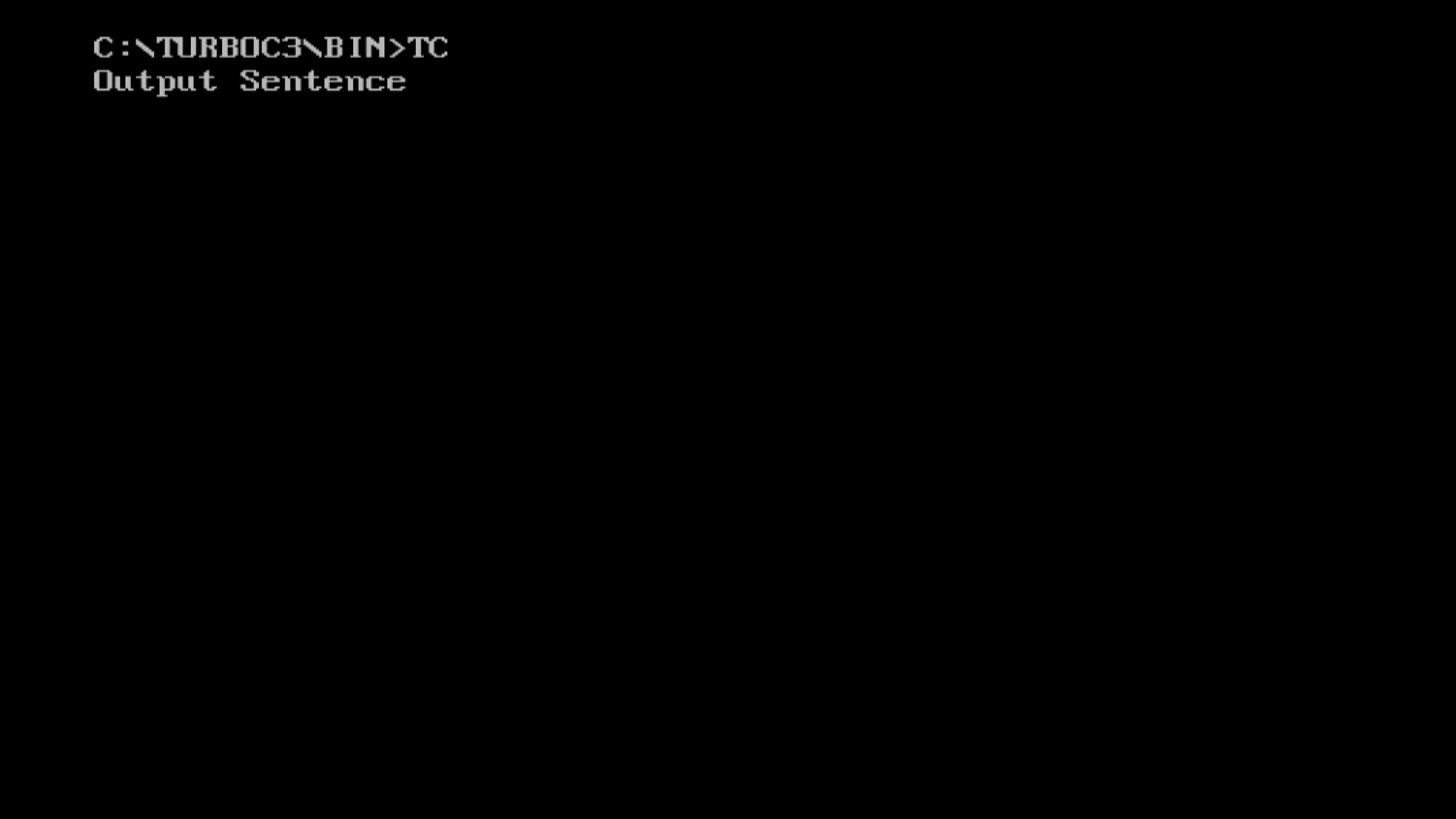
```
I am 24 years old and my zipcode is 90064
```

SOME WORKING EXAMPLES

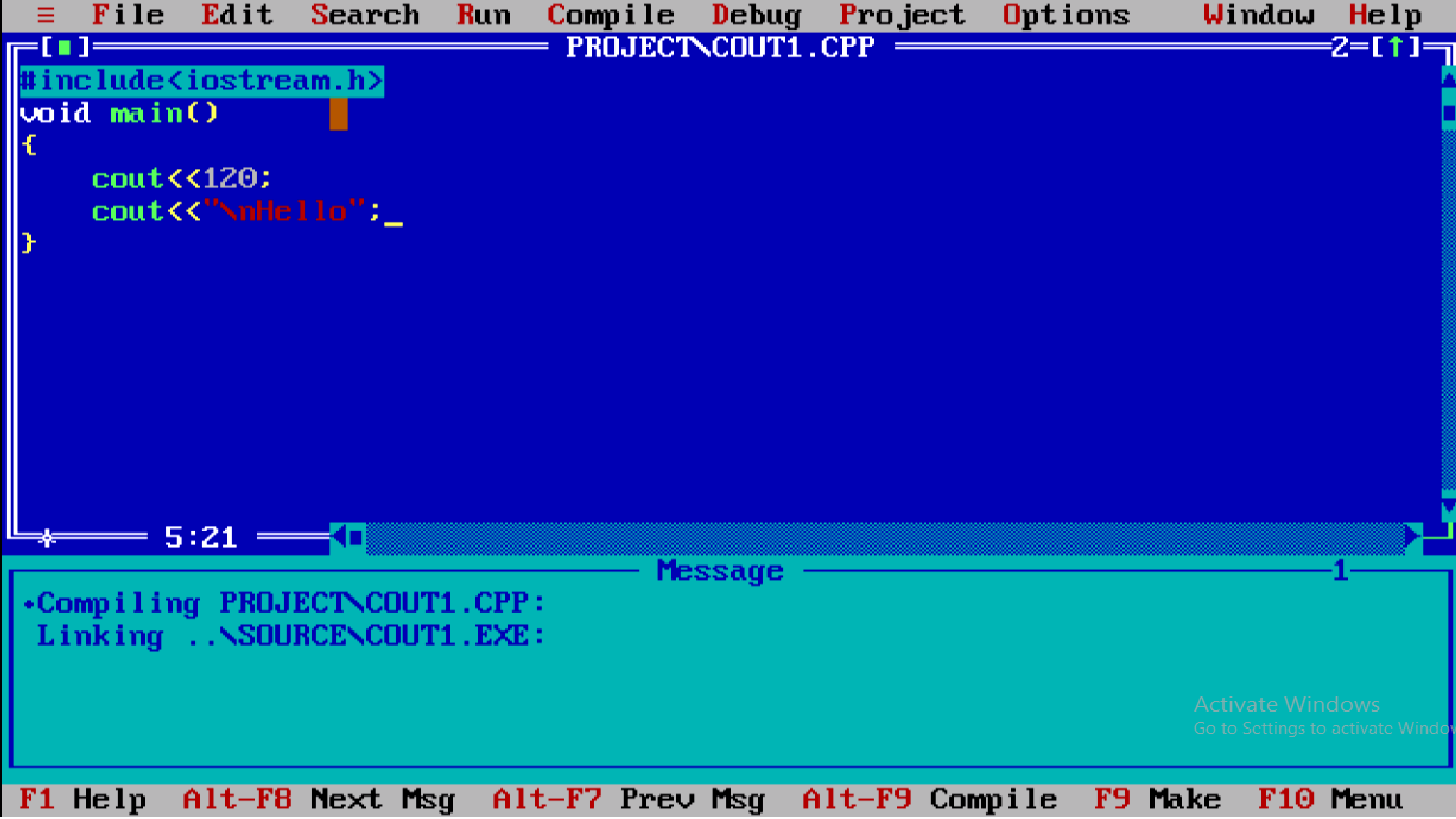


≡ File Edit Search Run Compile Debug Project Options Window Help
[■] NONAME00.CPP 2=[↕]
4:29
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9 Make F10 Menu

```
#include<iostream.h>
void main()
{
    cout<<"Output Sentence";
}
```

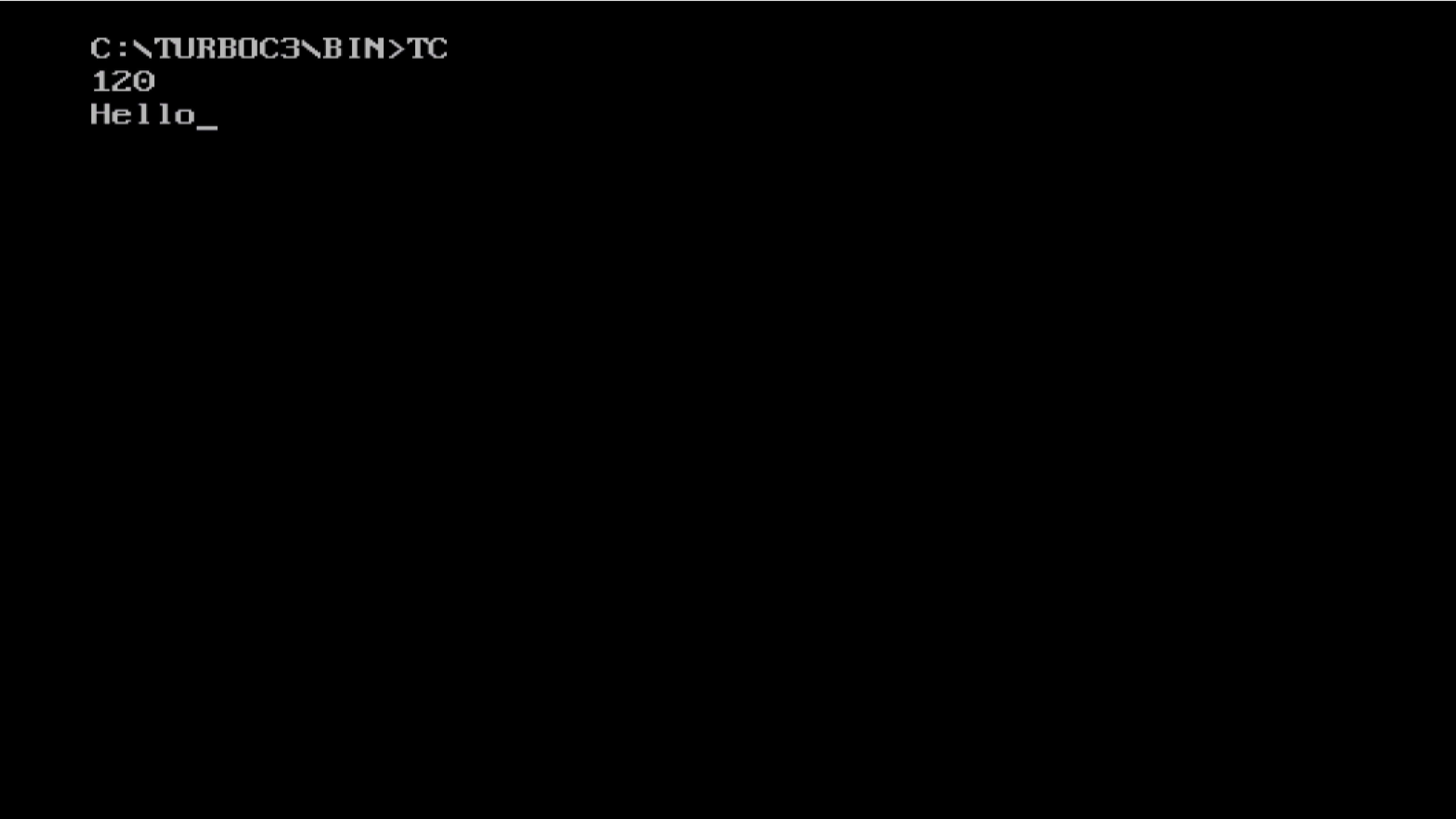


```
C:\TURBOC3\BIN>TC
Output Sentence
```

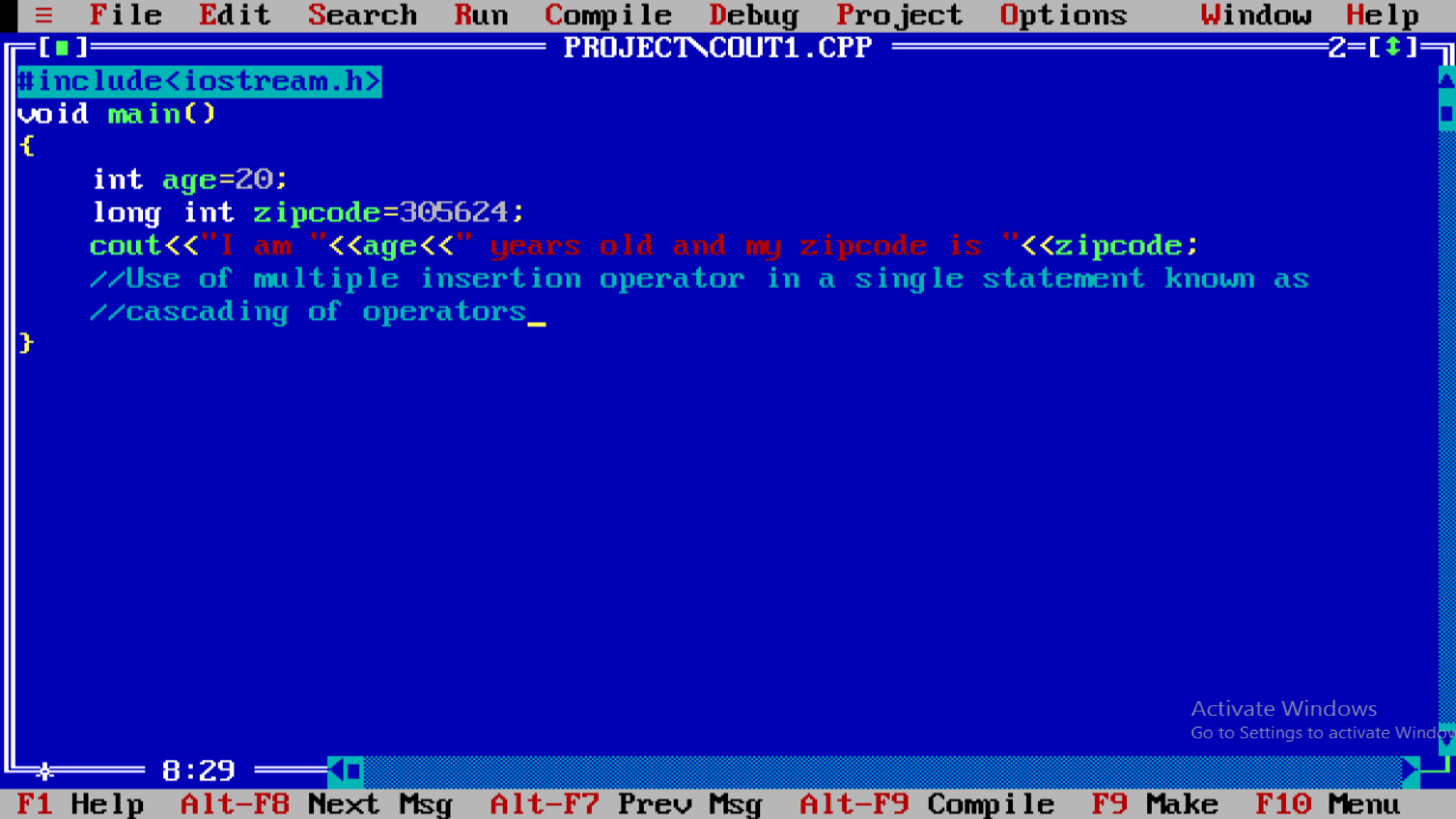

≡ File Edit Search Run Compile Debug Project Options Window Help
[■] PROJECT\COUT1.CPP 2 [↑]
5:21
Message 1
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9 Make F10 Menu
Activate Windows
Go to Settings to activate Windo

```
#include<iostream.h>
void main()
{
     cout<<120;
     cout<<"\nHello";
}
```

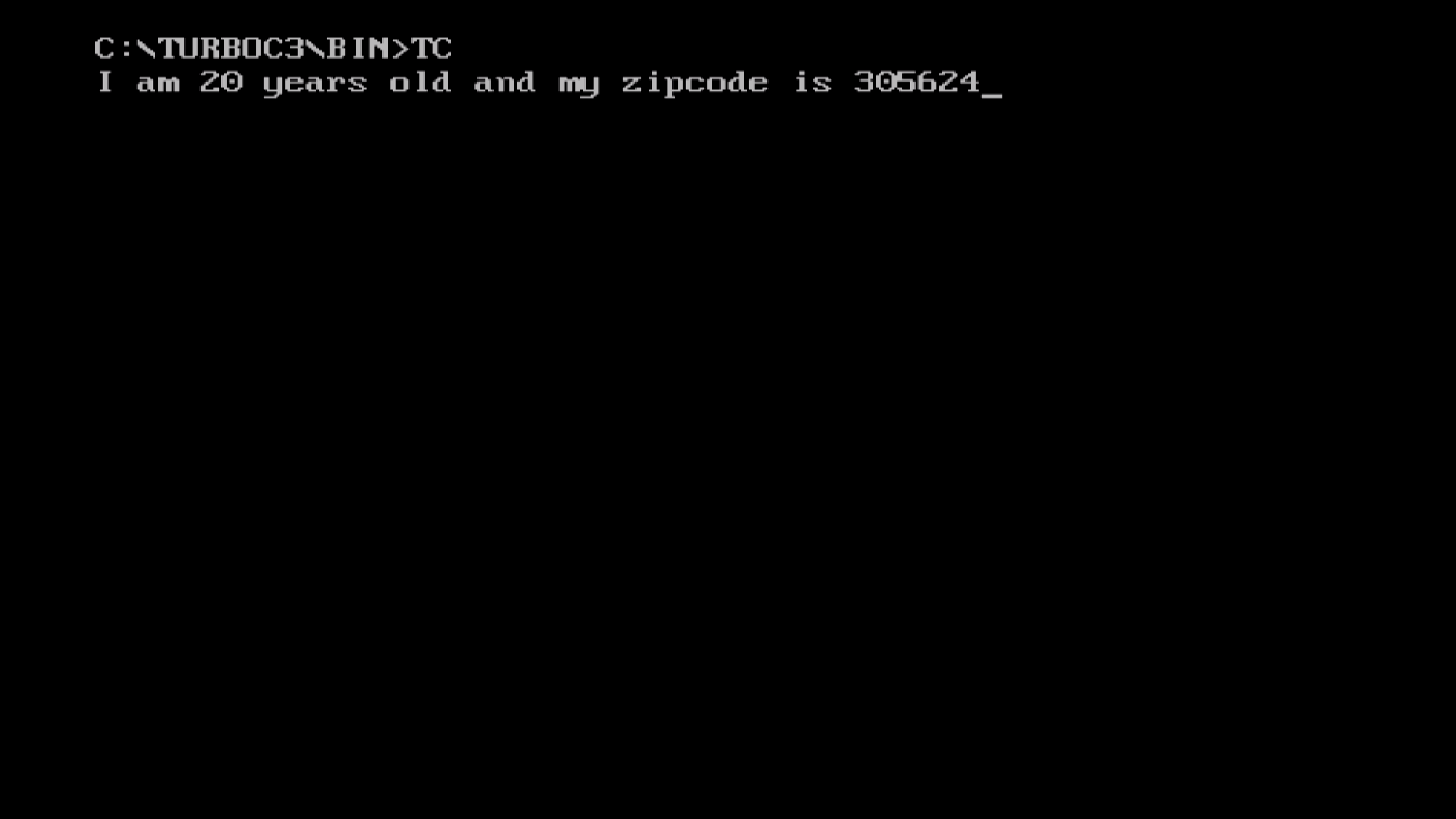
```
•Compiling PROJECT\COUT1.CPP:
 Linking ..\SOURCE\COUT1.EXE:
```



```
C:\TURBOC3\BIN>TC
120
Hello_
```


≡ File Edit Search Run Compile Debug Project Options Window Help
[■]
PROJECT\COUT1.CPP
2 [↕]
8:29
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9 Make F10 Menu

```
#include<iostream.h>
void main()
{
    int age=20;
    long int zipcode=305624;
    cout<<"I am "<<age<<" years old and my zipcode is "<<zipcode;
    //Use of multiple insertion operator in a single statement known as
    //cascading of operators
}
```



```
C:\TURBOC3\BIN>TC
I am 20 years old and my zipcode is 305624_
```

cin statement

This command is used to receive the input from the User.

