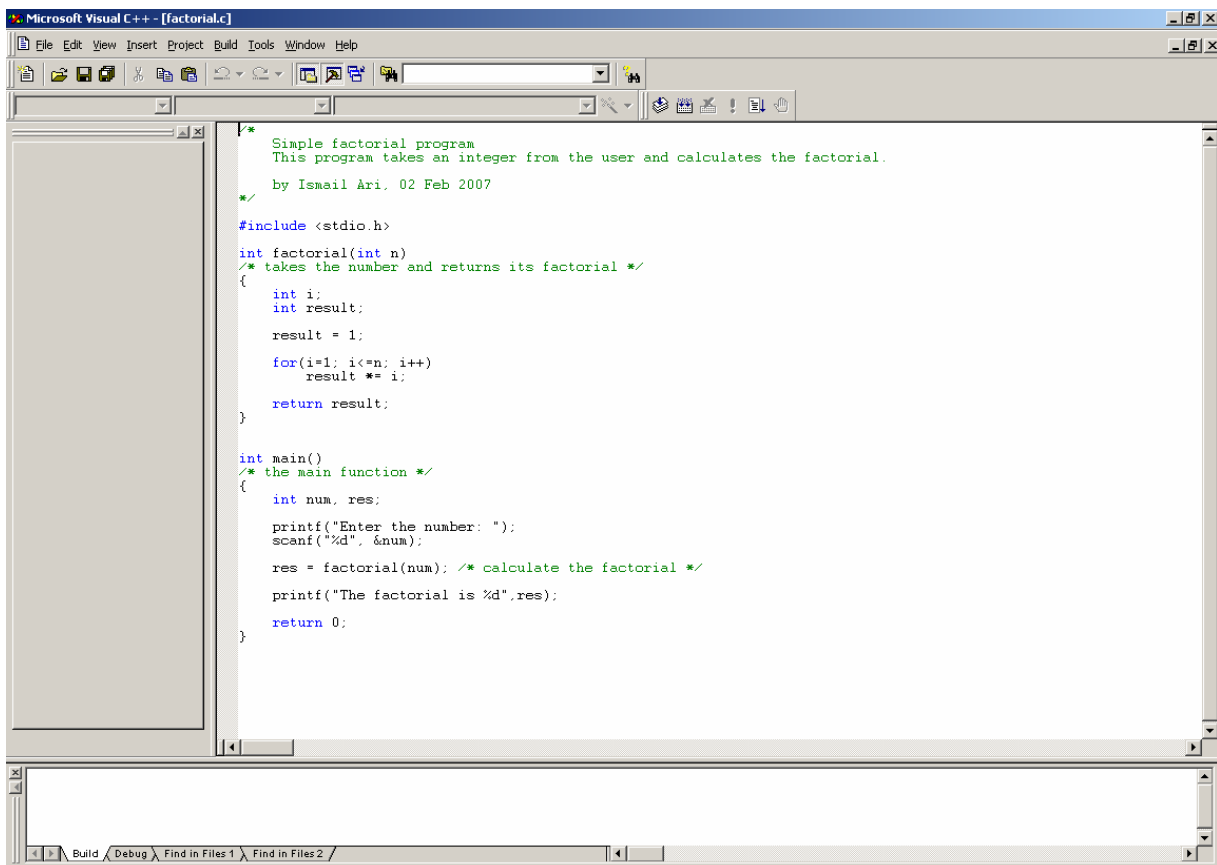


### Visual C++ Debug Tutorial

**QUESTION:** How do I set breakpoints, set watches, and set through an executing program line-by-line?

**ANSWER:** We will use the program *factorial.c* as our example. Note that we are using Visual C++ version 6. However, the steps in other Visual C++ versions are very similar. Open Visual C++.

1. You begin by opening the file *factorial.c* as the sample c document. Visual C++ will then display the file *factorial.c* in the code window.



