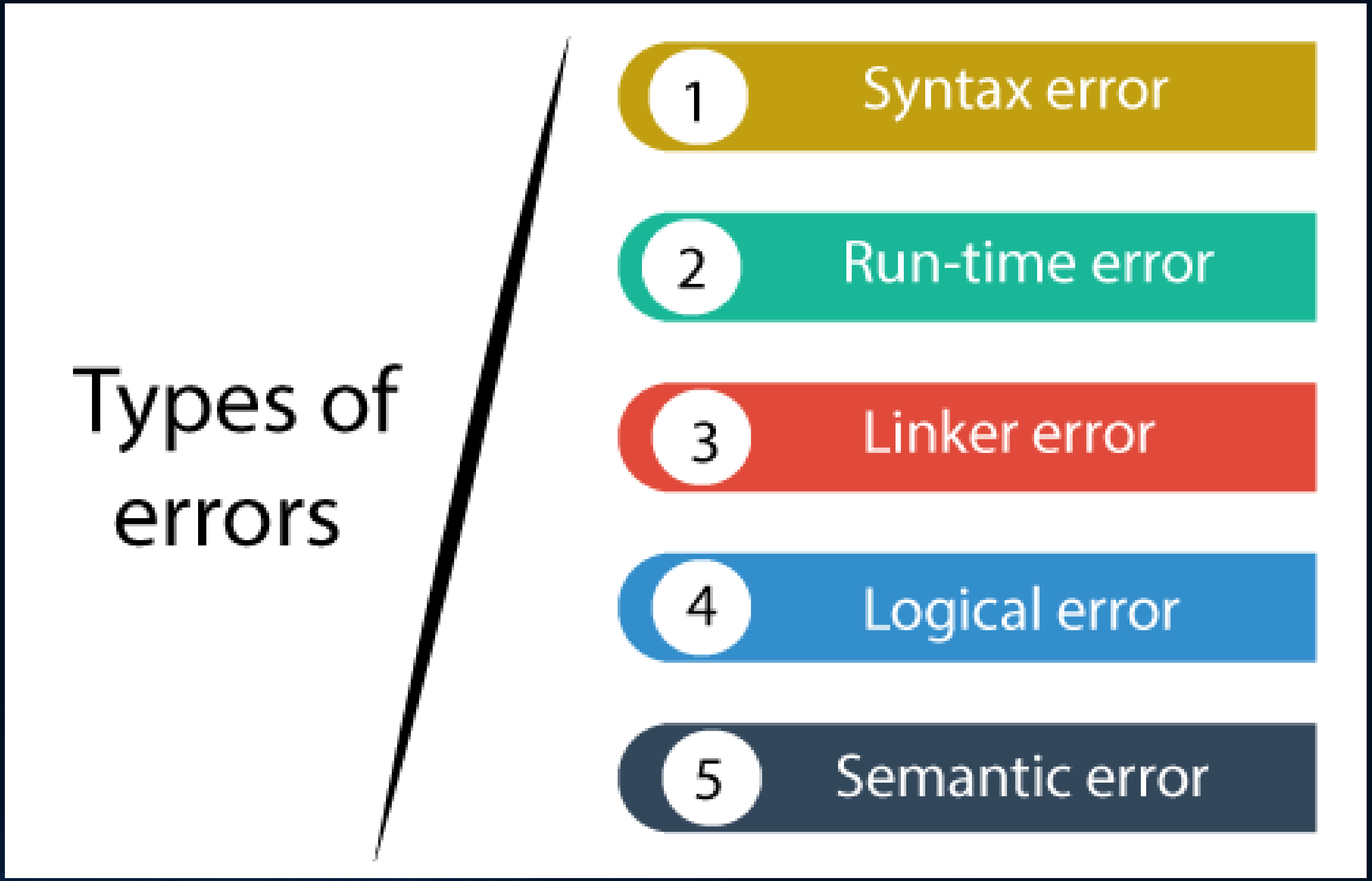
C-Language & Its Errors



Types of errors

1 Syntax error

2 Run-time error

3 Linker error

4 Logical error

5 Semantic error

When we compile any C -program sometimes it give us a Bug or an Error.

Program errors are also referred to as program bugs.