For formatted input operations, **cin** is used together with the **extraction operator**, which is written as **>>** (i.e., two "greater than" signs). This operator is then followed by the variable where the extracted data is stored. For example:

```
int age;
cin>>age:
```

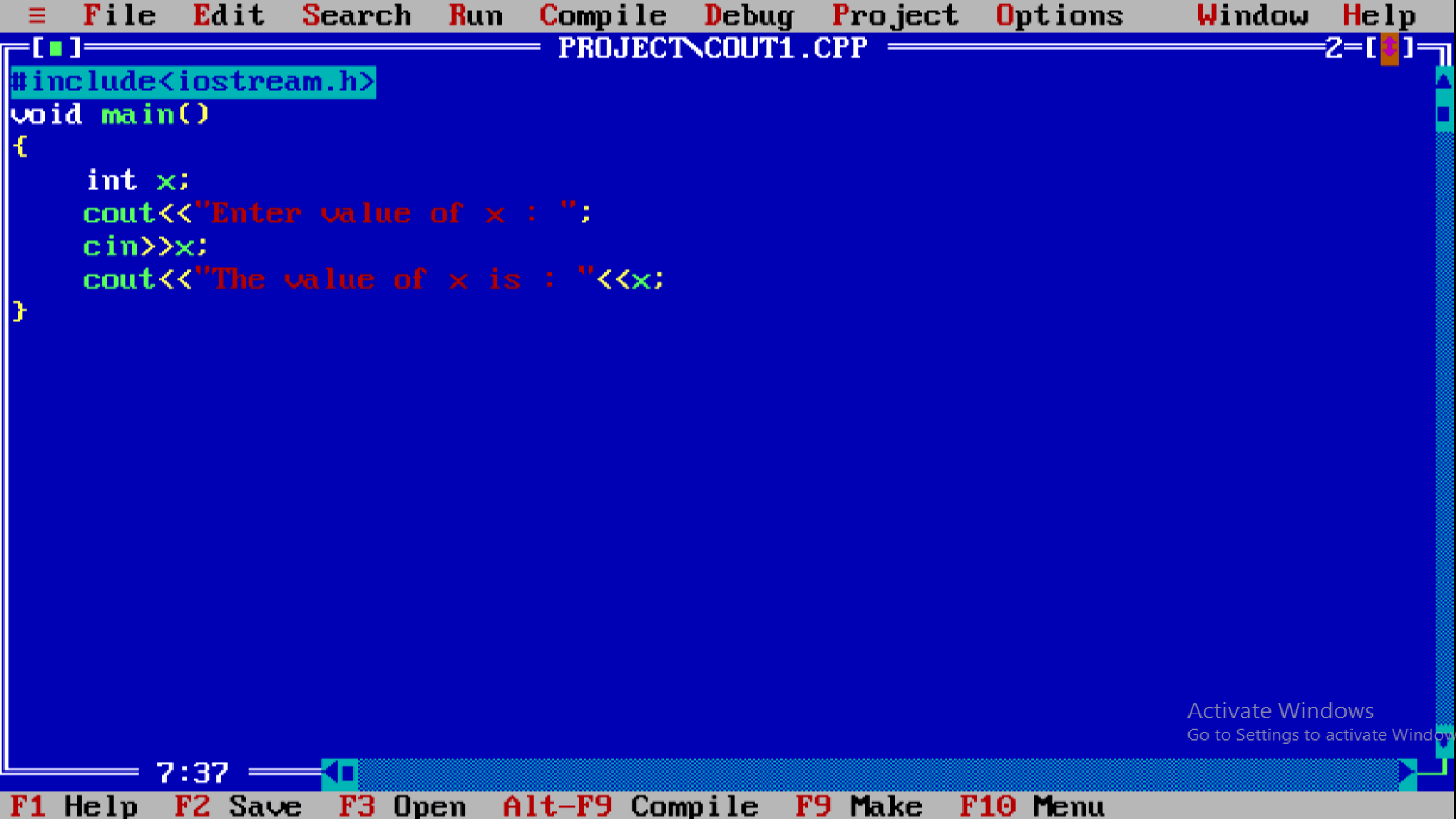
The second statement extracts from cin a value to be stored in it. Now the program will wait for the user to enter some sequence with the keyboard.

```
cin >> a;
cin >> b;
```

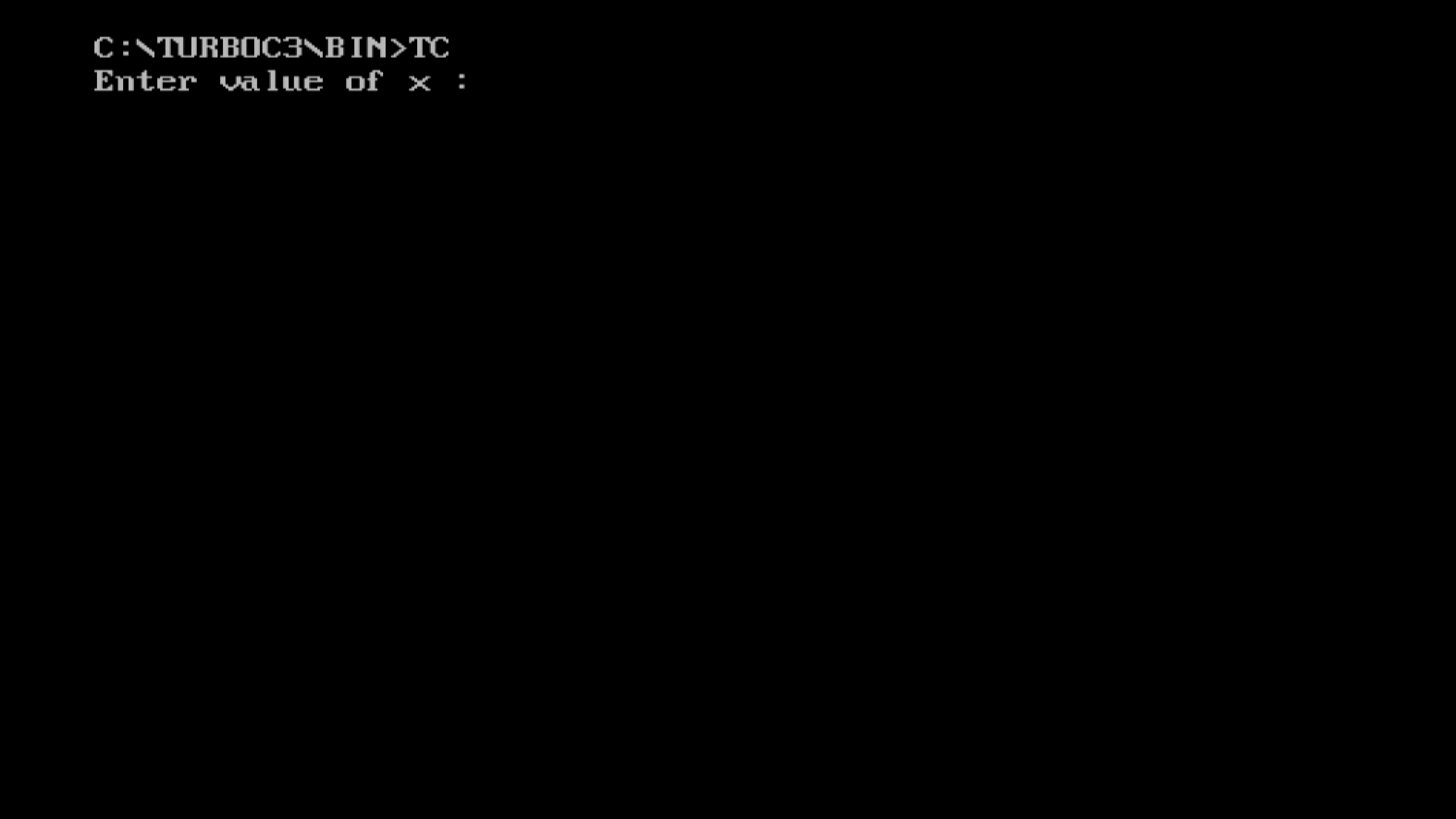
This is equivalent to

```
cin >> a>> b;
```

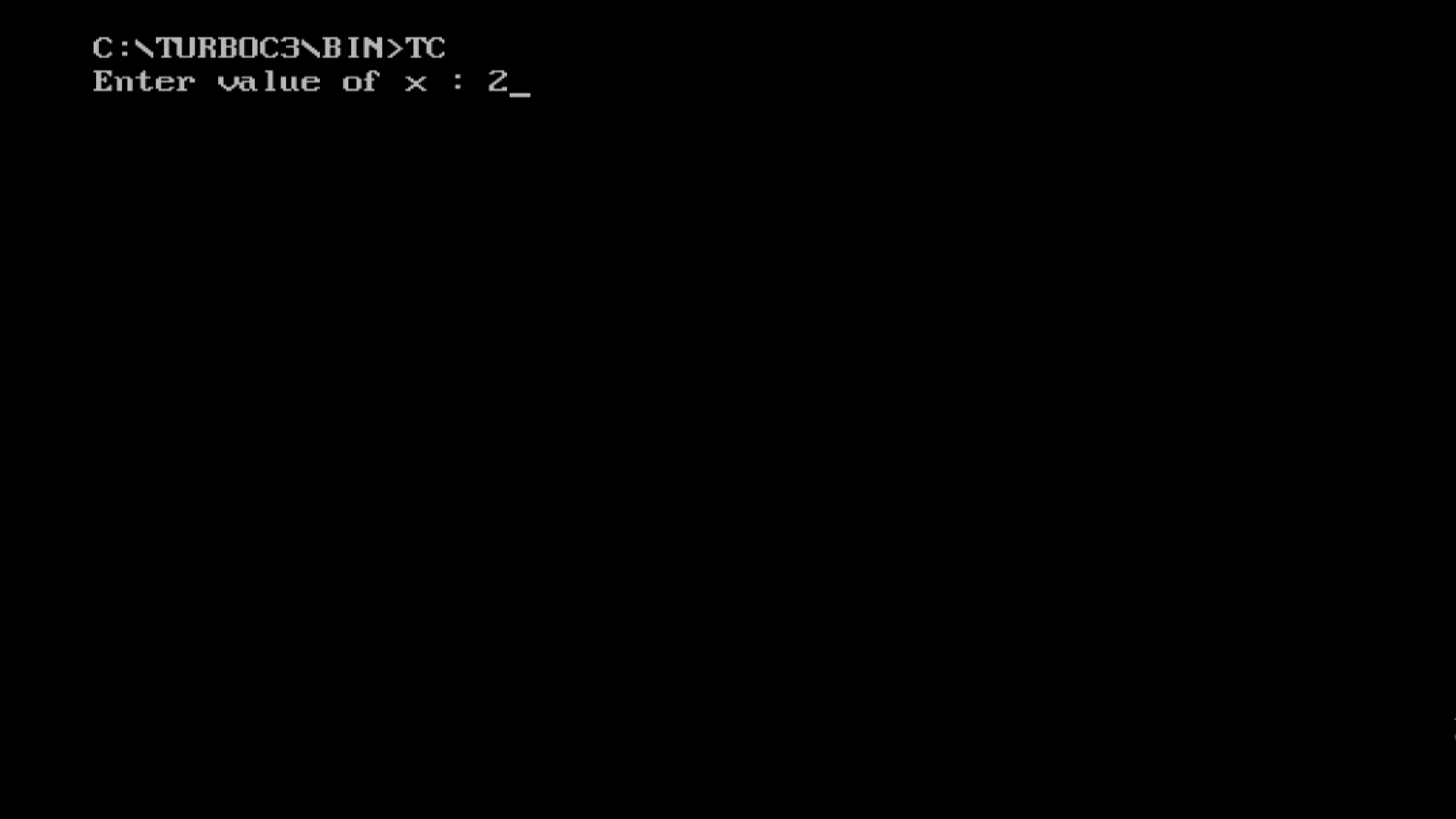
In both cases, the user is expected to introduce two values, one for variable a, and another for variable b. Any kind of space is used to separate two consecutive input operations; this may either be a space, a tab, or a new-line character.


≡ File Edit Search Run Compile Debug Project Options Window Help
[■]
PROJECT\COUT1.CPP
2
7:37
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu

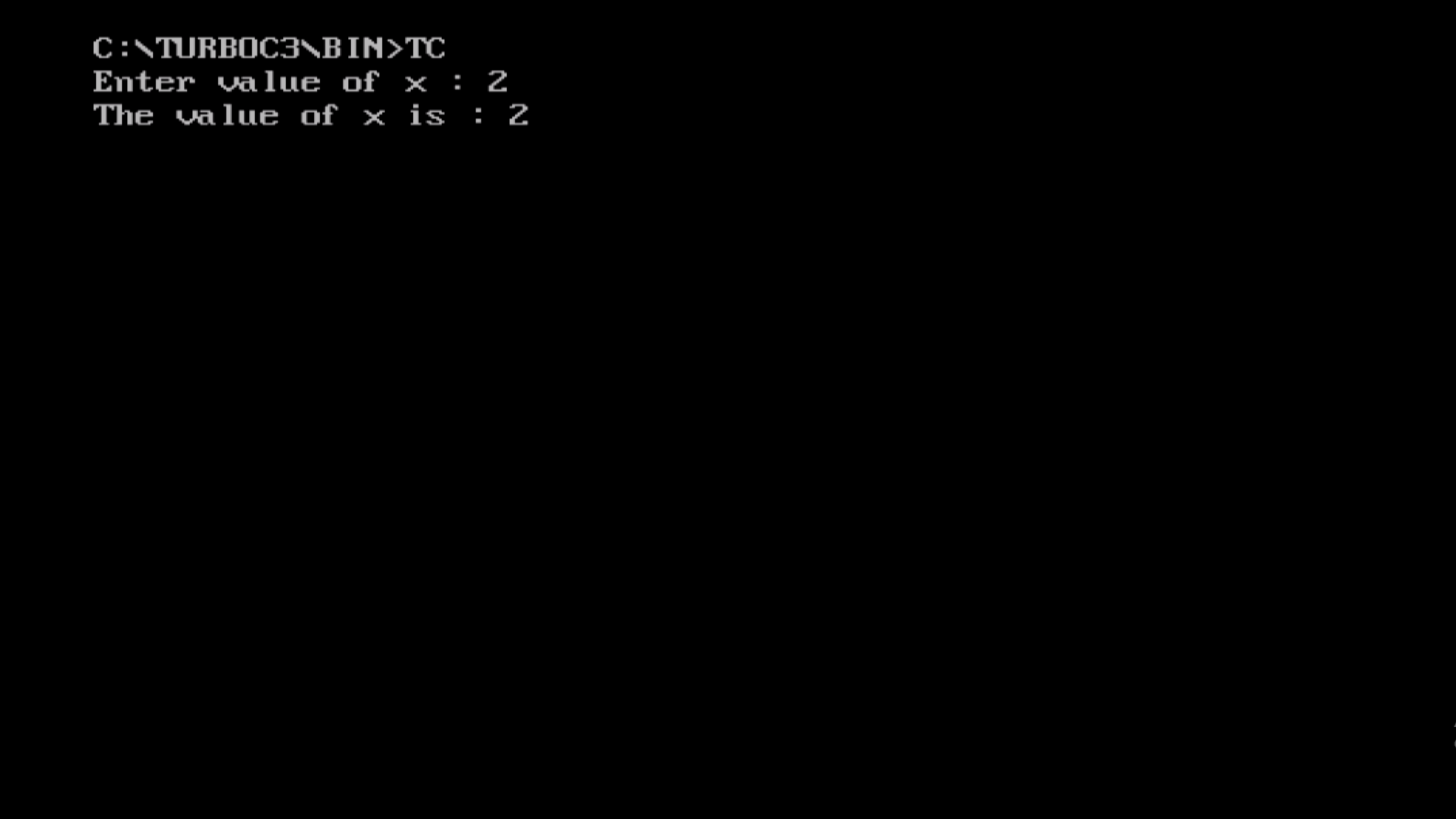
```
#include<iostream.h>
void main()
{
    int x;
    cout<<"Enter value of x : ";
    cin>>x;
    cout<<"The value of x is : "<<x;
}
```