```
Microsoft Visual C++ - [factorial.c]
File Edit View Insert Project Build Tools Window Help
Simple factorial program
This program takes an integer from the user and calculates the factorial.
by Ismail Ari. 02 Feb 2007
#include <stdio.h>
int factorial(int n)
/* takes the number and returns its factorial */
{
    int i;
    int result;

    result = 1;

    for(i=1; i<=n; i++)
        result *= i;

    return result;
}

int main()
/* the main function */
{
    int num, res;

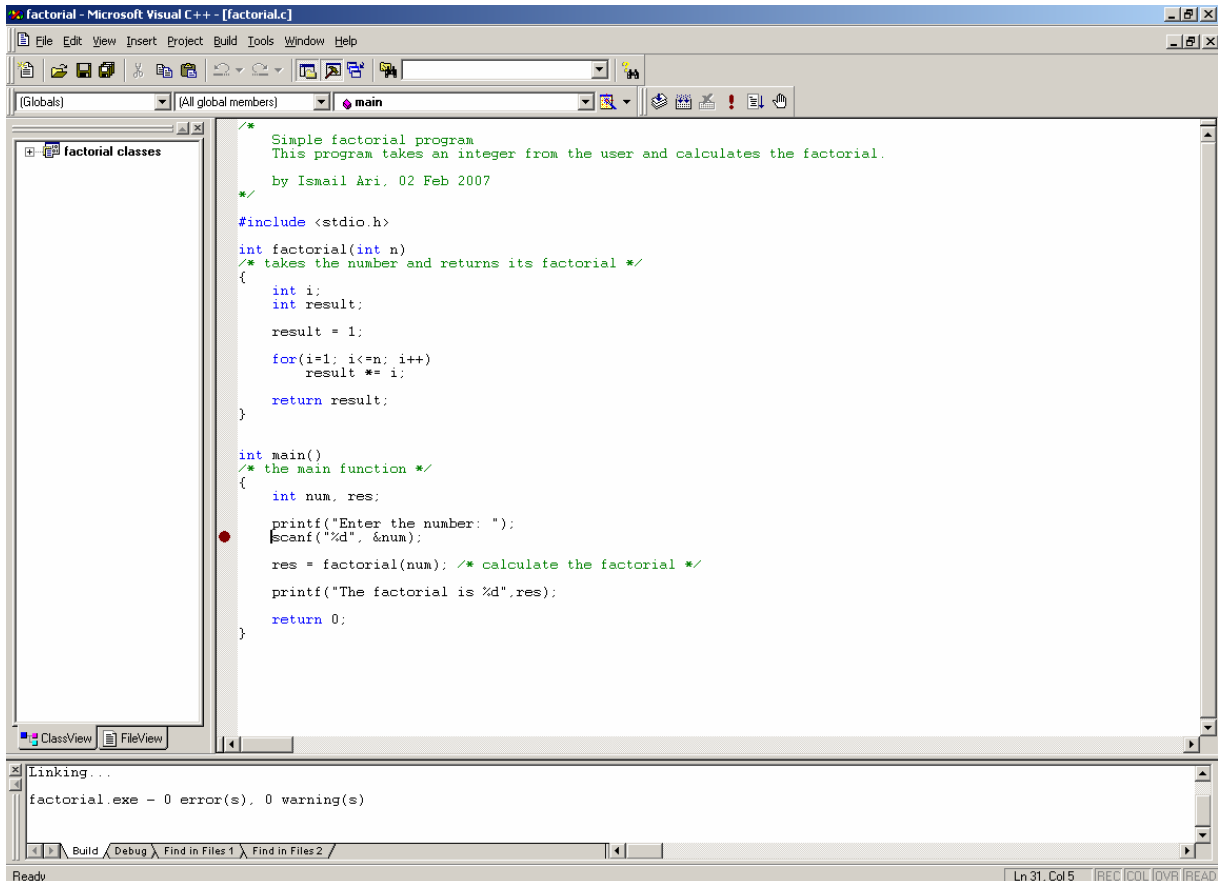
    printf("Enter the number: ");
    scanf("%d", &num);

    res = factorial(num); /* calculate the factorial */

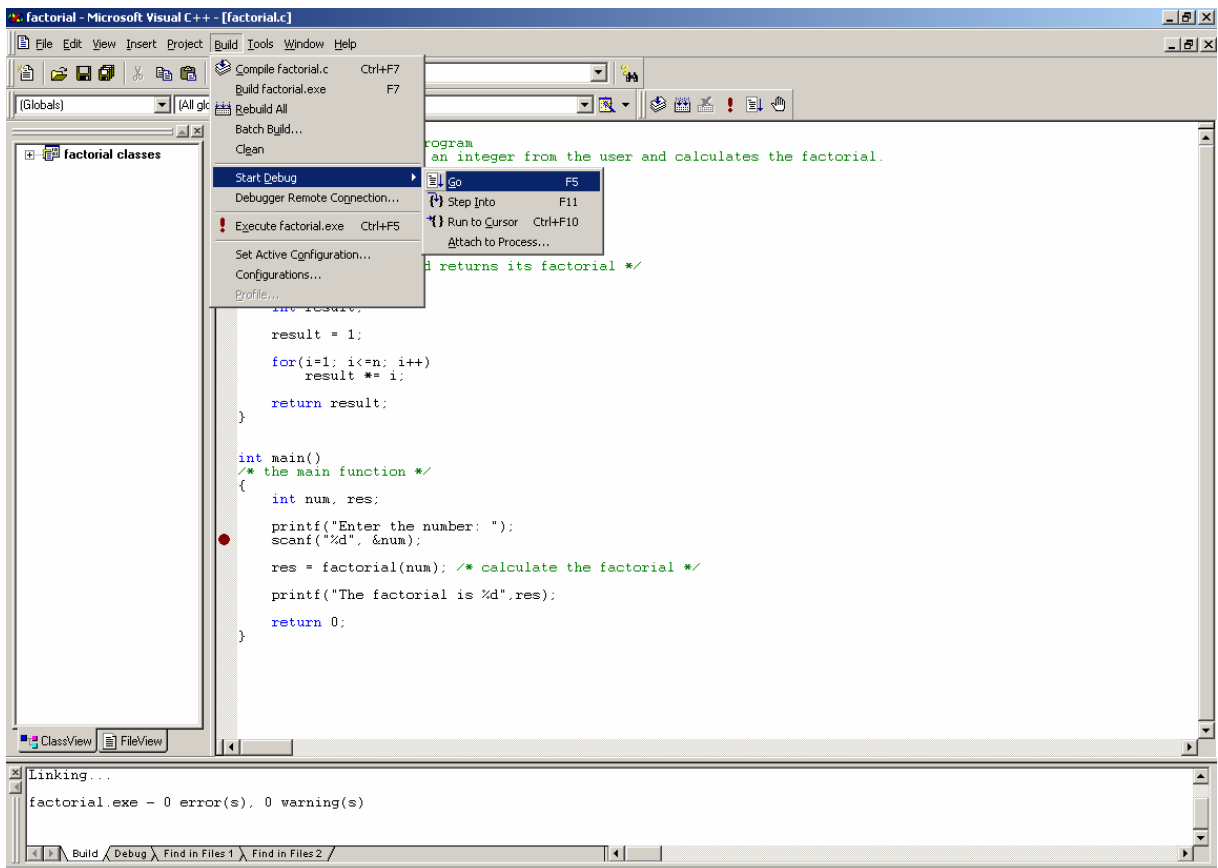
    printf("The factorial is %d",res);