A C program may have one or more of four types of errors:

- *Syntax errors (Compiler errors or Compile-time errors)*
- *Linker Errors*
- *Runtime errors*
- *Logic errors*
- *Semantic error*

Syntax Errors

The set of rules (grammatical rules) of a programming language for writing statements of the computer program is known as syntax of the language. The program statements are written strictly according to these rules.

Syntax error occur when syntax of a programming language are not followed in writing the source code. The compiler detects these errors at compiling time of source code. The compiler reports a proper error message about the error.

The compiler does not compile a program that contain syntax errors. The syntax errors are easy to detect and remove.

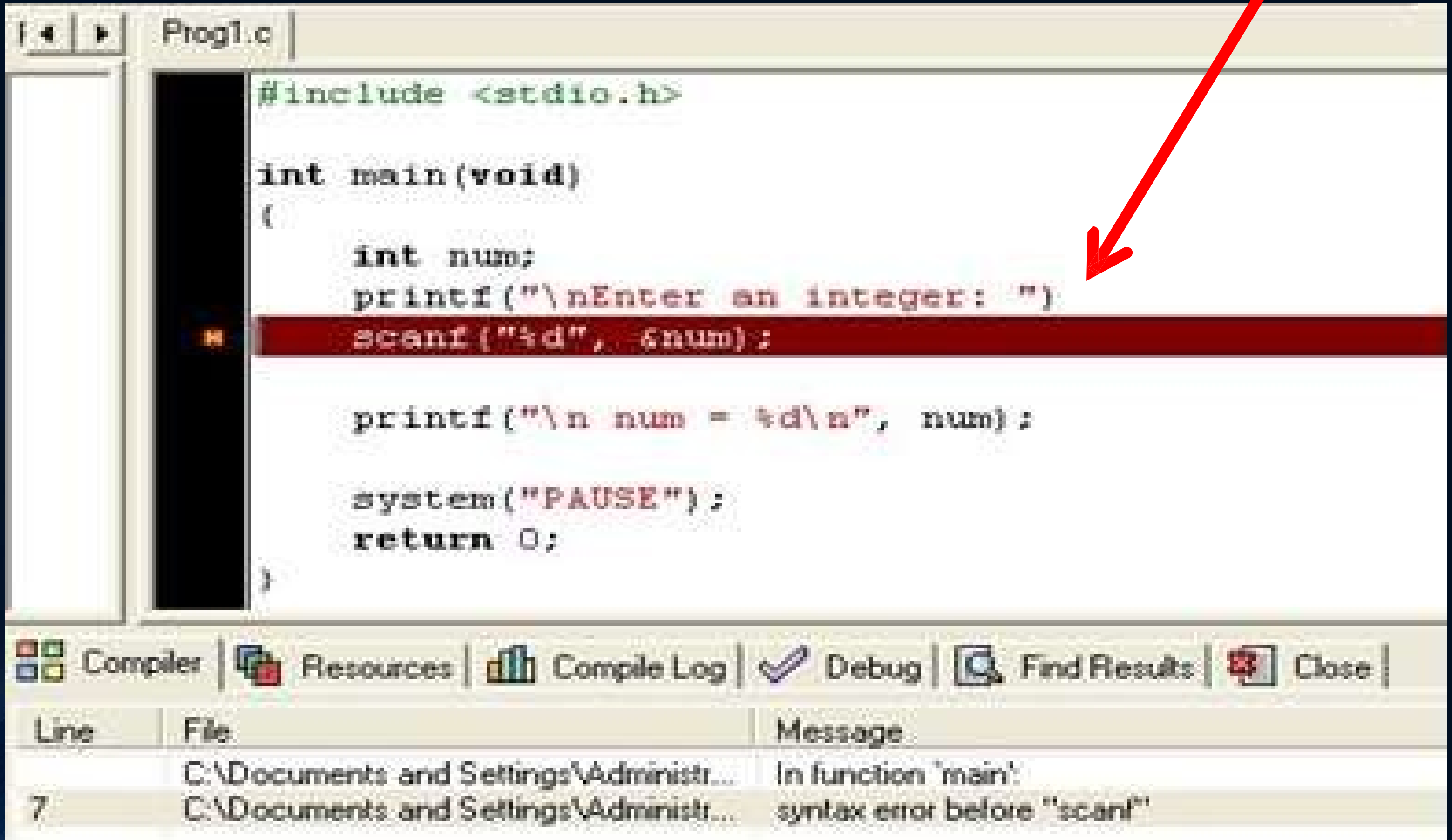
In C program, there can be many causes of syntax errors. Some examples are given below:-

- *Missing semicolon (;) at the end of statement.*
- *Missing any of delimiters i.e { or }*
- *Incorrect spelling of any keyword.*
- *Using variable without declaration etc.*

Syntax errors are the easiest to find and fix. Over the years, compiler developers have worked hard to make compilers smarter so that they can catch errors at compile time that might otherwise turn out to be runtime errors.