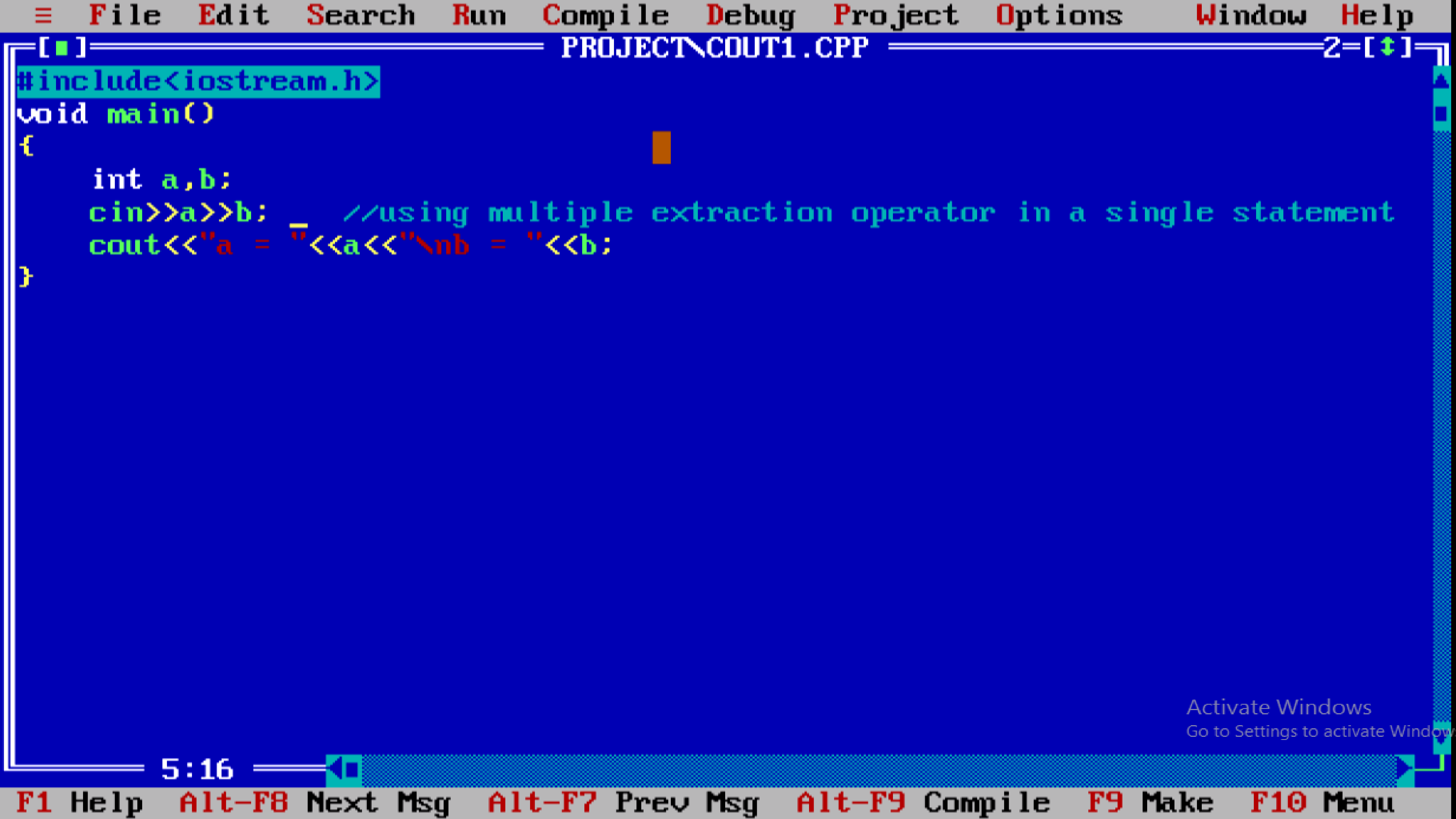
```
C:\TURBOC3\BIN>TC
Enter value of x :
```



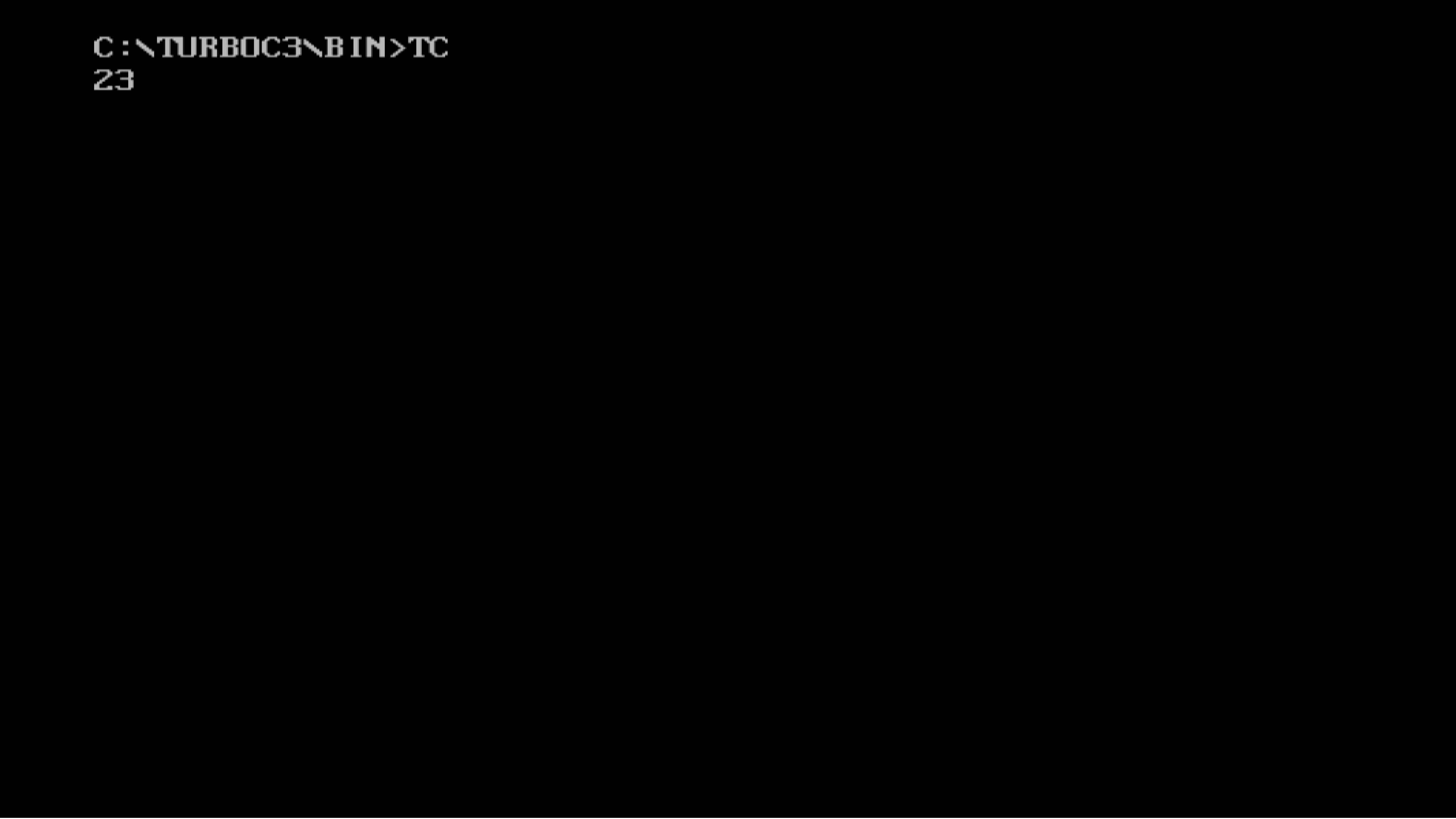
```
C:\TURBOC3\BIN>TC
Enter value of x : 2_
```



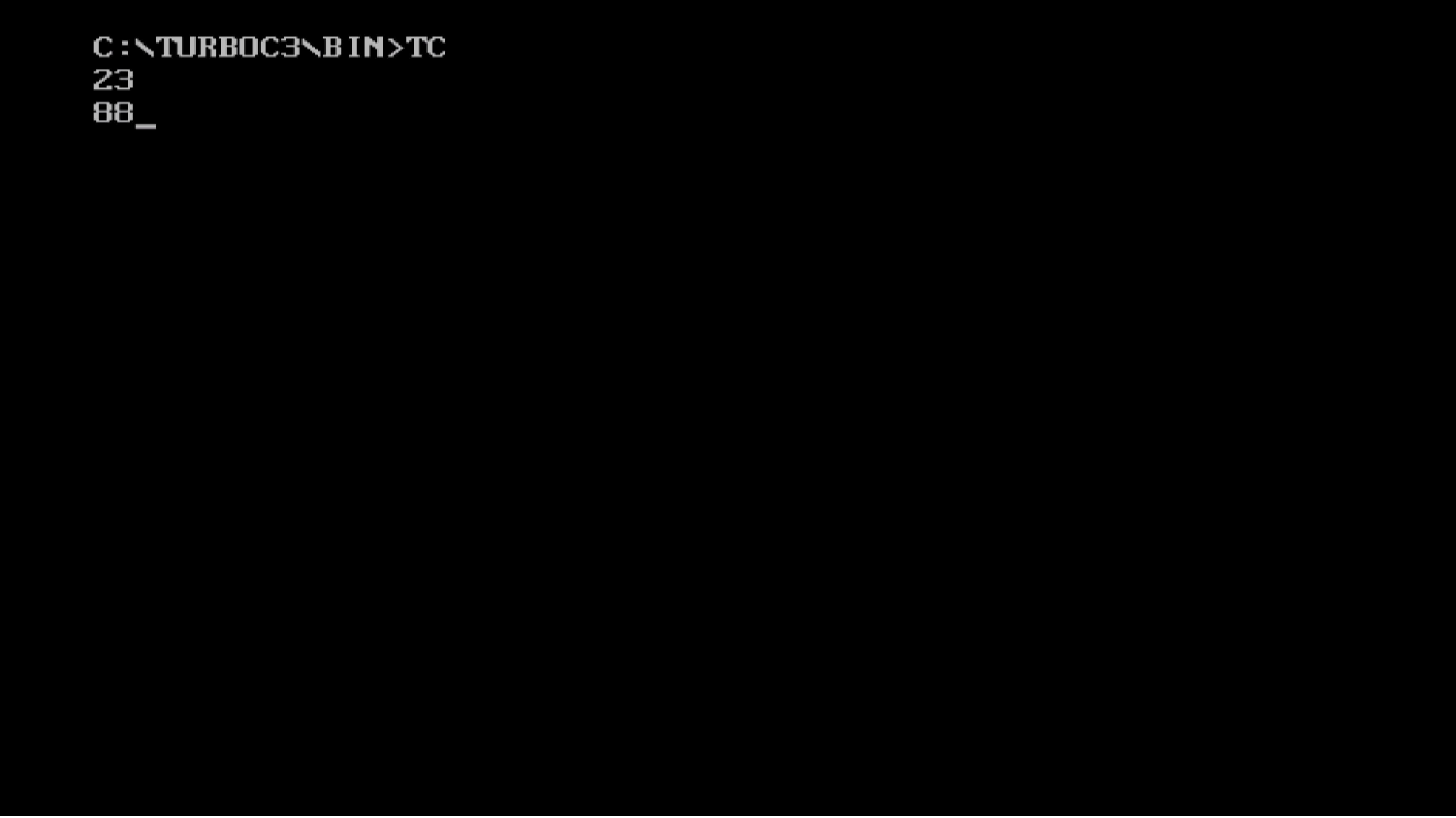
```
C:\TURBOC3\BIN>TC
Enter value of x : 2
The value of x is : 2
```


≡ File Edit Search Run Compile Debug Project Options Window Help
[■] PROJECT\COUT1.CPP 2 [↕]
5:16
Activate Windows
Go to Settings to activate Windows
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9 Make F10 Menu

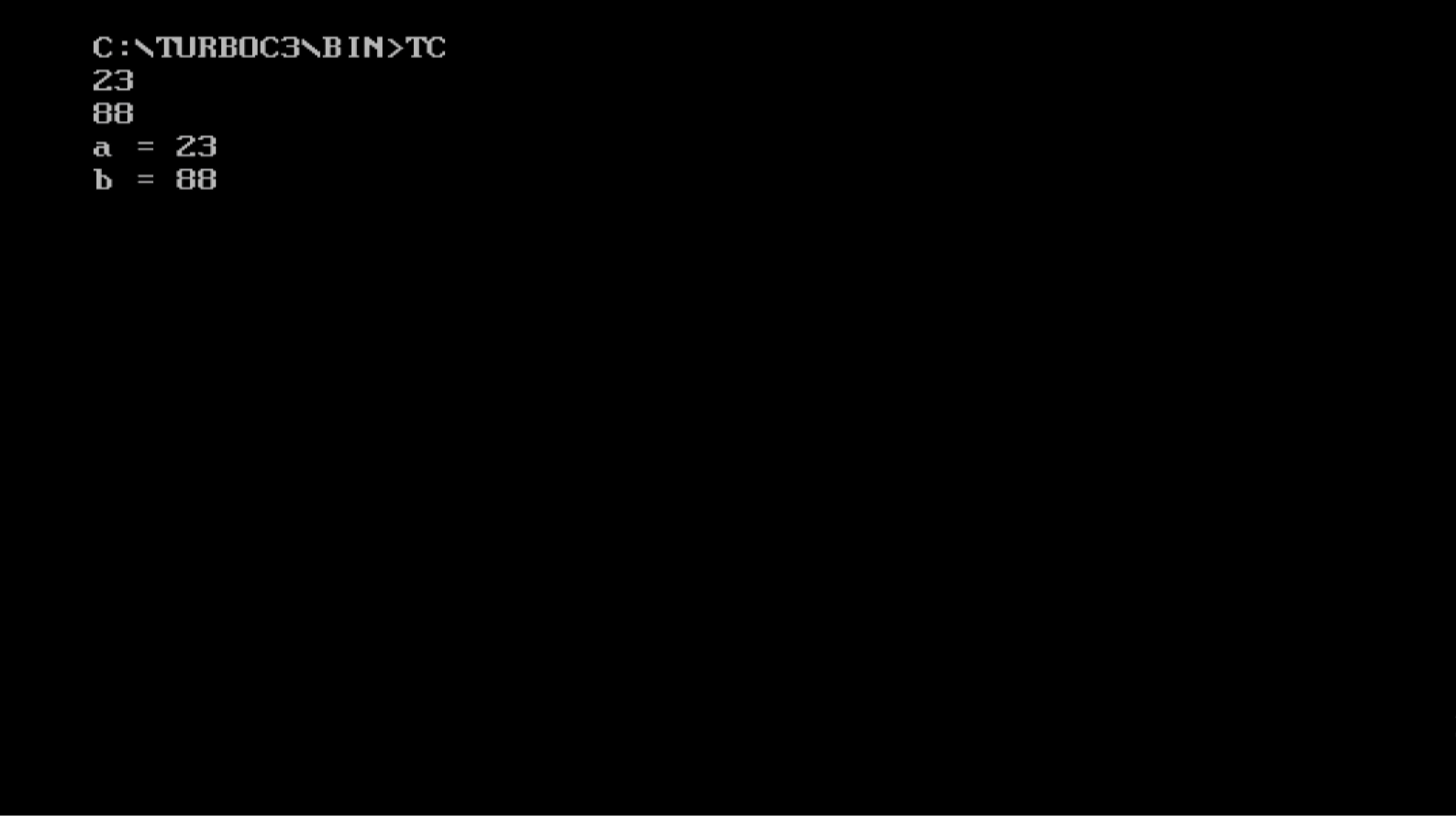
```
#include<iostream.h>
void main()
{
	int a,b;
	cin>>a>>b;	//using multiple extraction operator in a single statement
	cout<<"a = "<<a<<"\nb = "<<b;
}
```



```
C:\TURBOC3\BIN>TC
23
```



```
C:\TURBOC3\BIN>TC
23
88
```



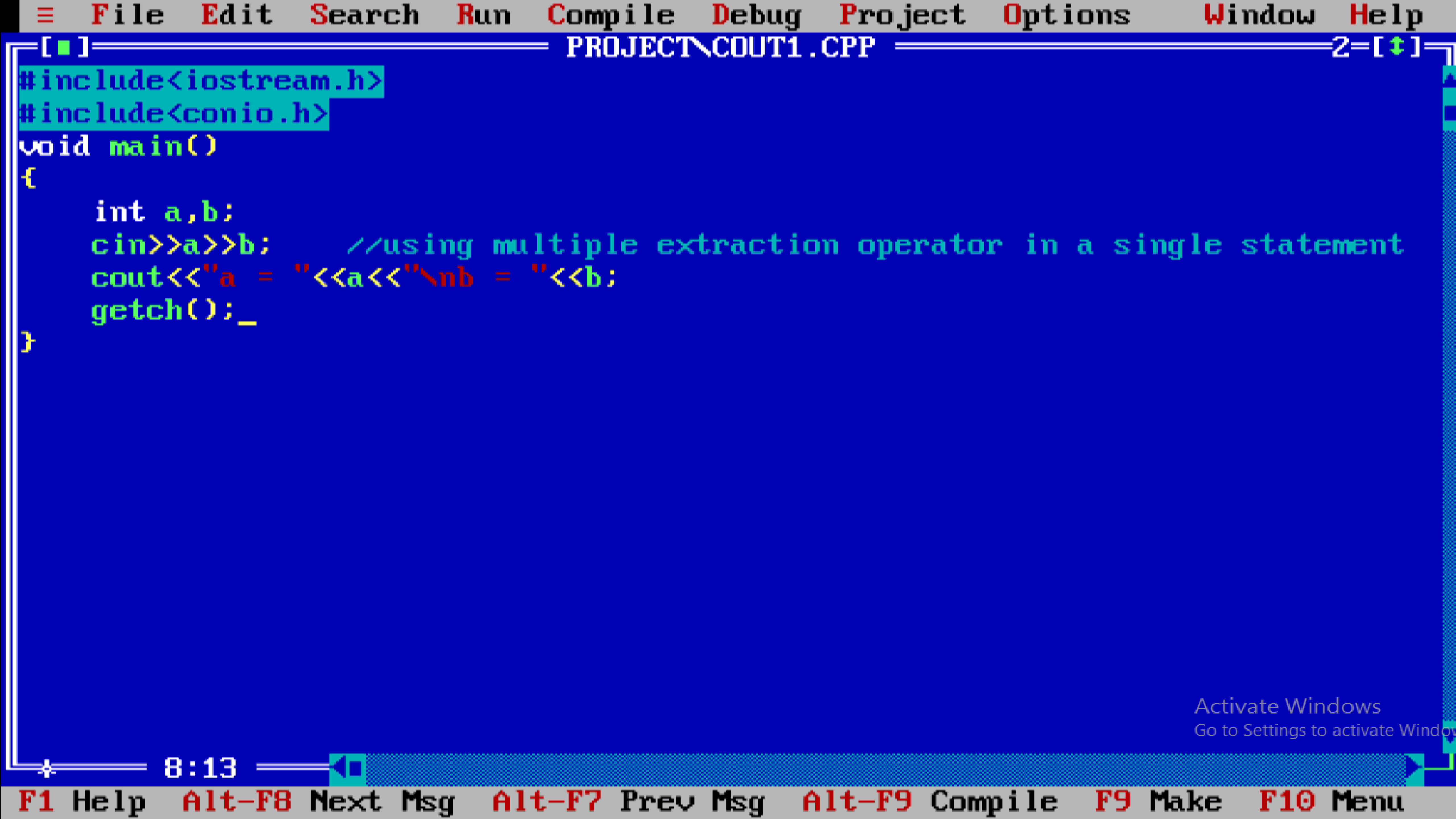
```
C:\TURBOC3\BIN>TC
23
88
a = 23
b = 88
```

getch() statement

It is a predefined function in "conio.h" (console input output header file), It will tell the console wait for some time until a key is pressed after the complete running of the program.