    return 0;
}
```

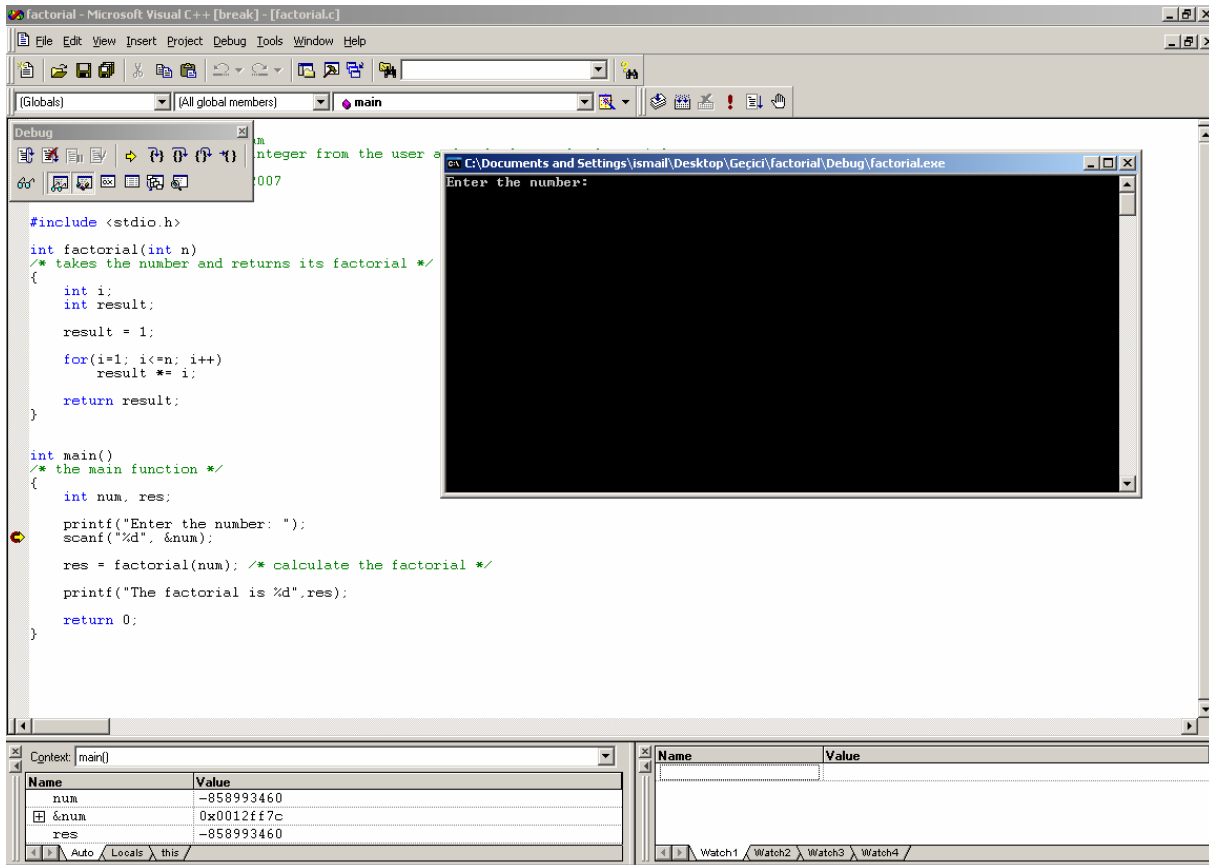
2. Once the file *factorial.c* is open, you build an executable version of the program: Select **Build** | **Build**. Visual C++ will ask if you want to create a default project workspace. Select **Yes**.
3. Set a breakpoint at the following line by right clicking the