Examples

1. Simple statement not terminated by semicolon:

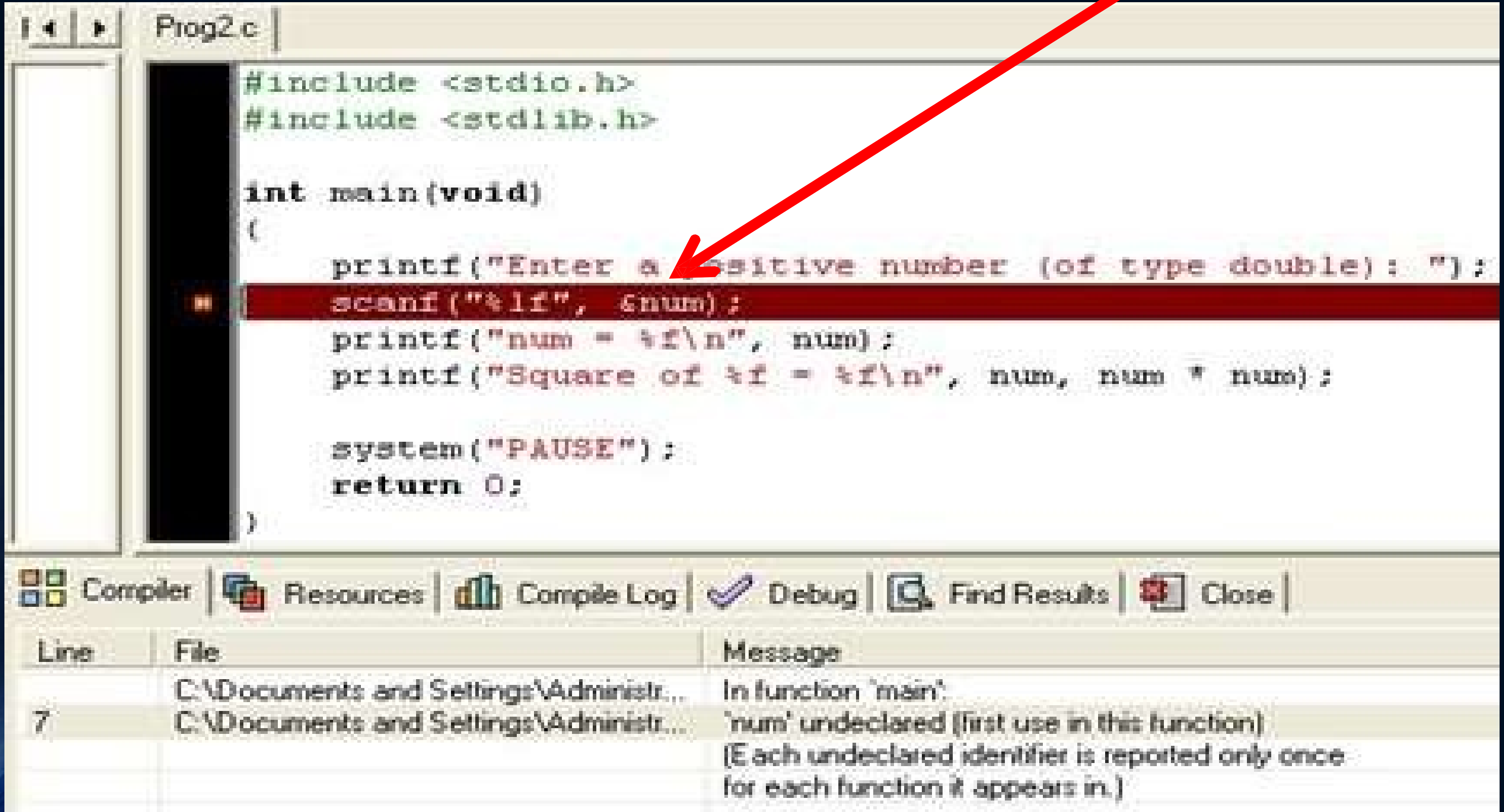


```c
#include <stdio.h>

int main(void)
{
    int num;
    printf("\nEnter an integer: ")
    scanf("%d", &num);

    printf("\n num = %d\n", num);

    system("PAUSE");
    return 0;
}
```

```
Line  File                                   Message
      C:\Documents and Settings\Administr...  In function 'main':
7     C:\Documents and Settings\Administr...  syntax error before "scanf"
```