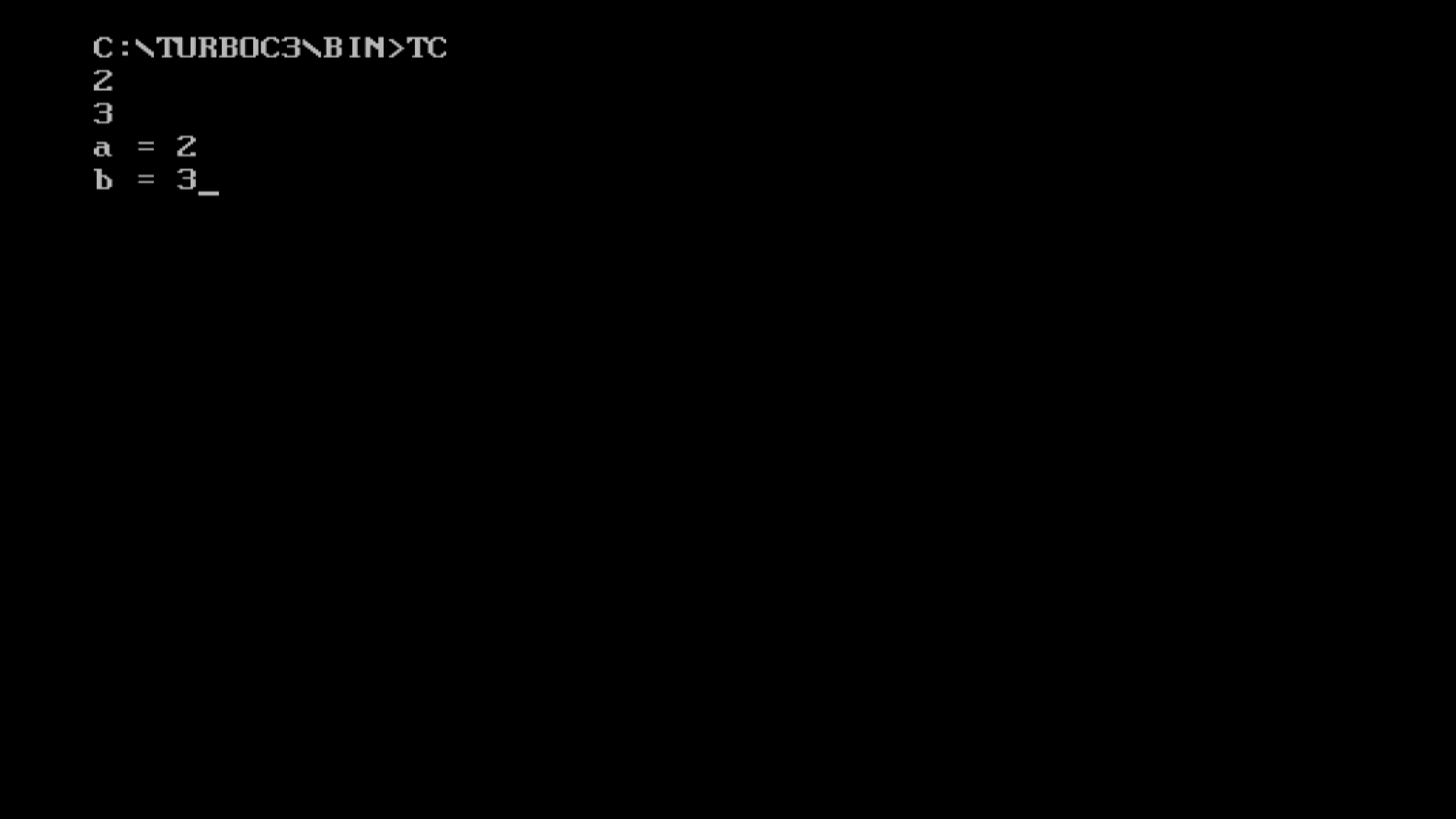
By using this function we can read a character directly from the keyboard.

getch() is used to directly see the output without the change of screen.


≡ File Edit Search Run Compile Debug Project Options Window Help
[■] PROJECT\COUT1.CPP 2 [↕]
8:13
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9 Make F10 Menu
Activate Windows
Go to Settings to activate Windo

```
#include<iostream.h>
#include<conio.h>
void main()
{
    int a,b;
    cin>>a>>b;    //using multiple extraction operator in a single statement
    cout<<"a = "<<a<<"\nb = "<<b;
    getch();_
}
```

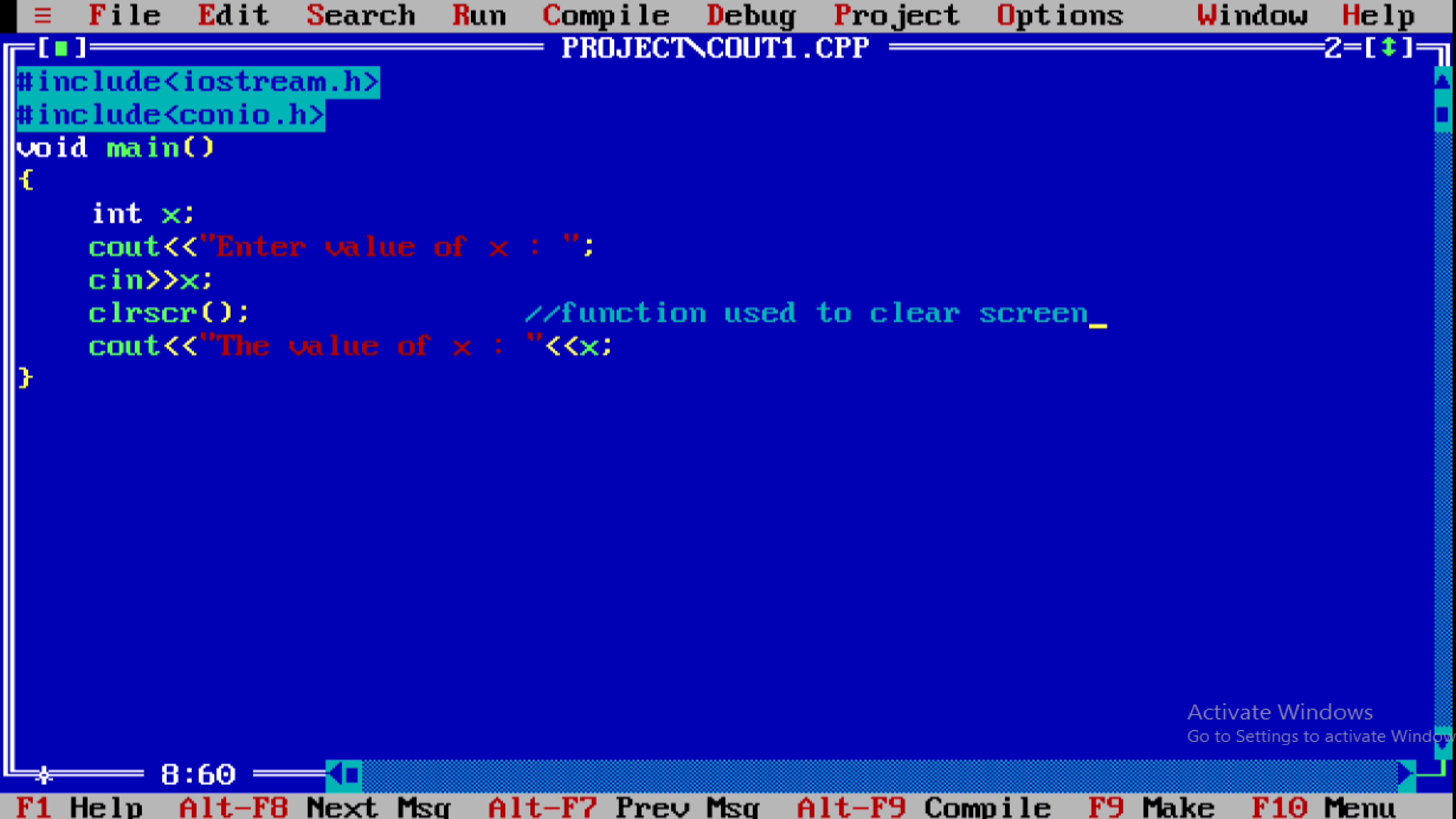
```
C:\TURBOC3\BIN>TC
2
3
```



```
C:\TURBOC3\BIN>TC
2
3
a = 2
b = 3_
```

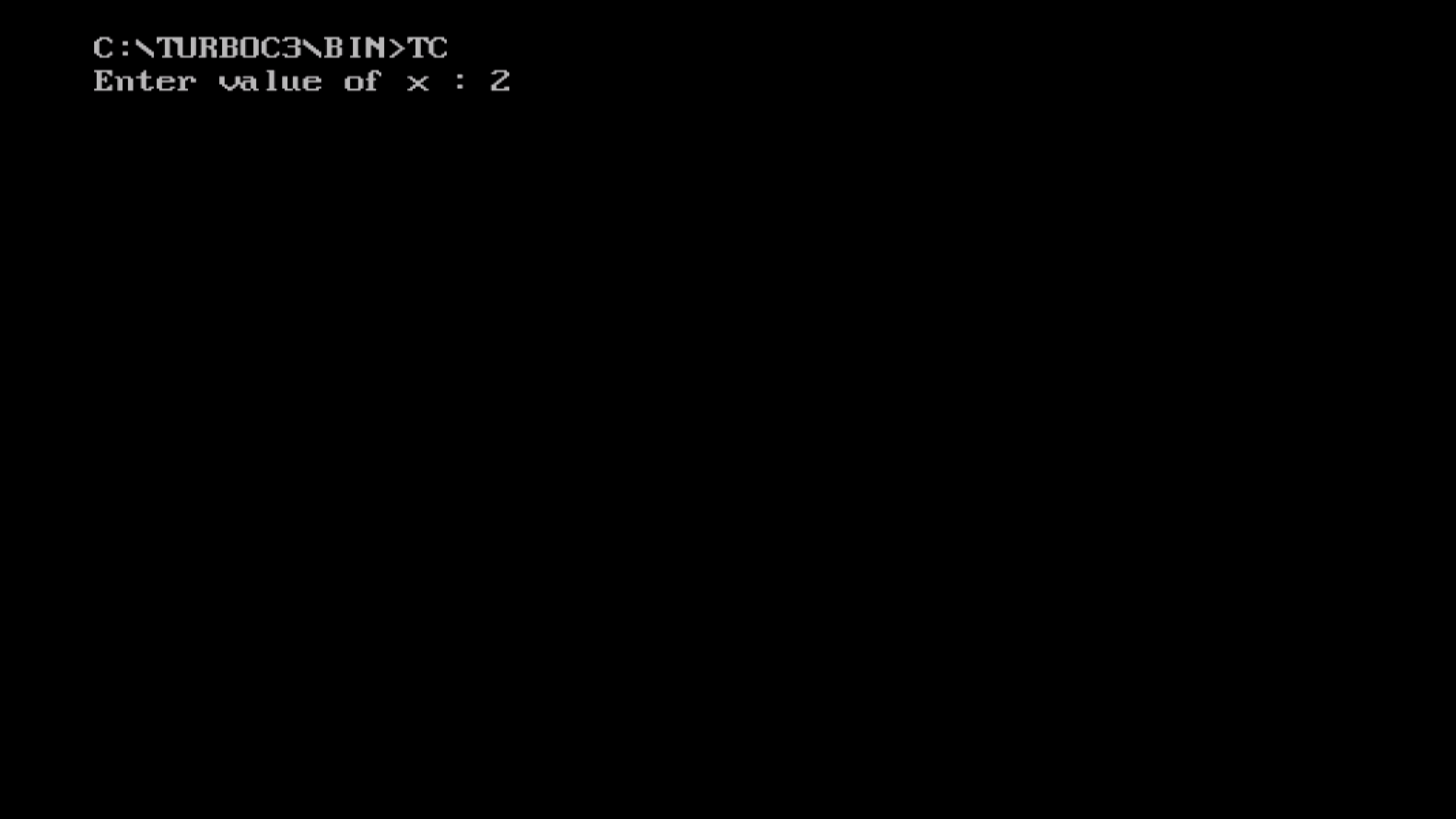
clrscr statement

It is a predefined function in "conio.h" (console input output header file) used to clear the console screen.

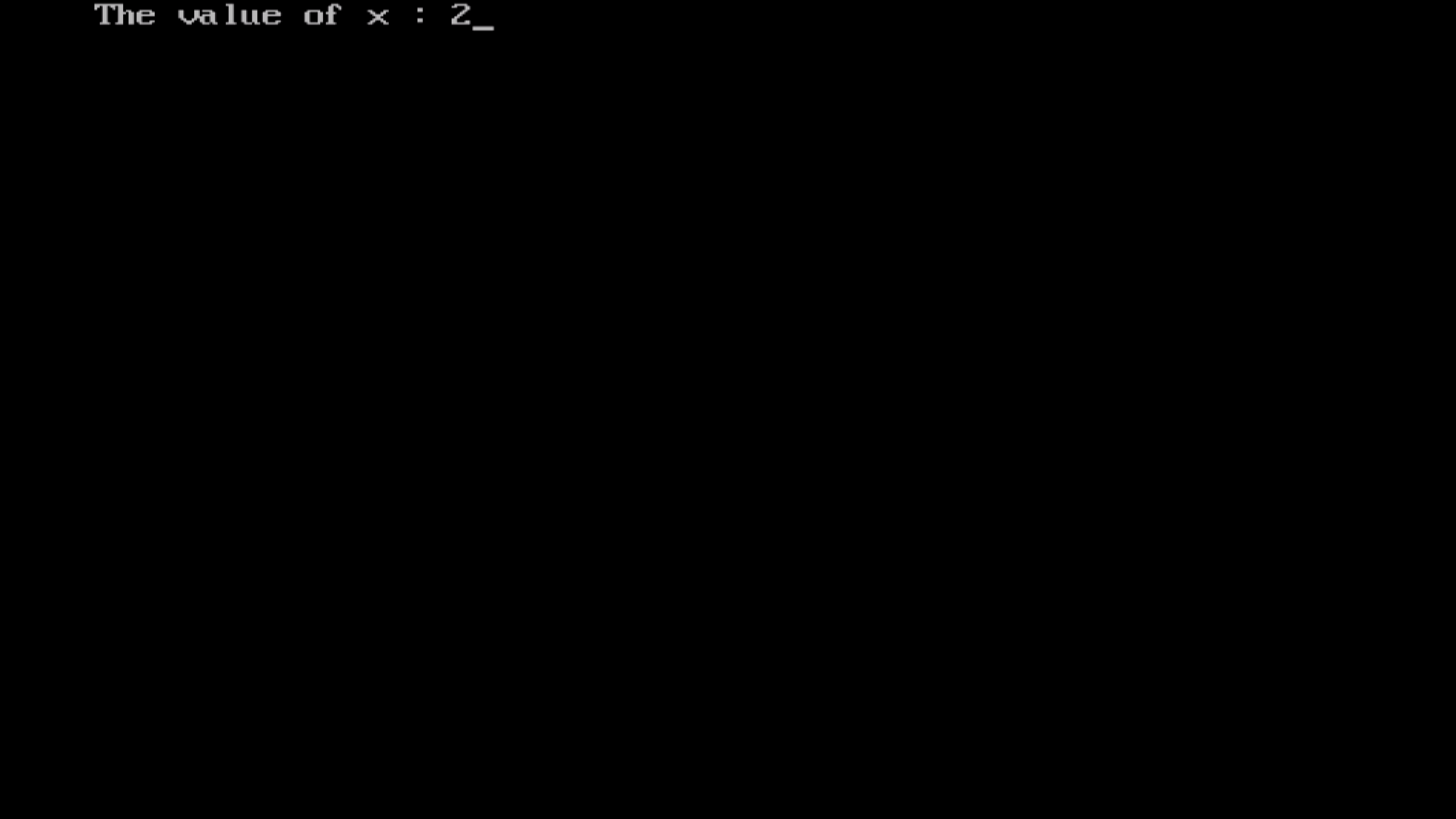

≡ File Edit Search Run Compile Debug Project Options Window Help
[■]
PROJECT\COUT1.CPP
2
[↕]

```
#include<iostream.h>
#include<conio.h>
void main()
{
    int x;
    cout<<"Enter value of x : ";
    cin>>x;
    clrscr();                //function used to clear screen
    cout<<"The value of x : "<<x;
}
```