cursor on this line, and then clicking on the "hand" (Insert/Remove Breakpoint). Alternately, you can use F9 special key. A red breakpoint "stop sign" appears to the left of the line of code.



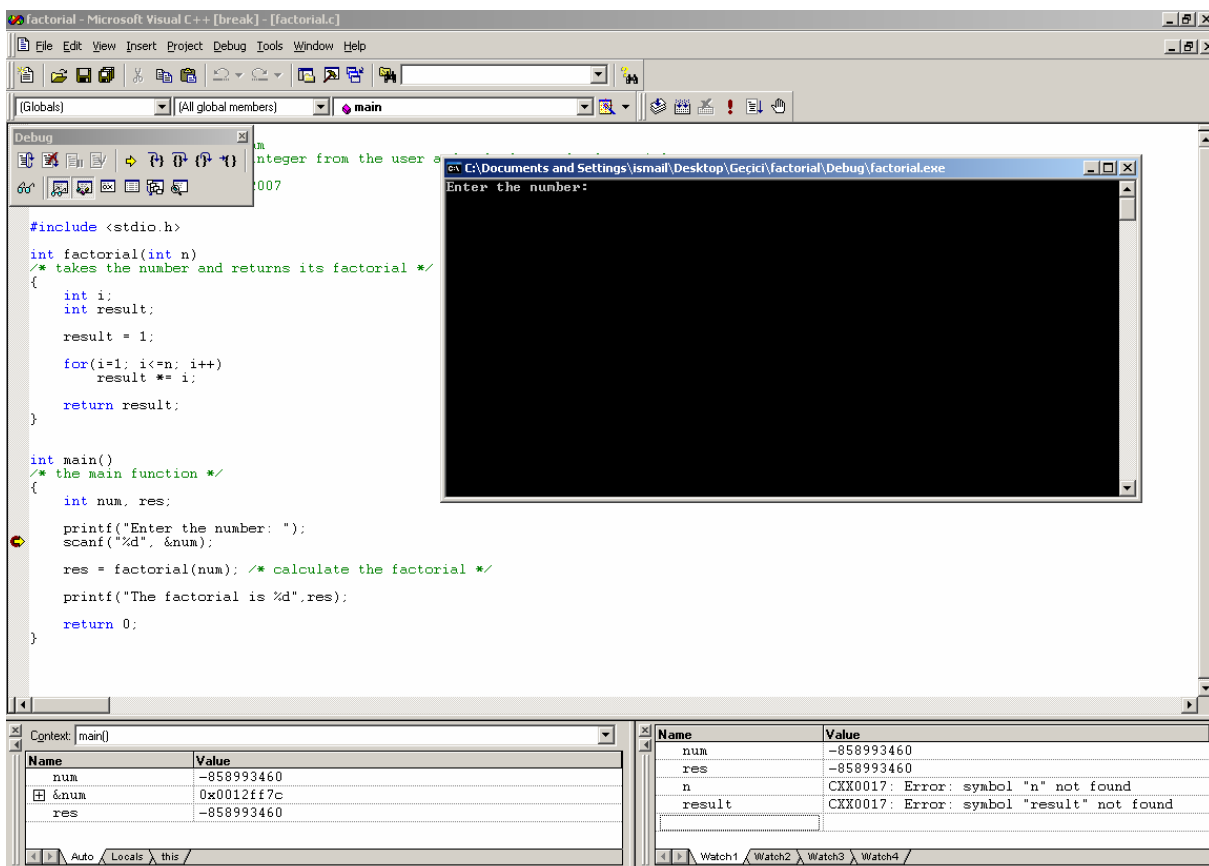
4. Run `factorial.c` in debug mode by choosing **Build | Start Debug | Go**. Alternately, you can press **F5**.