Examples

2. Using a variable before it is declared:

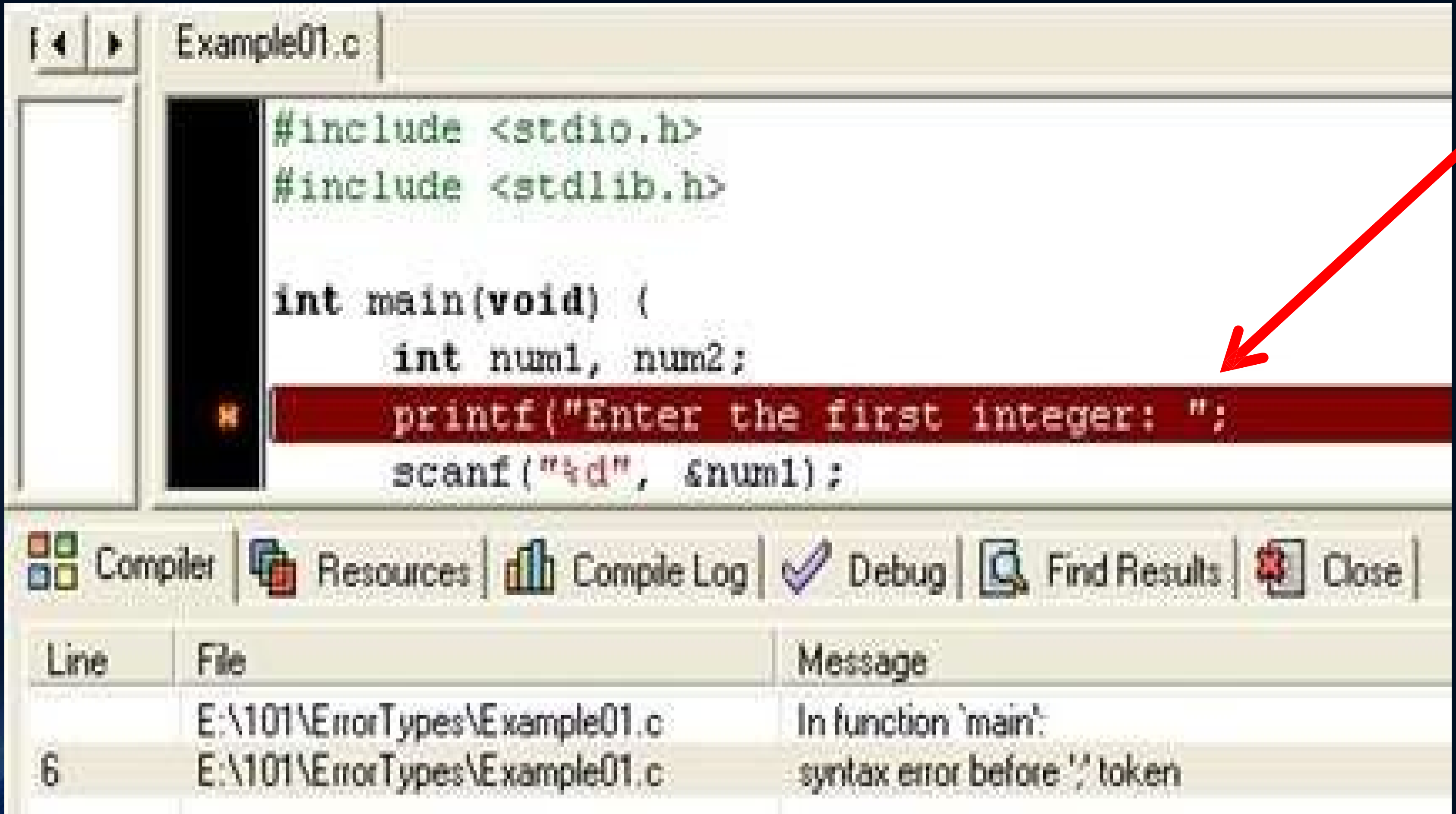


```
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
    printf("Enter a positive number (of type double): ");
    scanf("%lf", &num);
    printf("num = %f\n", num);
    printf("Square of %f = %f\n", num, num * num);

    system("PAUSE");
    return 0;
}
```