8:60
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9 Make F10 Menu



```
C:\TURBOC3\BIN>TC
Enter value of x : 2
```



```
The value of x : 2_
```

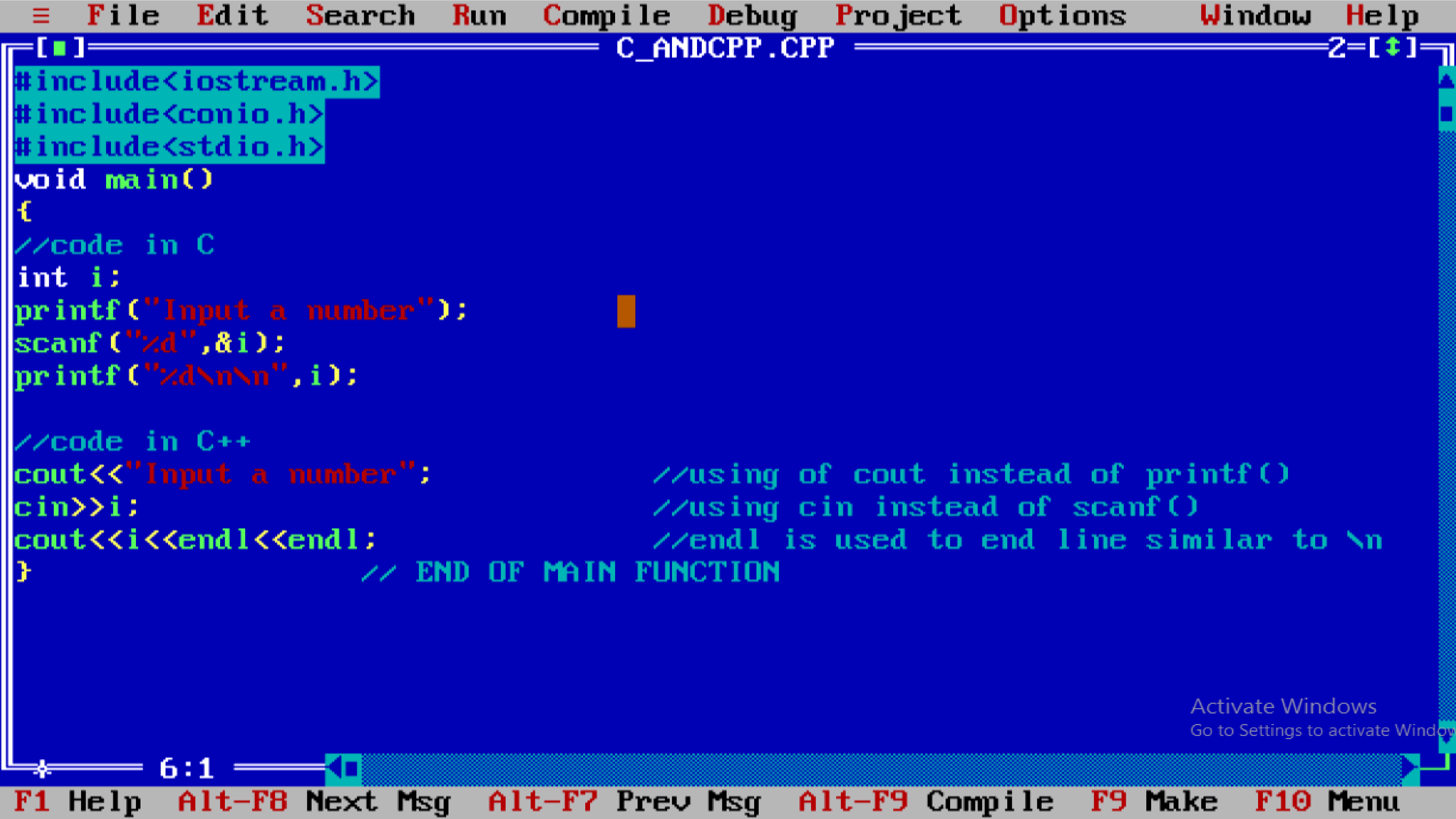
Some other differences between C and C++

C requires all the variables to be defined at the starting of a scope or a block while C++ allows the declaration of variable anywhere in the scope i.e. at time of its First use.

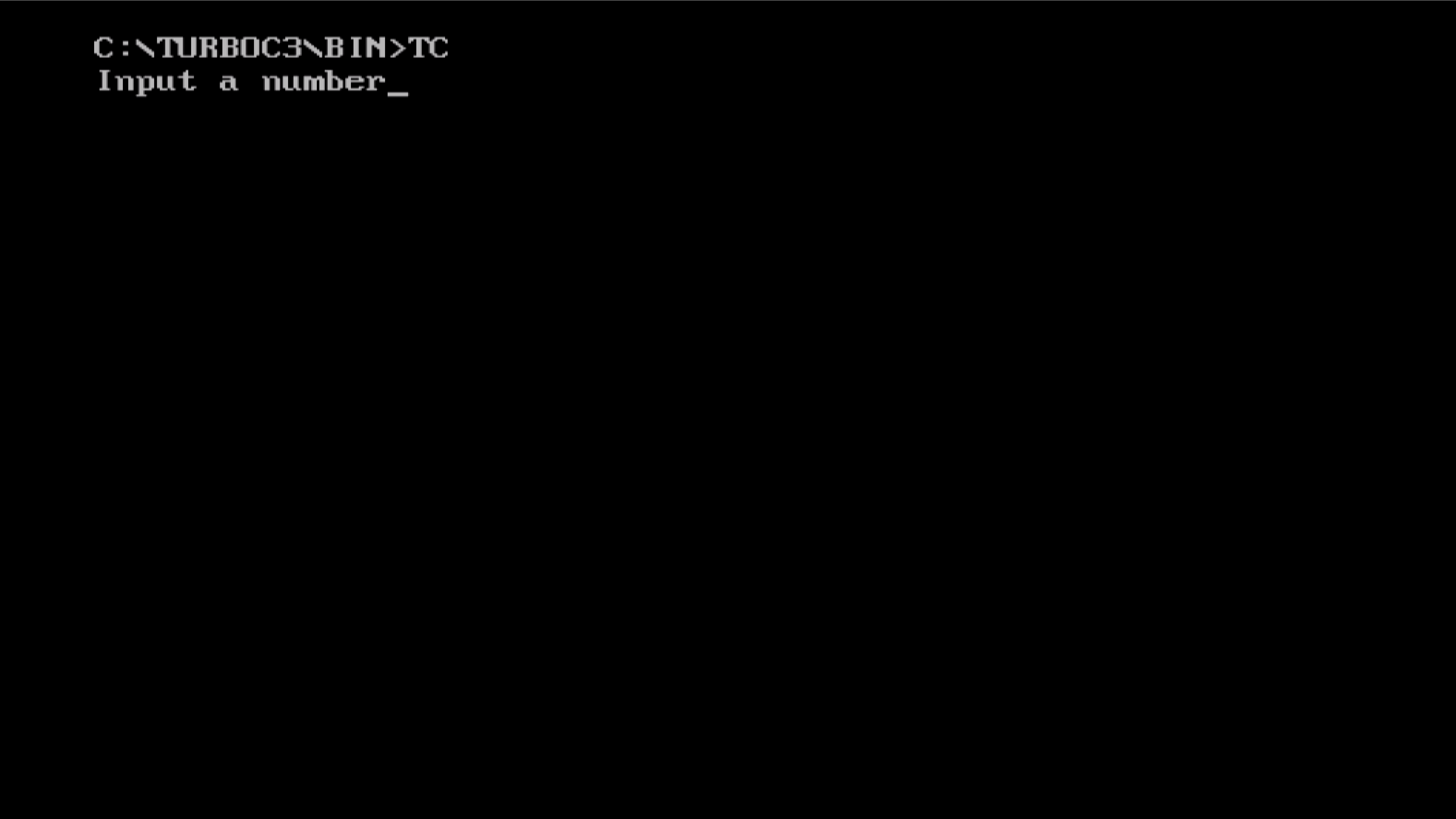
C++ supports both built-in and user defined data types but **C** supports built-in and primitive data types

You can use structs without writing struct before every declaration or using typedefs.

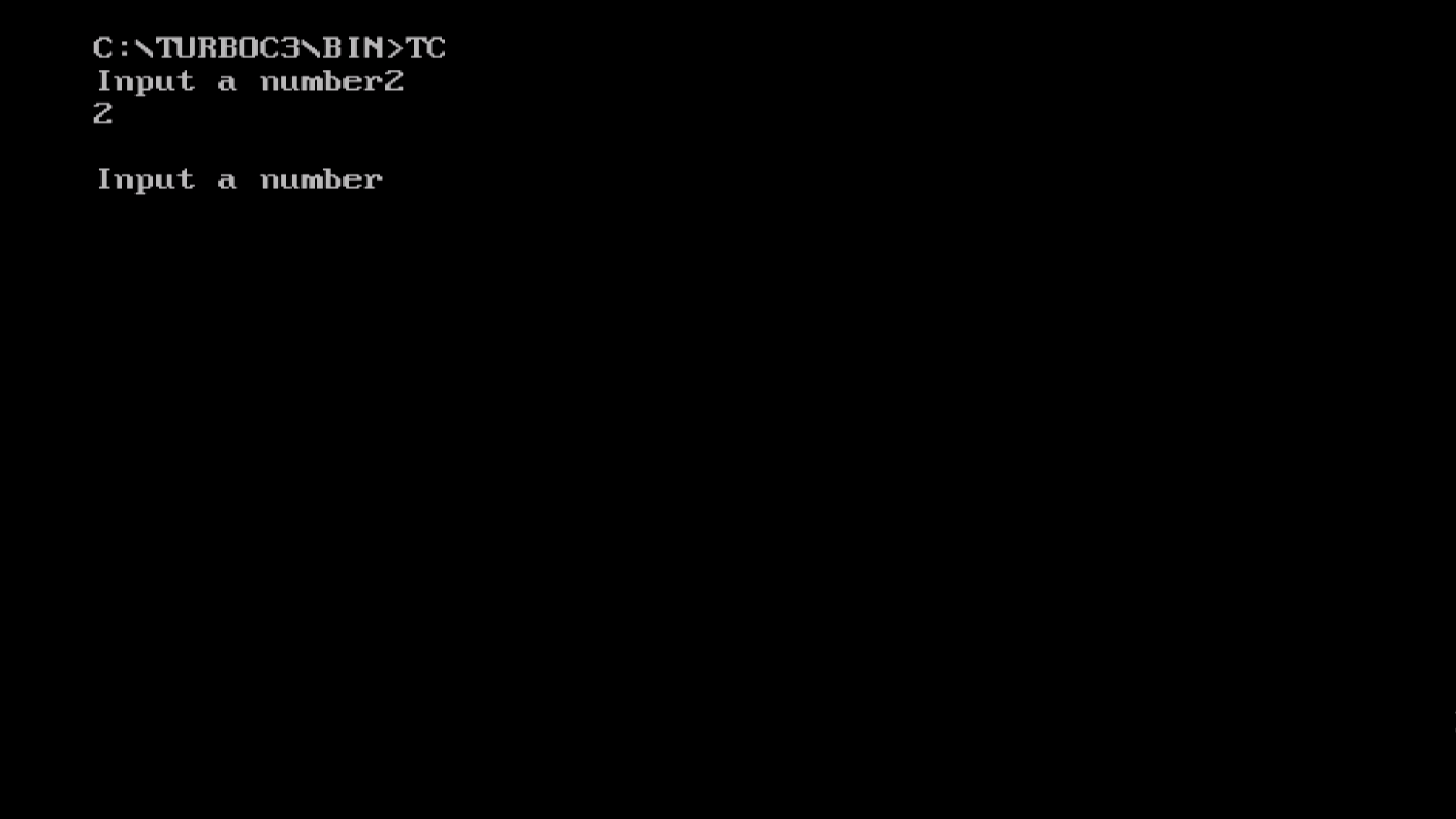
STL(Standard Template Library) is also a feature of C++.


≡ File Edit Search Run Compile Debug Project Options Window Help
[■]
C_ANDCPP.CPP
2=[↕]
6:1
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9 Make F10 Menu

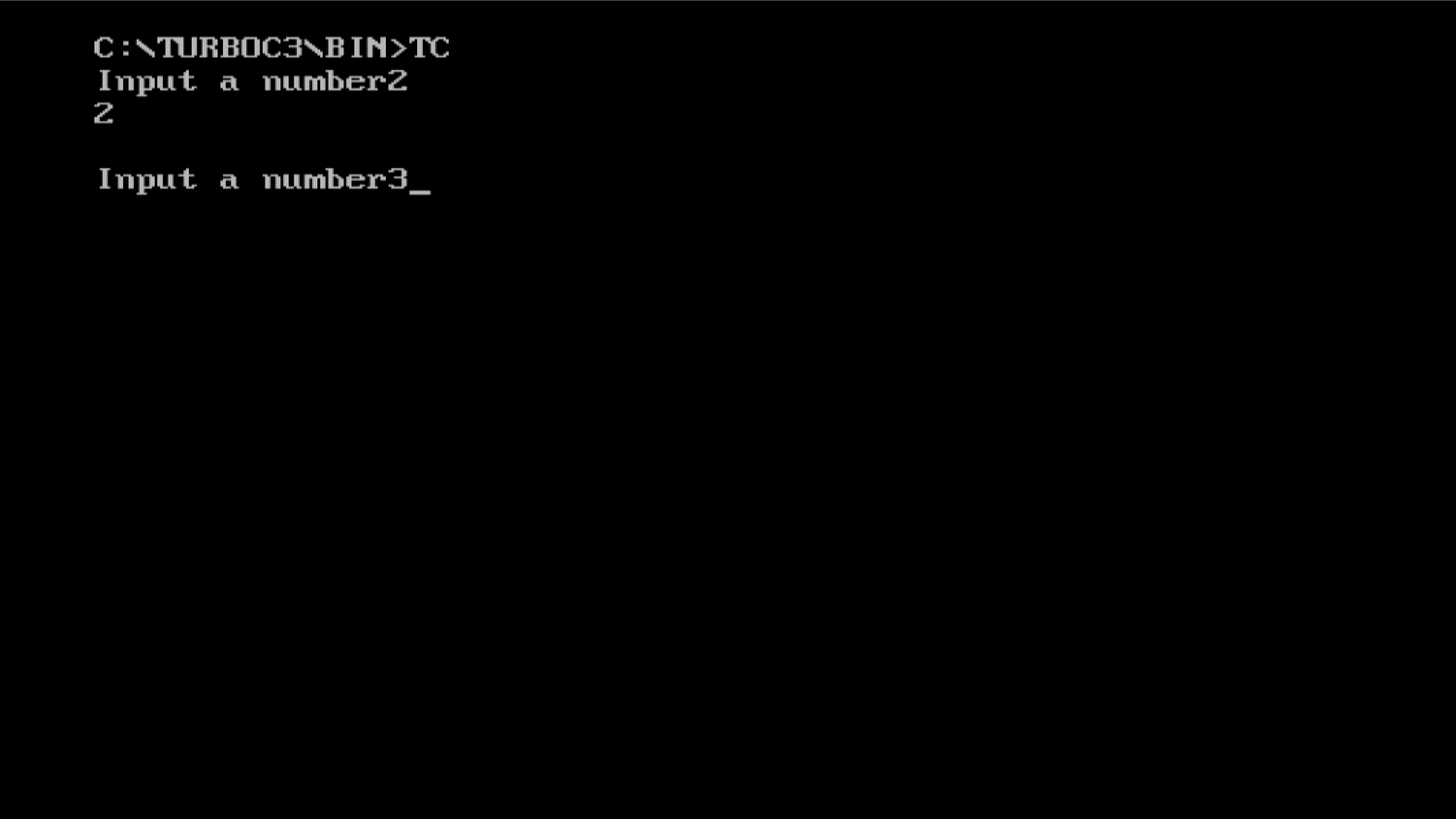
```
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
void main()
{
//code in C
int i;
printf("Input a number");
scanf("%d",&i);
printf("%d\n\n",i);

//code in C++
cout<<"Input a number";            //using of cout instead of printf()
cin>>i;                            //using cin instead of scanf()
cout<<i<<endl<<endl;               //endl is used to end line similar to \n
}                  // END OF MAIN FUNCTION
```