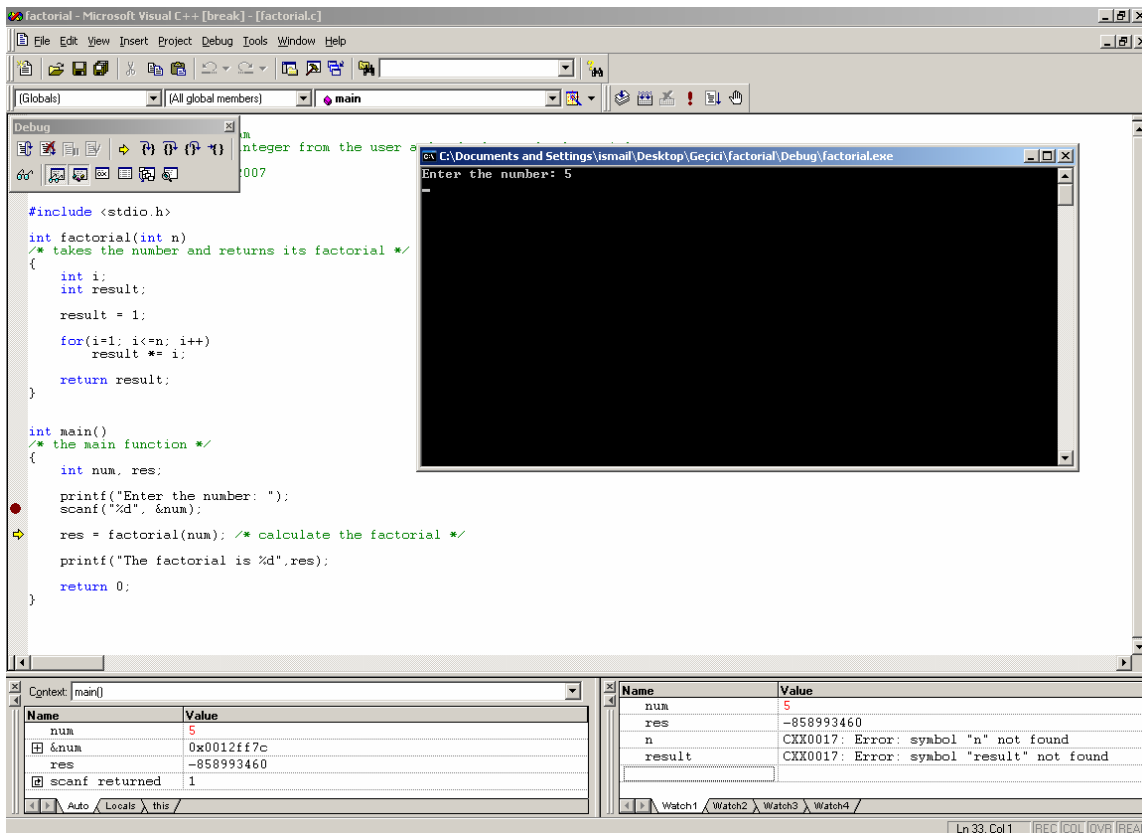
5. Note that execution stops at the point where the breakpoint was set.



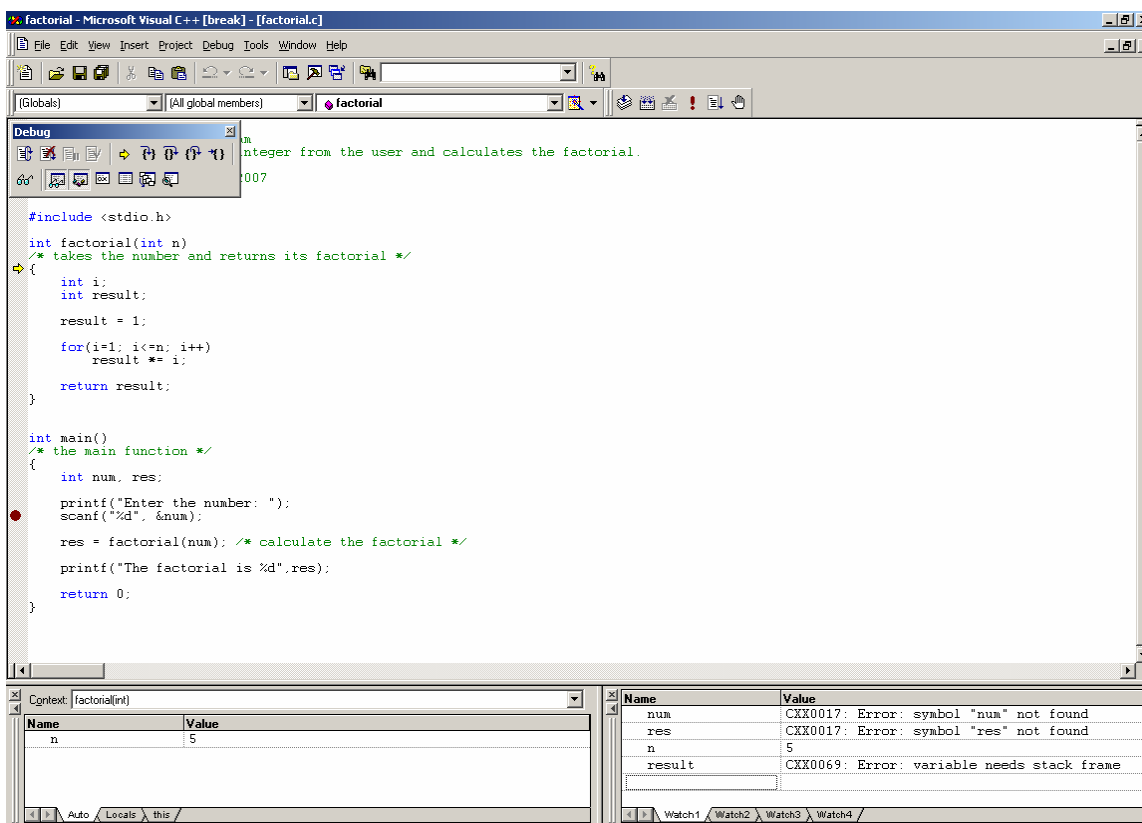
6. Set a watch on the variable “num”, “res”, “n” and “result” by typing them in the name column of the Watch1 window in the lower part of the screen as follows. Note that “num” and “res” are not assigned yet, so they have garbage values. But “n” and “result” are not in the scope of main function, so they are not recognized. You will understand it better later.