Line	File	Message
	C:\Documents and Settings\Administr...	In function `main':
7	C:\Documents and Settings\Administr...	`num' undeclared (first use in this function)
		(Each undeclared identifier is reported only once
		for each function it appears in.)

Examples

3. Missing closing parenthesis,) , in the printf satatement:



```c
#include <stdio.h>
#include <stdlib.h>

int main(void) {
      int num1, num2;
      printf("Enter the first integer: ";
      scanf("%d", &num1);
```

Line	File	Message
	E:\101\ErrorTypes\Example01.c	In function `main':
6	E:\101\ErrorTypes\Example01.c	syntax error before ';' token

Logical Errors

The errors in the logic of the program are called logical error. The compiler cannot detect logical errors. A program with logical errors is compiled (translated) and run successfully but it does not give correct result.

Logic errors are the hardest to find and fix because:

- *The compiler does not detect these errors*
- *There is no indication of error when the program is executed.*
- *The program may produce correct results for some input data and wrong results for other input data.*

Examples

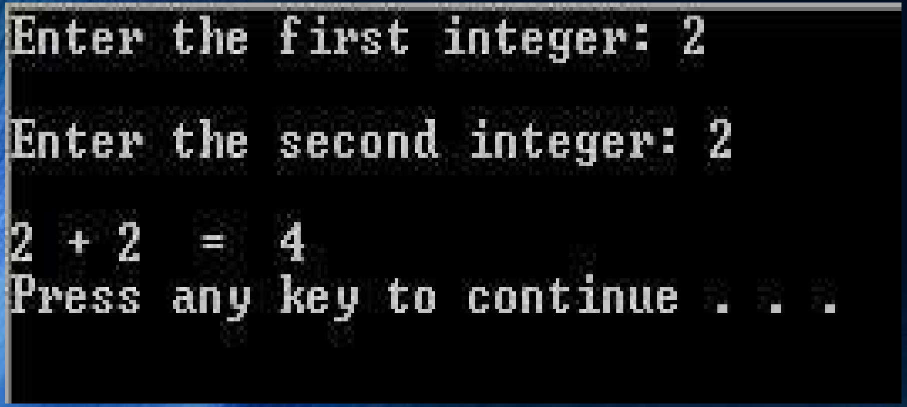
Example: The following program has a logic error in the 3rd printf statement

```
#include <stdio.h>
  #include <stdlib.h>
   int main(void)
  {  int num1, num2;
 printf("Enter the first integer: ");
   scanf("%d", &num1);
 printf("\nEnter the second integer: ");
   scanf("%d", &num2);
   printf("\n%d + %d =  %d\n", num1, num2, num1 * num2);
      system("PAUSE");
     return 0;
 }
```

Results

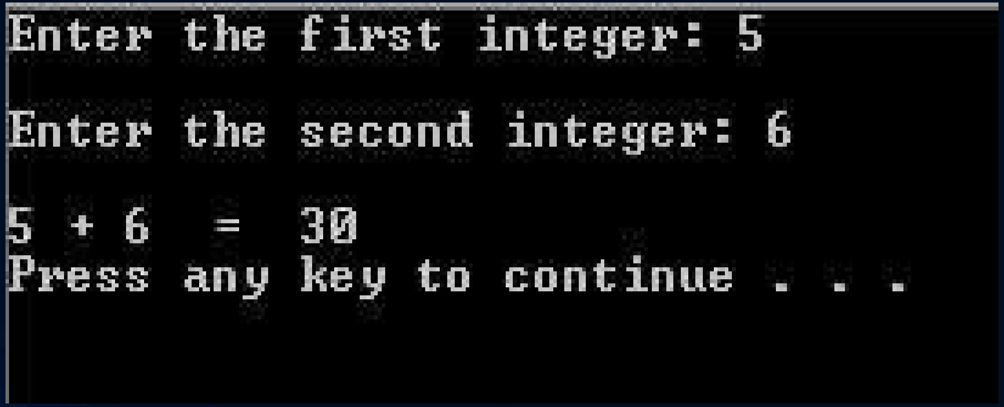
Notice that the program produces correct results for some input (num1 = 2, num2 = 2
and num1 = 0, num2 = 0) and wrong results for others:

Result 1



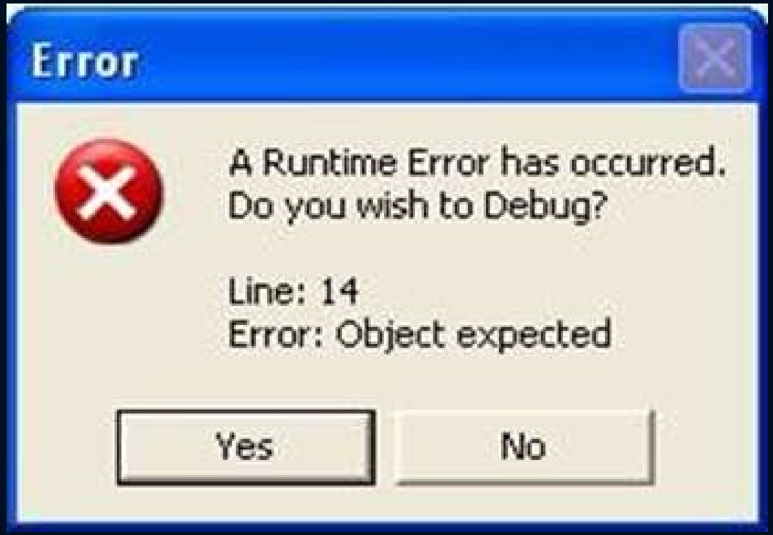
```
Enter the first integer: 2

Enter the second integer: 2

2 + 2  =  4
Press any key to continue . . .
```

Result 2



```
Enter the first integer: 5

Enter the second integer: 6

5 + 6  =  30
Press any key to continue . . .
```

Runtime Errors



A Runtime Error has occurred.
Do you wish to Debug?

Line: 14
Error: Object expected

The errors that occur during the execution of program are called the runtime errors. These types of errors may occur due to the following reasons.

- *When the program attempts to perform an illegal operation such as dividing a number by zero.*

- *If input data given to the program is not in a correct format or input data file is not found in the specified path.*

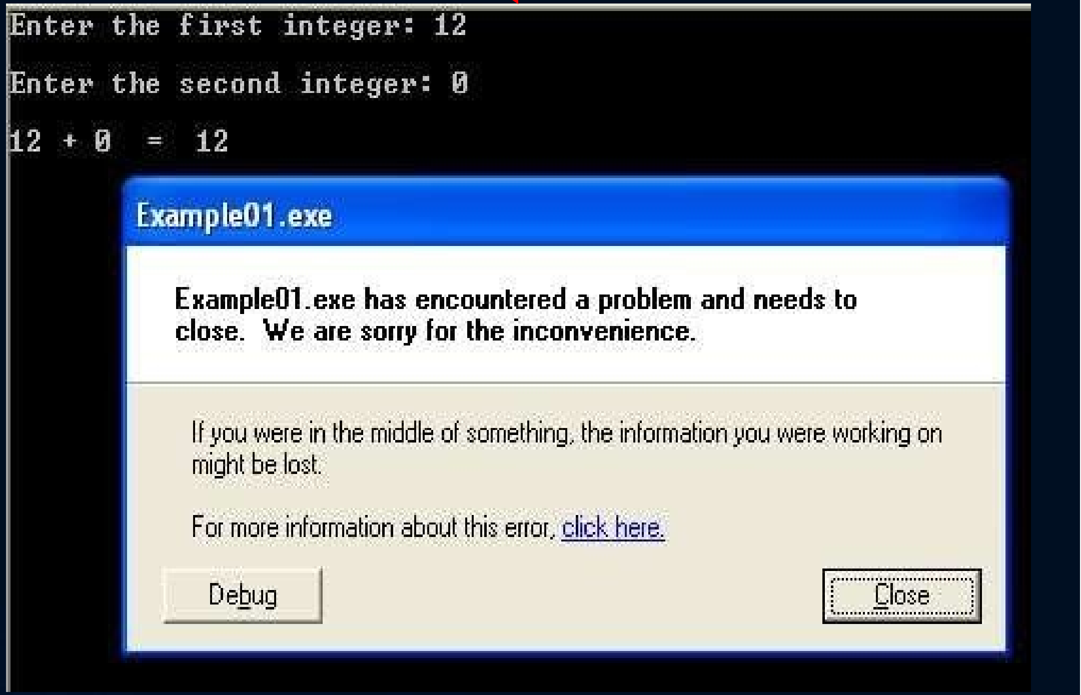
- *If hardware problem occurs such as hard disk error, or disk full or printer error etc.*