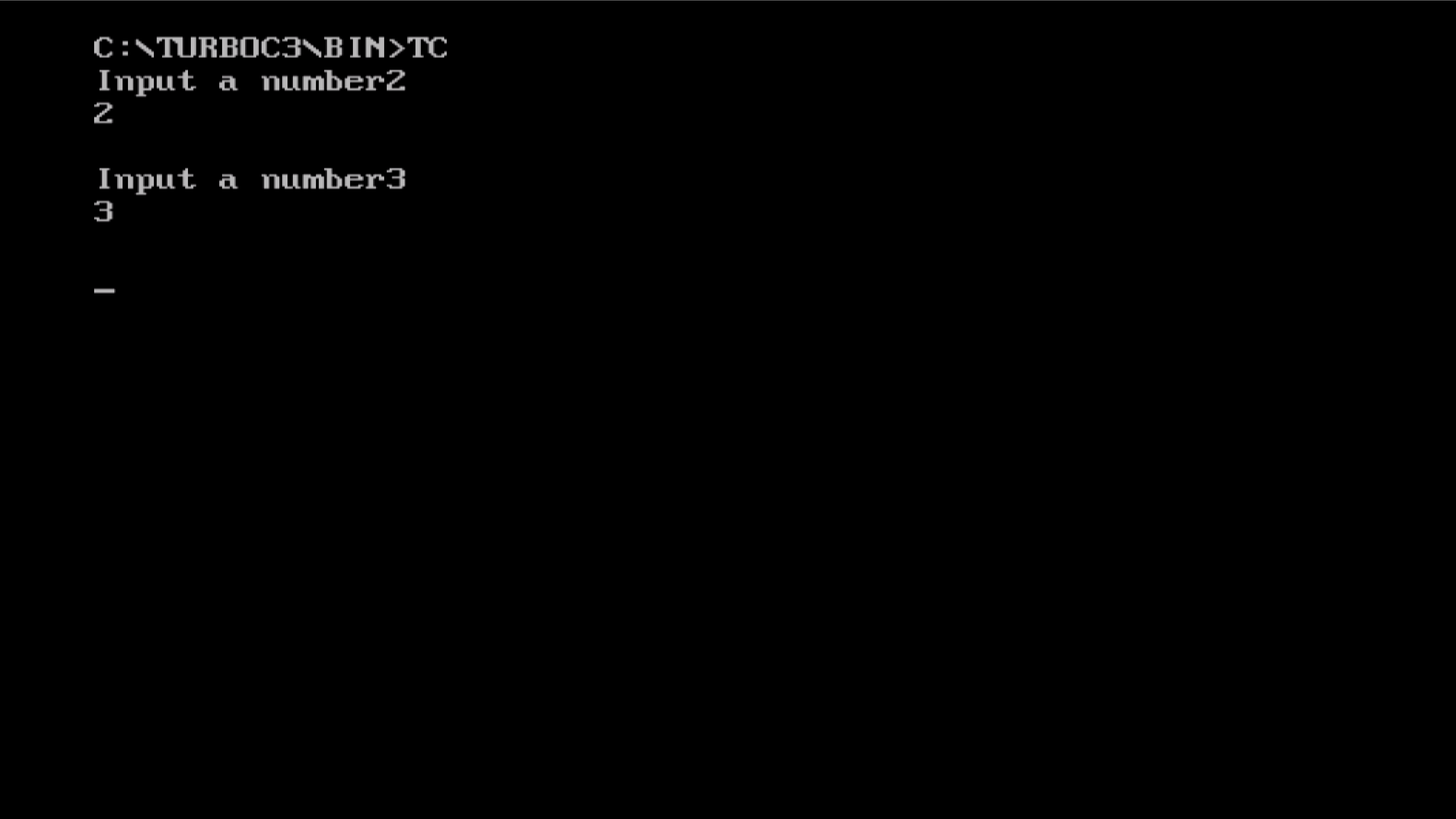
```
C:\TURBOC3\BIN>TC
Input a number_
```



```
C:\TURBOC3\BIN>TC
Input a number2
2

Input a number
```

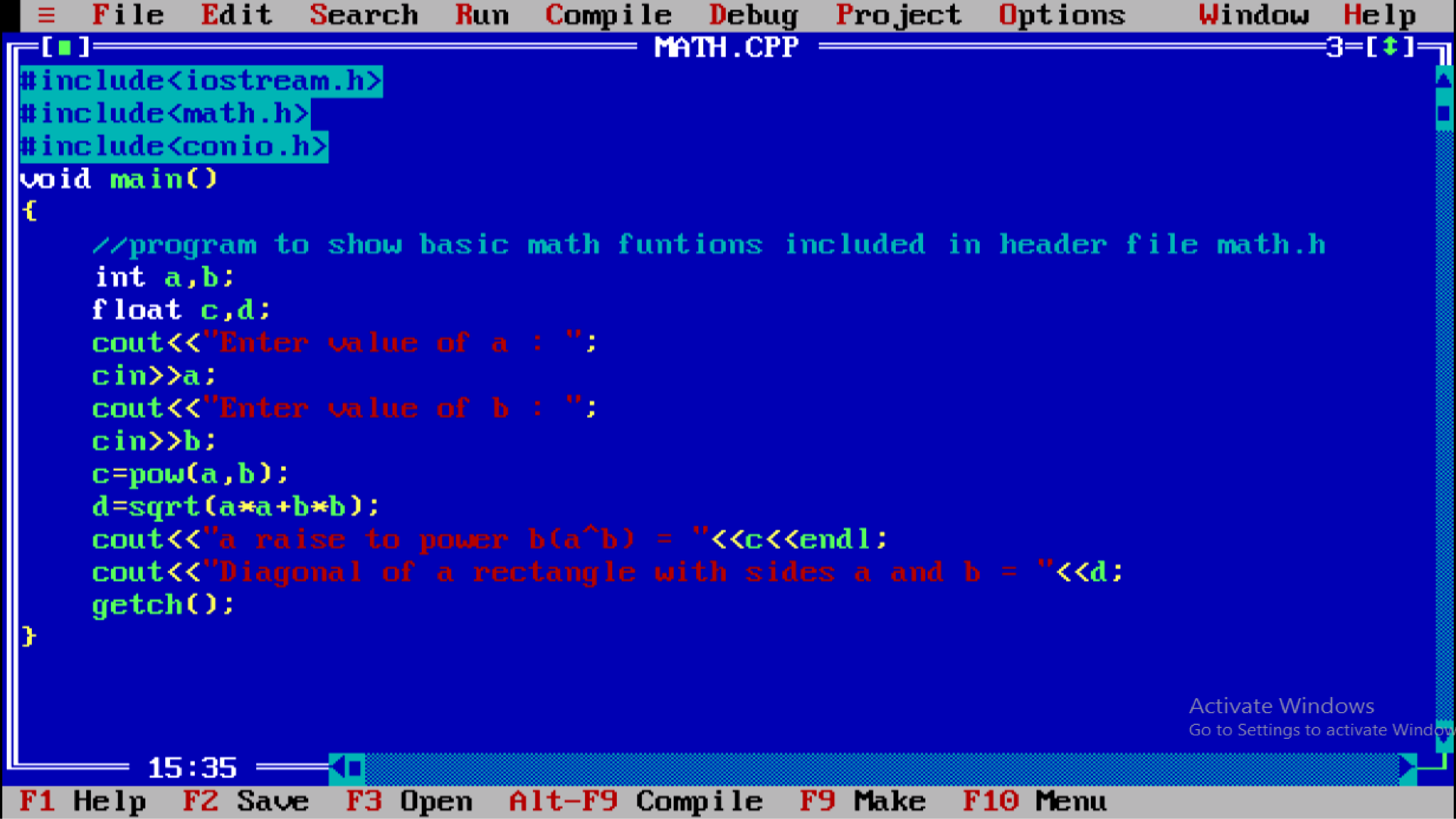


```
C:\TURBOC3\BIN>TC
Input a number2
2

Input a number3_
```

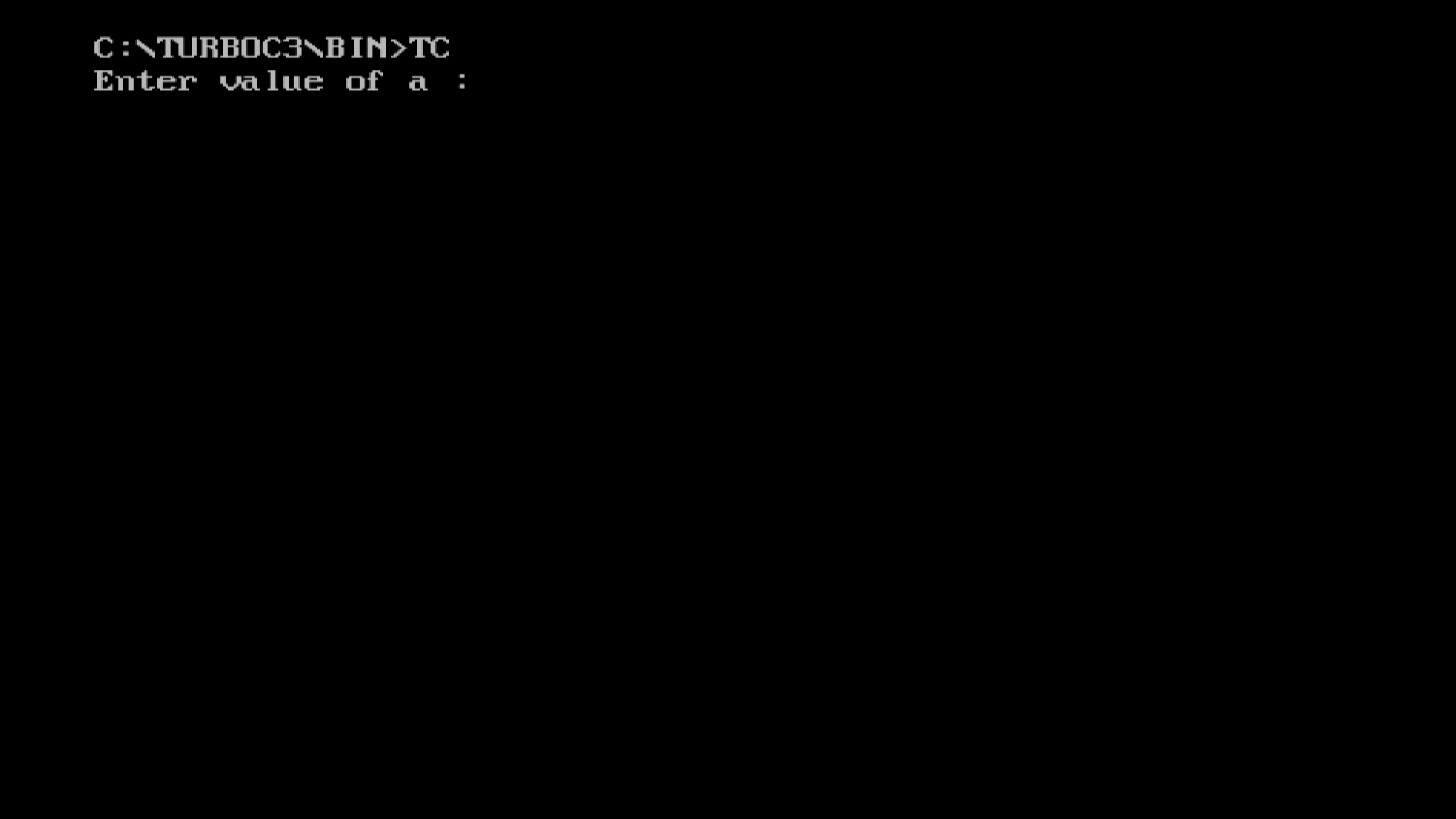


```
C:\TURBOC3\BIN>TC
Input a number2
2

Input a number3
3


_
```

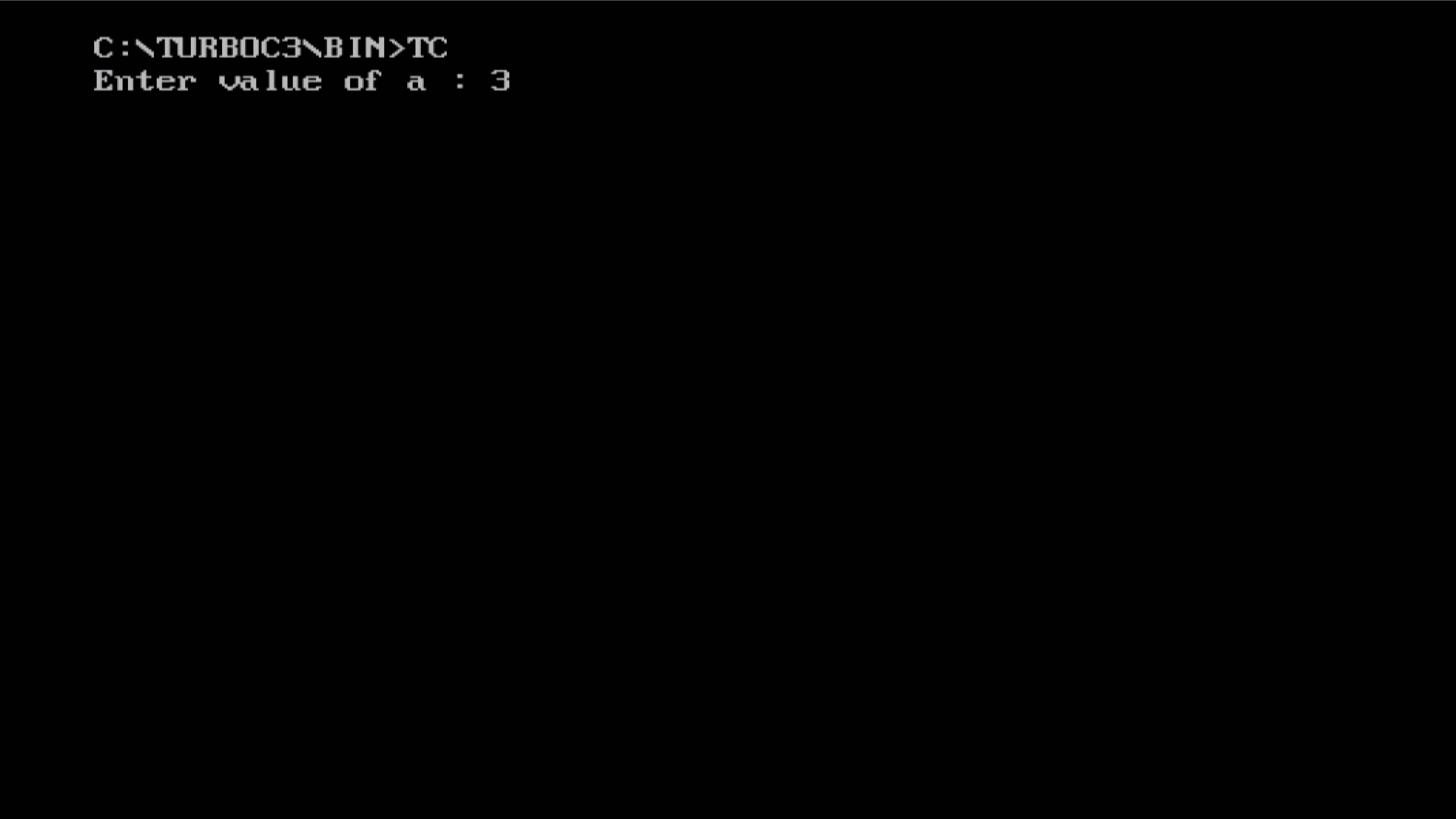
<math.h> in C++


≡ File Edit Search Run Compile Debug Project Options Window Help
[■] MATH.CPP 3 [↕]
15:35
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu
Activate Windows
Go to Settings to activate Windows

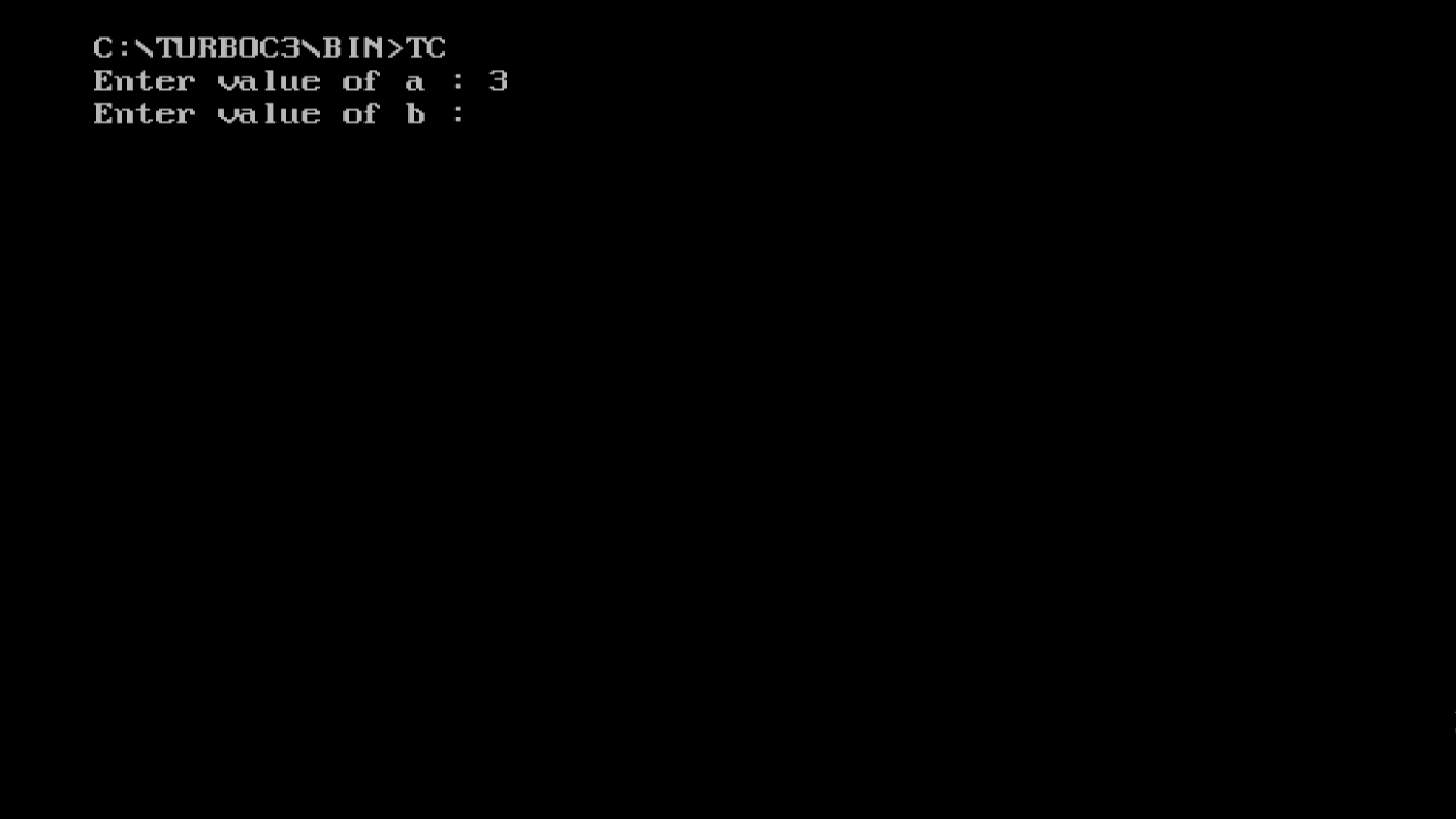
```
#include<iostream.h>
#include<math.h>
#include<conio.h>
void main()
{
    //program to show basic math funtions included in header file math.h
    int a,b;
    float c,d;
    cout<<"Enter value of a : ";
    cin>>a;
    cout<<"Enter value of b : ";
    cin>>b;
    c=pow(a,b);
    d=sqrt(a*a+b*b);
    cout<<"a raise to power b(a^b) = "<<c<<endl;
    cout<<"Diagonal of a rectangle with sides a and b = "<<d;
    getch();
}
```