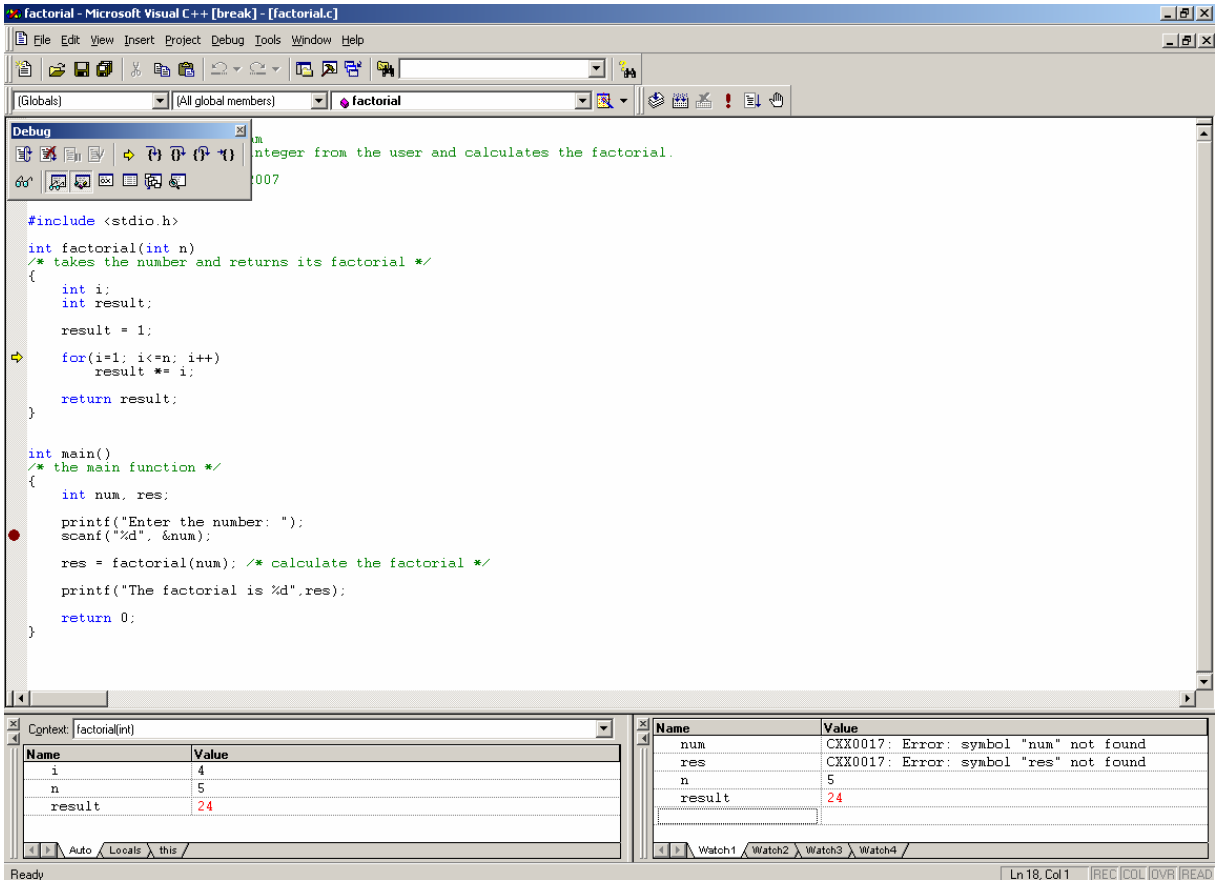
- Enter your number from the running program interface and press Enter. The yellow arrow will point to the next line in the Visual C++ screen indicating that you are in that screen. In addition, notice that the value of “num” has changed to 5 in the watch screen. You can also use Auto Watch screen if you like.