When a runtime error occurs, the computer stops the execution of program and displays an error message.

Examples

Example: When the following program is executed and the user inputs 0 for num2, a run-time error occurs due to division by zero in the expression num1 / num2

```
 #include <stdio.h>
  #include <stdlib.h>

  int main(void)
     {  int num1,
     num2;
  printf("Enter the
     first integer:
     ");
  scanf("%d",
     &num1);
     printf("\nEnter the second integer: ");
     scanf("%d", &num2);
     printf("\n%d + %d  =  %d", num1, num2, num1
+ num2);
  printf("\n%d / %d  =  %d", num1, num2, num1
/ num2);
     printf("\n%d * %d  =  %d\n", num1, num2,
 num1 * num2);
         system("PAU
     SE");  return 0;
      }
```



```
Enter the first integer: 12

Enter the second integer: 0

12 + 0  =  12
```

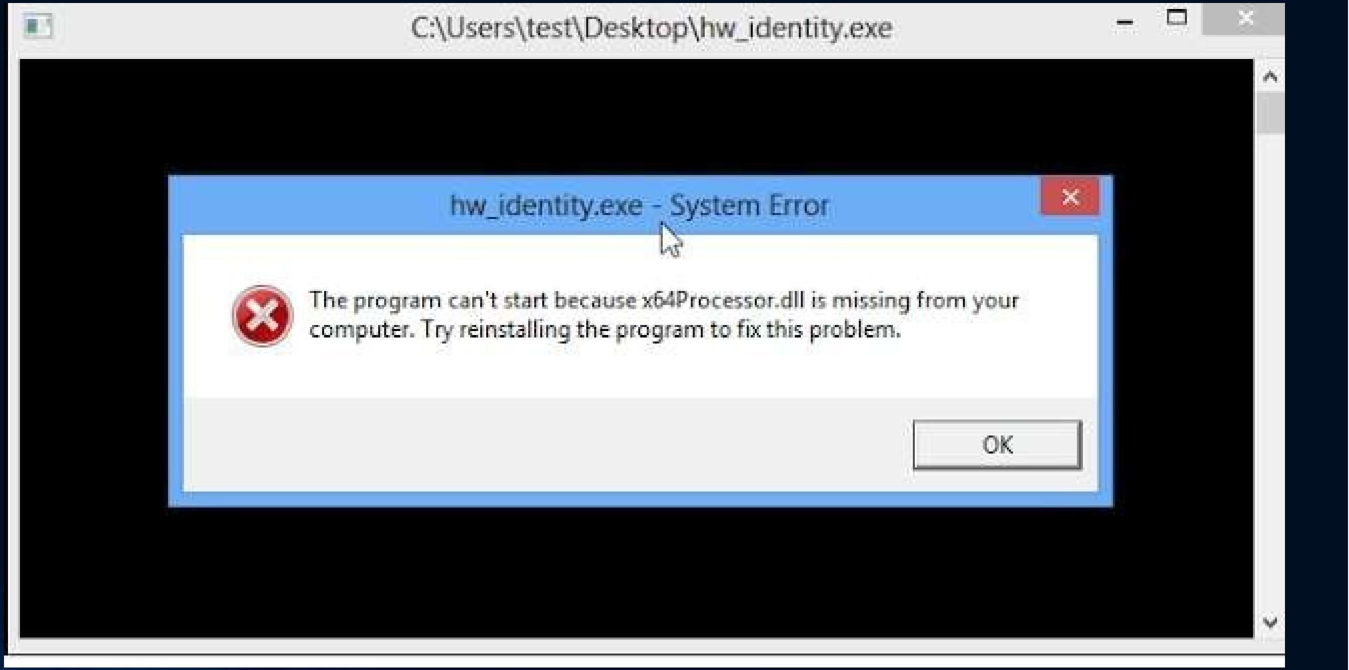
Example01.exe

Example01.exe has encountered a problem and needs to close. We are sorry for the inconvenience.