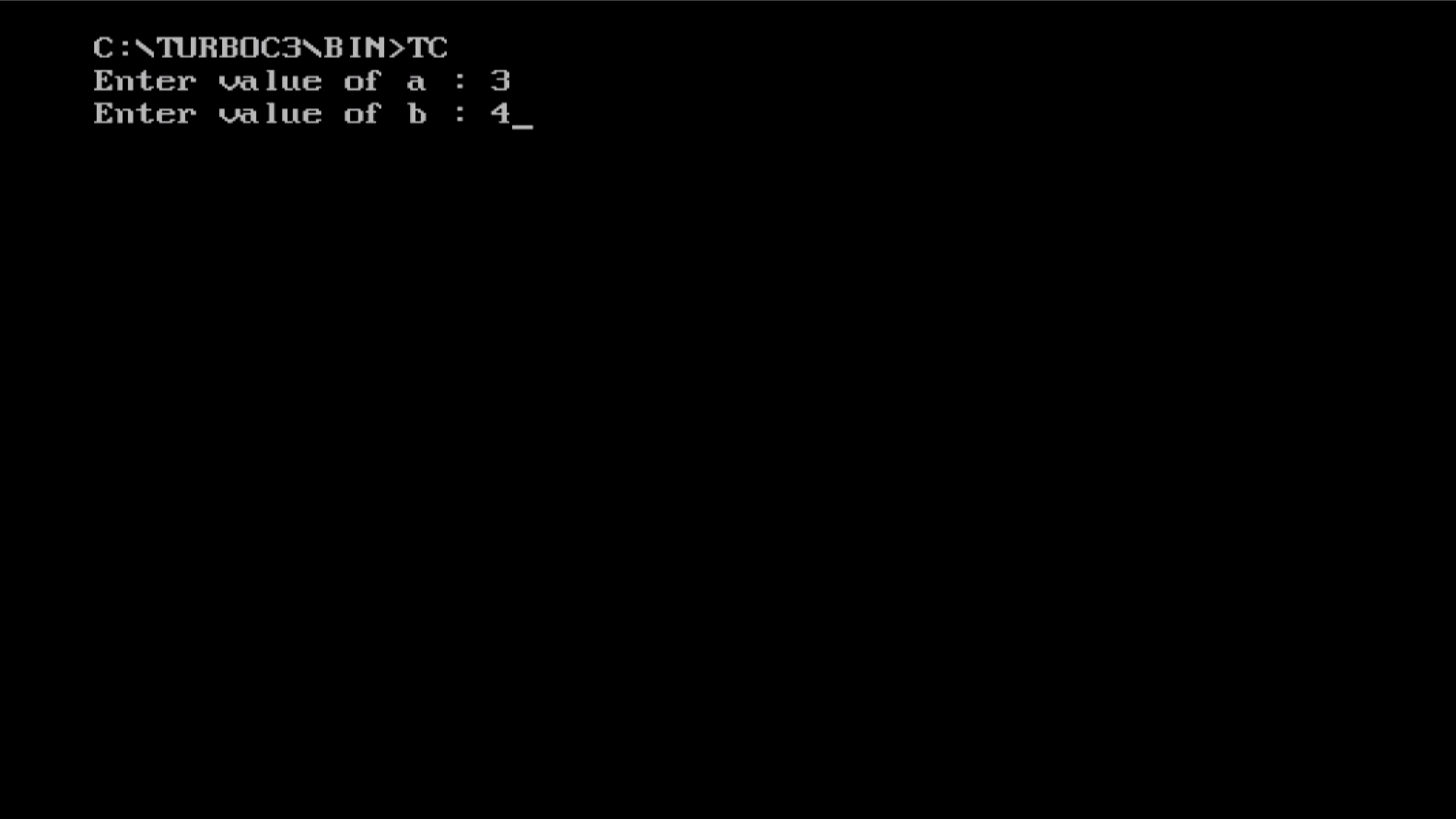
```
C:\TURBOC3\BIN>TC
Enter value of a :
```



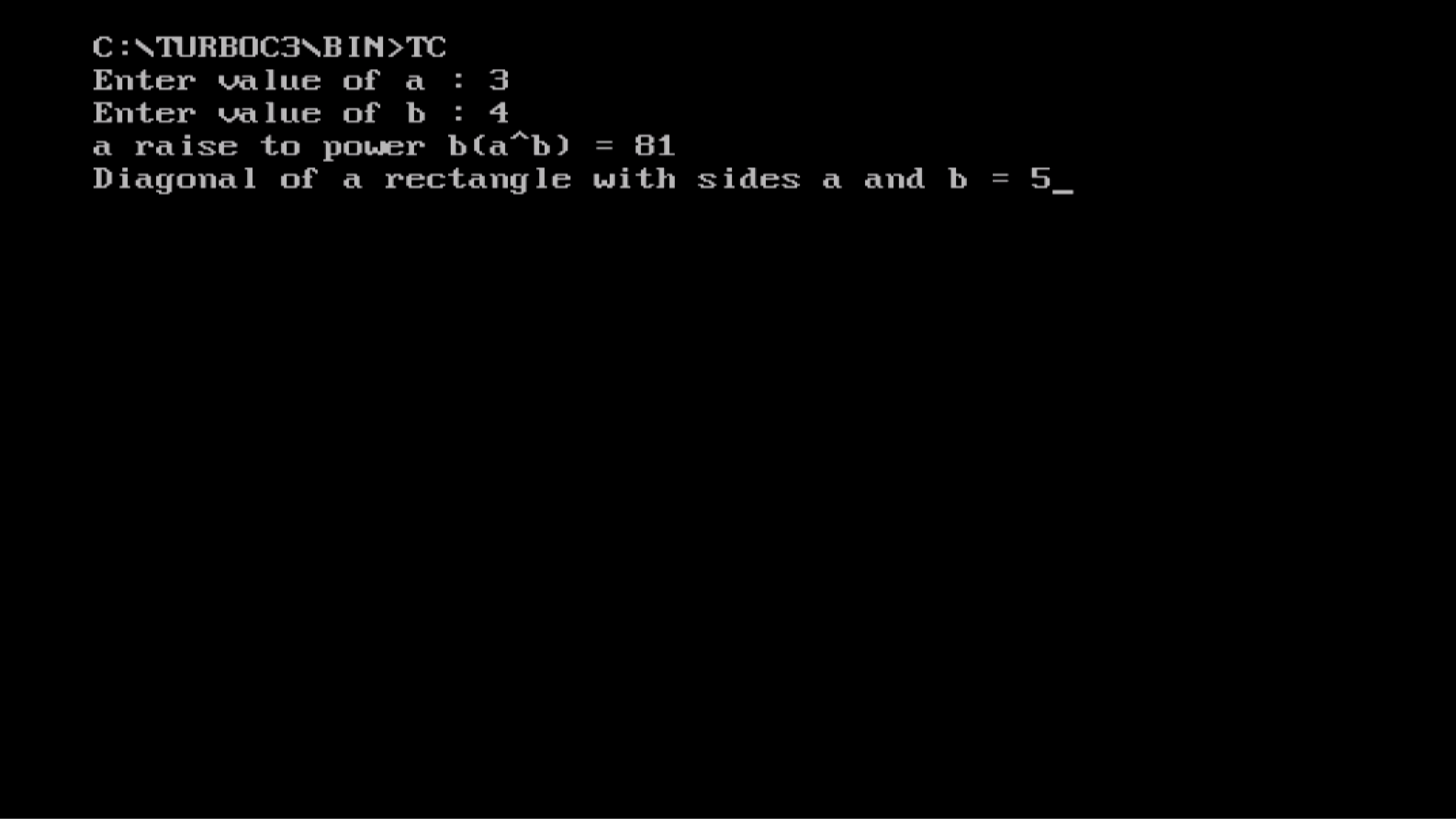
```
C:\TURBOC3\BIN>TC
Enter value of a : 3
```



```
C:\TURBOC3\BIN>TC
Enter value of a : 3
Enter value of b :
```



```
C:\TURBOC3\BIN>TC
Enter value of a : 3
Enter value of b : 4_
```



```
C:\TURBOC3\BIN>TC
Enter value of a : 3
Enter value of b : 4
a raise to power b(a^b) = 81
Diagonal of a rectangle with sides a and b = 5_
```

OOP(Object Oriented Programming)

In object based programming data and its associated meaningful functions are enclosed in one single entity a **class**.

Class enforces information hiding and abstraction thereby separating the implementation details and the user interface.

For example: - consider a calculator, interface includes the display screen and some buttons that is available to user and the implementation details i.e., how the actual calculations are done are hidden from the user.

CLASSES

The most important feature of C++ are Classes and Objects.

A class is a way to bind the data, describing an entity and its associated functions together.

The declaration of class involves declaration of its four associated attributes:
1)Data Members
2) Member Functions
3) Program Access Levels
4) Class Tag Name

GENERAL FORM OF A CLASS DEFINITION

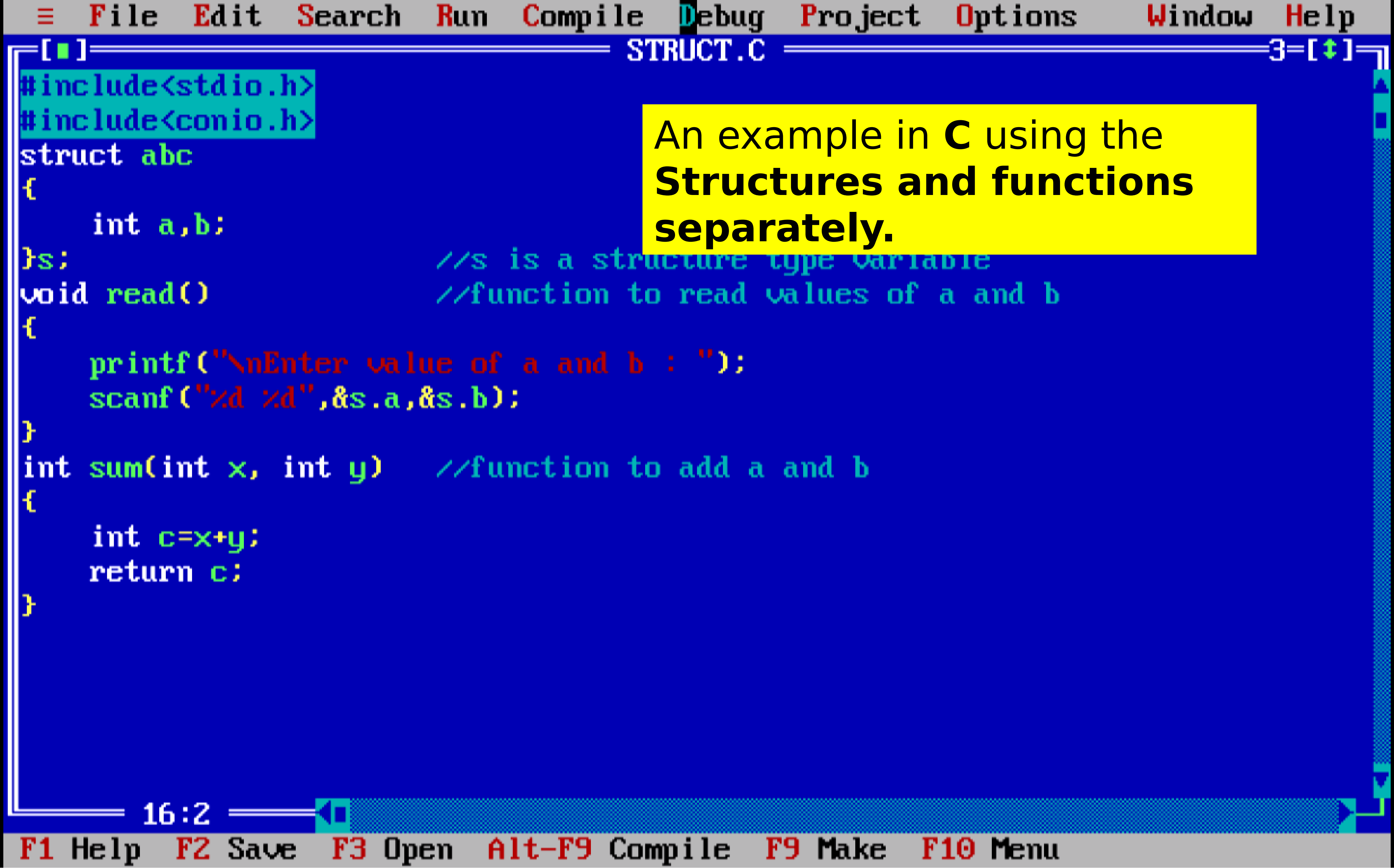
```
class class-name{
        private:
            [variable declaration;]
            [function declaration;]
        protected:
            [variable declaration;]
            [function declaration;]
        public:
            [variable declaration;]
            [function declaration;]
};
```

Where the **keyword class** specifies that it is a class; **class-name** is the tag name of the class using which objects of this class types can be created.

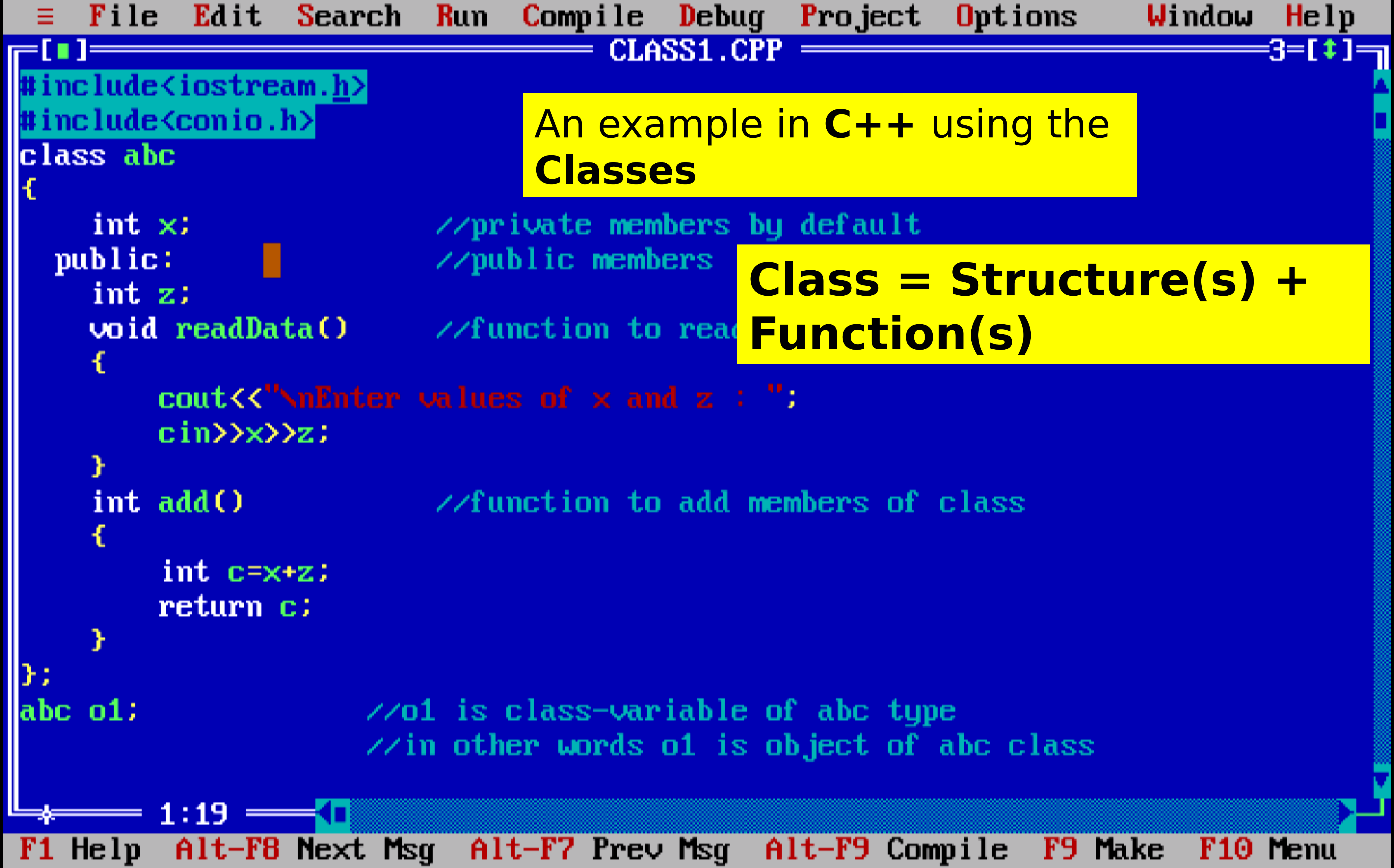
The **class body** contains the declaration of its members under three access levels namely **Private, Protected and Public**.

The **Private members** can be accessed only from within the class and the **public members** can be accessed from outside the class also whereas the **Protected members** are the members are the members that can be used only by member functions and friends of the class in which it is declared.