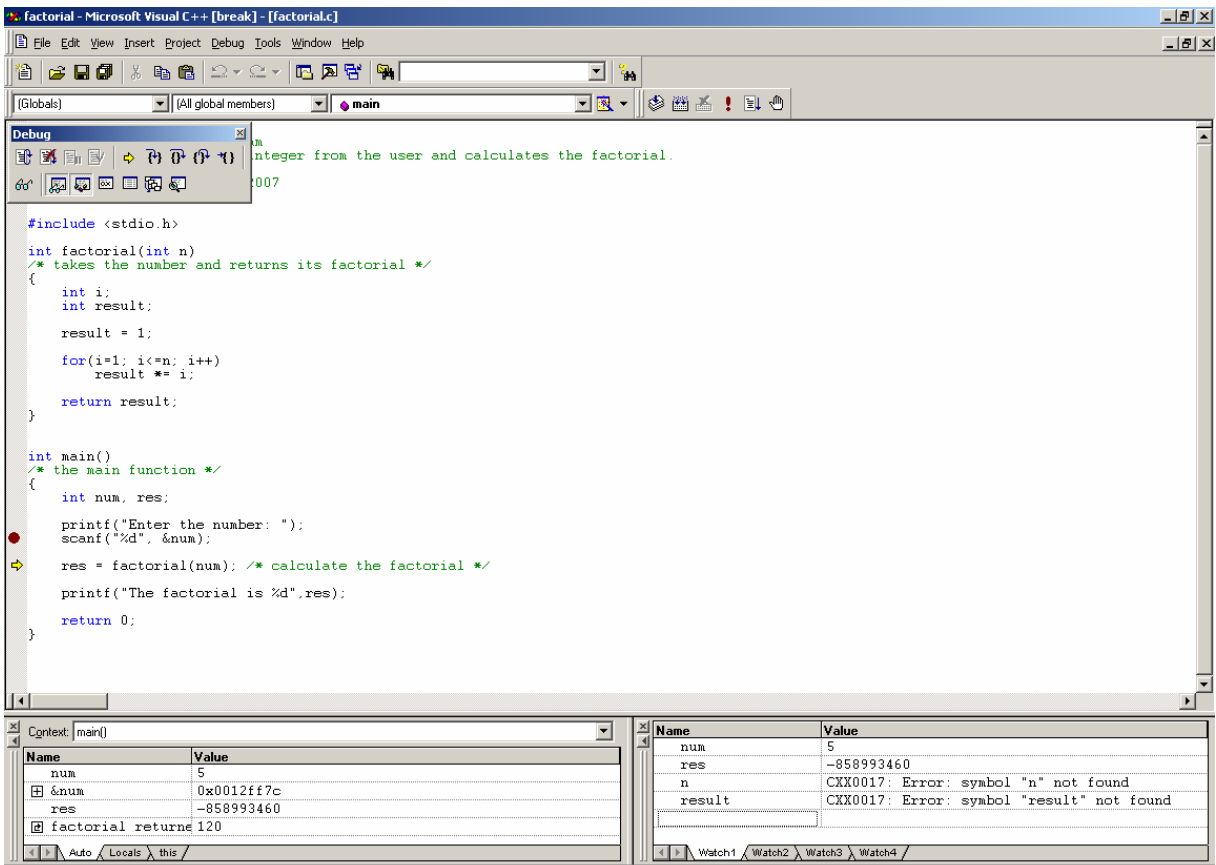
- Step into the function we have written using **Debug | Step Into** (or F11). Now, you will see the yellow arrow in the function we wrote. Notice “n” is assigned to 5 in *factorial* function. All other variables are not defined in this scope. We will define “result” in the next lines.



9. Step over the functions using **Debug | Step Over** (or F10) and notice the changes of “result” variable from the watch screen. An intermediate step is as follows



10. Once you want to step out the *factorial* function, use **Debug | Step Out** (or Shift F11). This will return you to the step where we have called *factorial* function from the parent function.



- Again step over the line by using F10. Realize that “res” is now set to 120. Another important way to watch the value is to move the cursor to a variable. Notice that the popup box shows “res = 120”, because the cursor is standing on it.