An example in **C** using the **Structures and functions separately.**


≡ File Edit Search Run Compile Debug Project Options Window Help
STRUCT.C
3
16:2
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu

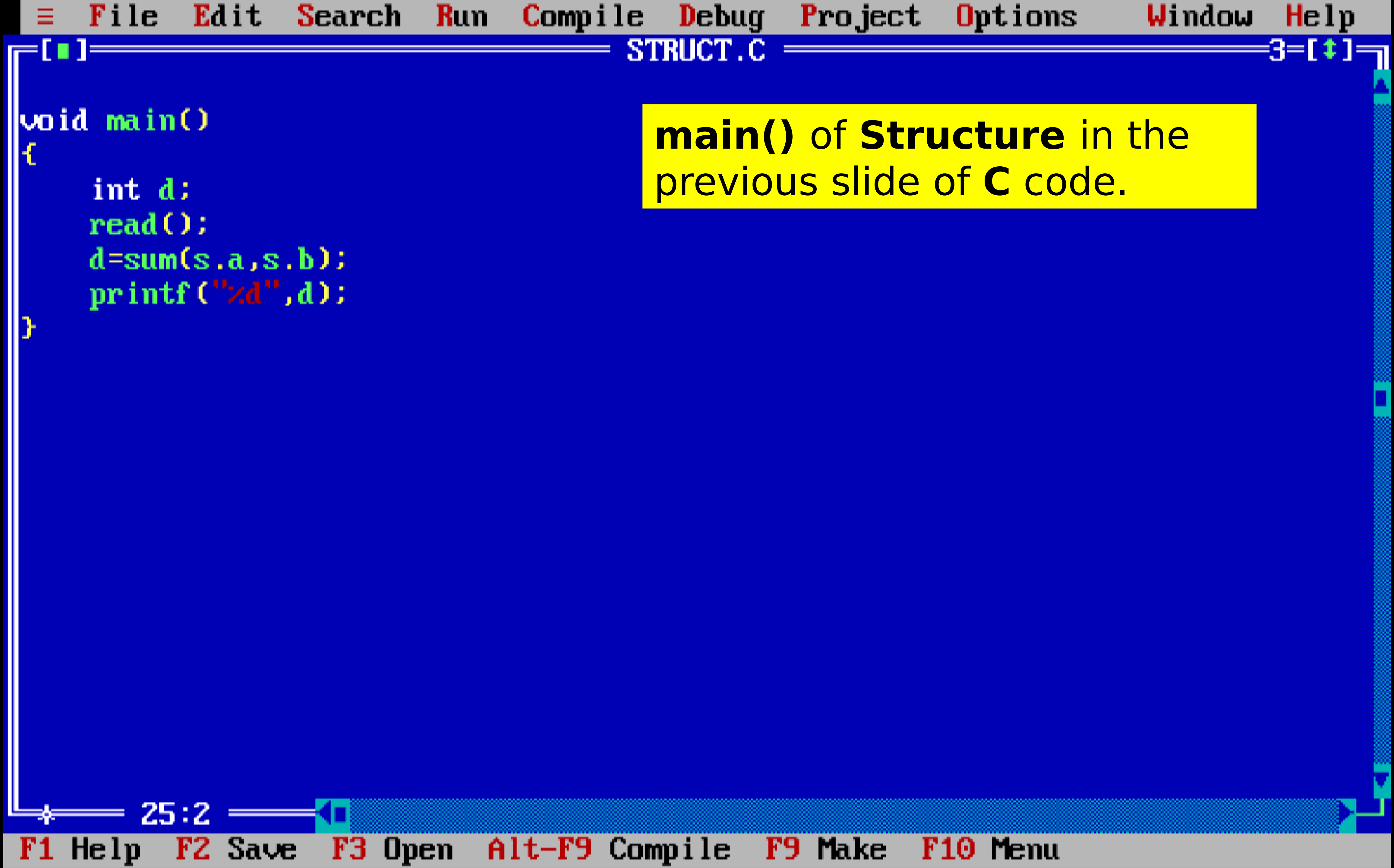
```
#include<stdio.h>
#include<conio.h>
struct abc
{
    int a,b;
}s;                         //s is a structure type variable
void read()                 //function to read values of a and b
{
    printf("\nEnter value of a and b : ");
    scanf("%d %d",&s.a,&s.b);
}
int sum(int x, int y)       //function to add a and b
{
    int c=x+y;
    return c;
}
```


≡ File Edit Search Run Compile Debug Project Options Window Help
CLASS1.CPP
3
1:19
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9 Make F10 Menu

```
#include<iostream.h>
#include<conio.h>
class abc
{
    int x;              //private members by default
  public:               //public members
    int z;
    void readData()     //function to rea
    {
        cout<<"\nEnter values of x and z : ";
        cin>>x>>z;
    }
    int add()           //function to add members of class
    {
        int c=x+z;
        return c;
    }
};
abc o1;             //o1 is class-variable of abc type
                    //in other words o1 is object of abc class
```

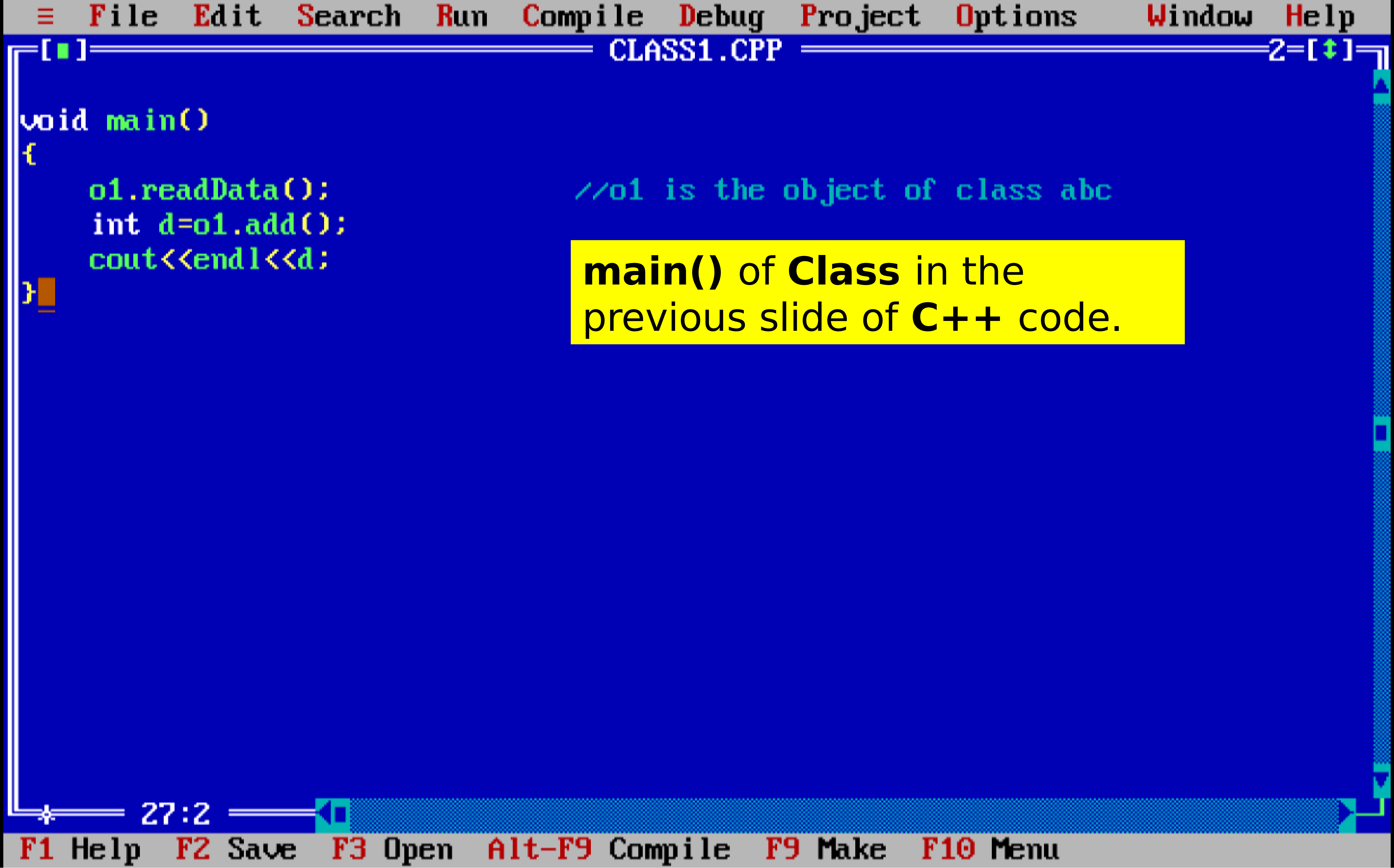
An example in **C++** using the **Classes**

Class = Structure(s) + Function(s)


≡ File Edit Search Run Compile Debug Project Options Window Help
[■] STRUCT.C 3 [↕]
25:2
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu

```
void main()
{
    int d;
    read();
    d=sum(s.a,s.b);
    printf("%d",d);
}
```

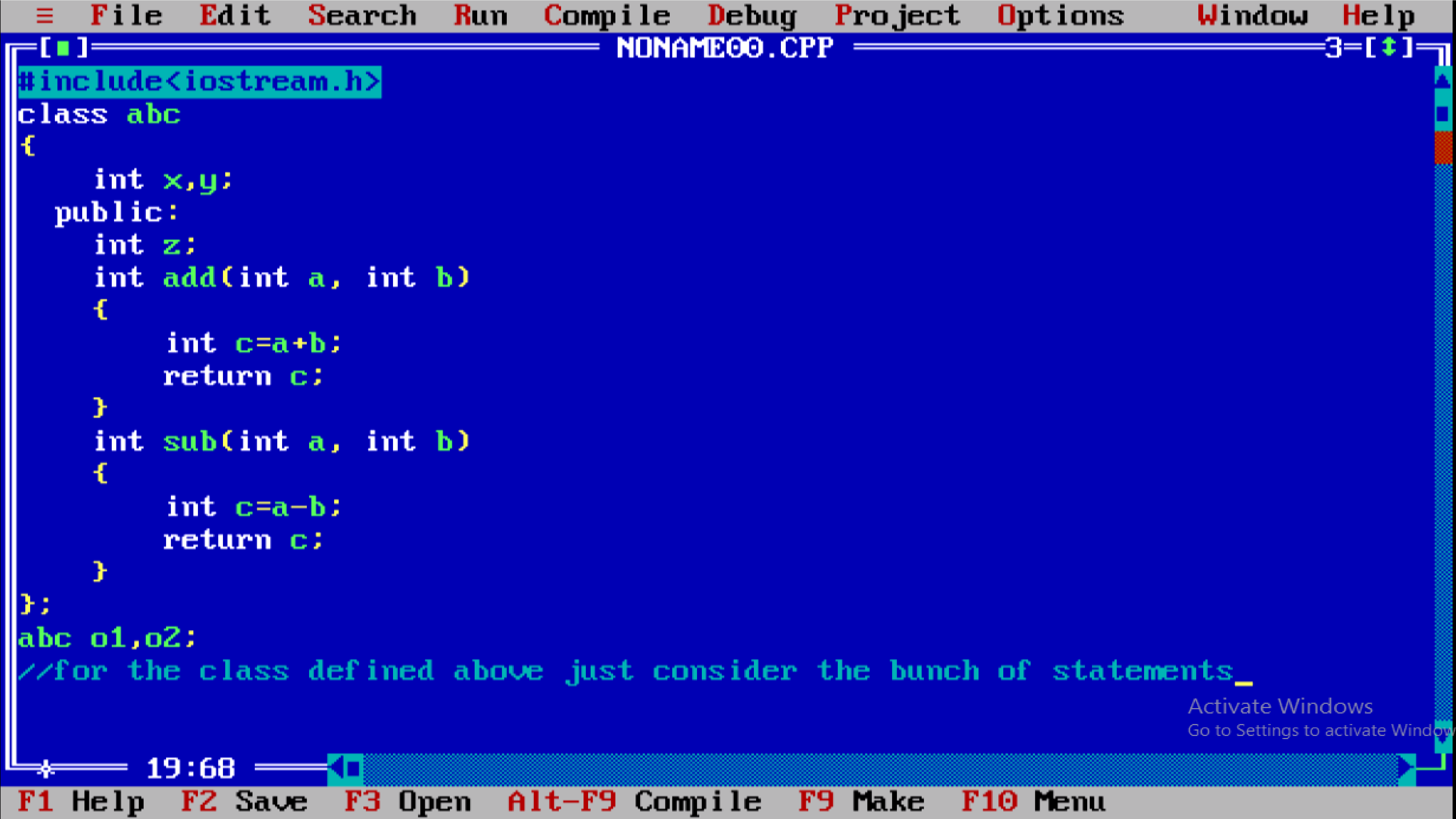
main() of **Structure** in the previous slide of **C** code.



≡ File Edit Search Run Compile Debug Project Options Window Help
[■] CLASS1.CPP 2 [↕]
27:2
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu

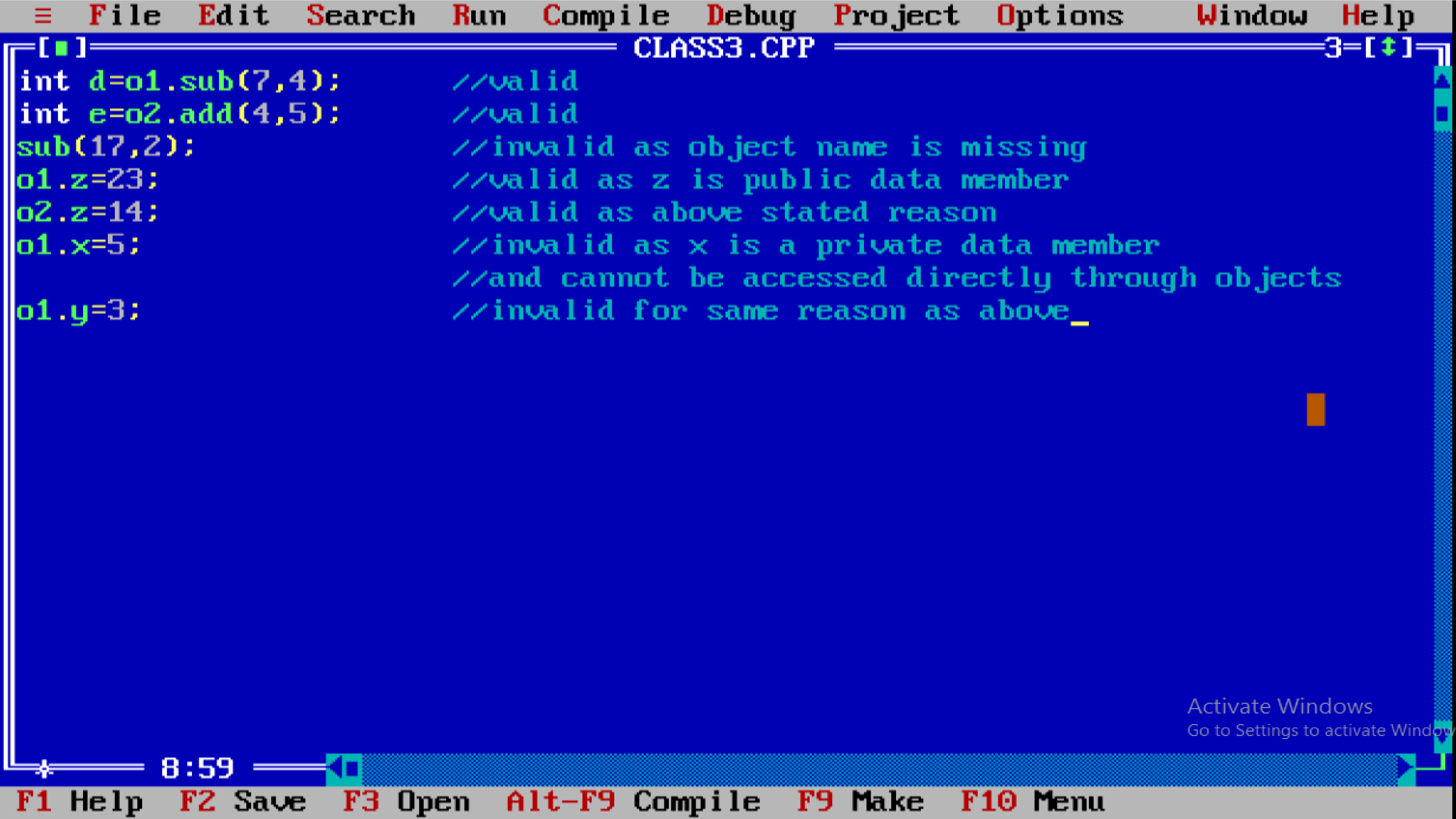
```

void main()
{
    o1.readData();                      //o1 is the object of class abc
    int d=o1.add();
    cout<<endl<<d;
}
```

main() of **Class** in the previous slide of **C++** code.



≡ File Edit Search Run Compile Debug Project Options Window Help
[■]
NONAME00.CPP
3 [↕]
19:68
Activate Windows
Go to Settings to activate Windo
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu

```
#include<iostream.h>
class abc
{
    int x,y;
  public:
    int z;
    int add(int a, int b)
    {
        int c=a+b;
        return c;
    }
    int sub(int a, int b)
    {
        int c=a-b;
        return c;
    }
};
abc o1,o2;
//for the class defined above just consider the bunch of statements
```


≡ File Edit Search Run Compile Debug Project Options Window Help
[■]
CLASS3.CPP
3
[↕]
8:59
Activate Windows
Go to Settings to activate Windows
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu

```
int d=o1.sub(7,4);       //valid
int e=o2.add(4,5);       //valid
sub(17,2);               //invalid as object name is missing
o1.z=23;                 //valid as z is public data member
o2.z=14;                 //valid as above stated reason
o1.x=5;                  //invalid as x is a private data member
                         //and cannot be accessed directly through objects
o1.y=3;                  //invalid for same reason as above
```

To Read about **graphics.h** and
its included functions,
follow the link :

https://www.tinyurl.com/graphicsdoth

Please come prepared
with your doubts

C++ Class timings :
8:30 pm onwards

Venue:
Any Available LT