If you were in the middle of something, the information you were working on might be lost.

For more information about this error, click here.

Linker errors



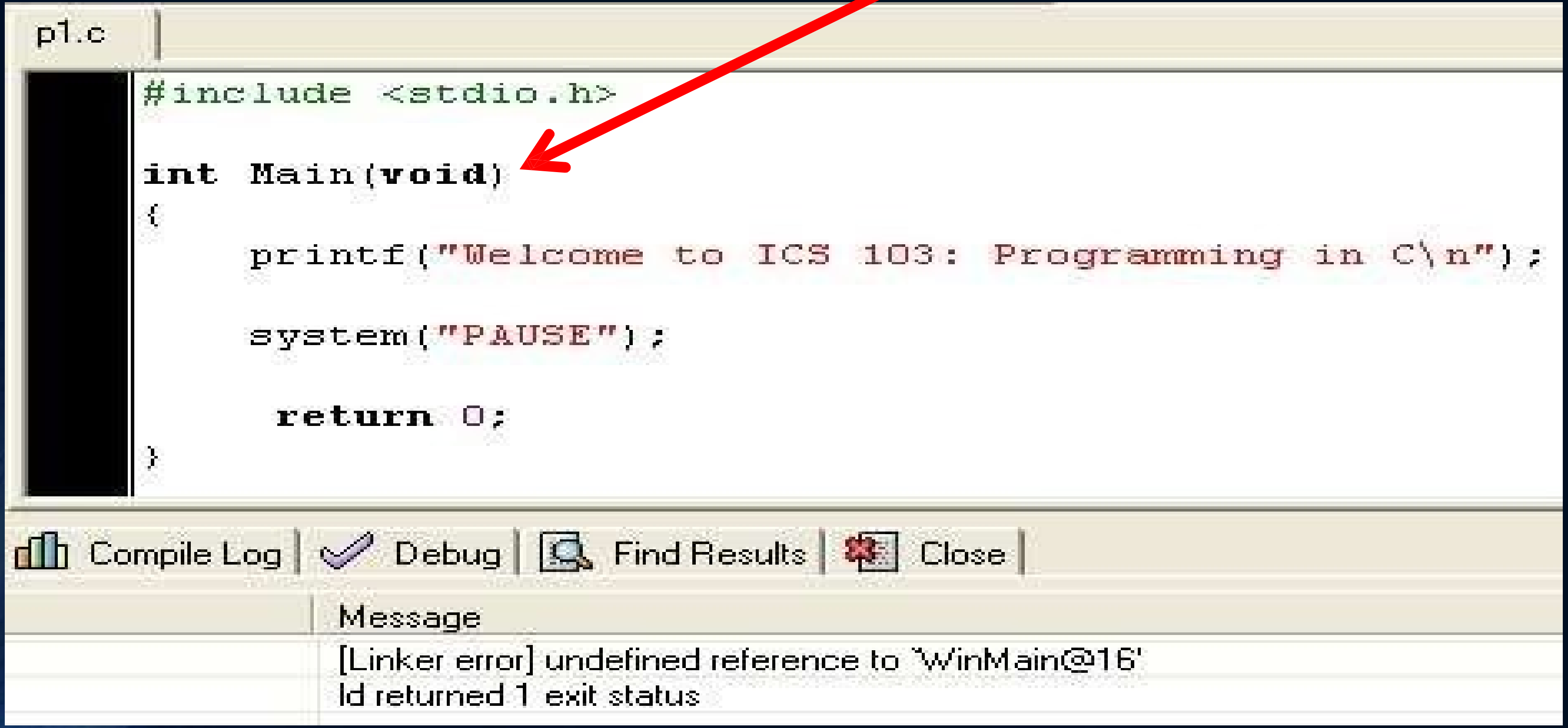
hw_identity.exe - System Error

The program can't start because x64Processor.dll is missing from your computer. Try reinstalling the program to fix this problem.

Linker errors are generated when the linker encounters what looks like a function call; but it cannot find a function with that name. This is usually caused by misspelling a C standard function (like main) or not including the header file for a function.

Examples

1. *Misspelling a standard C function:*

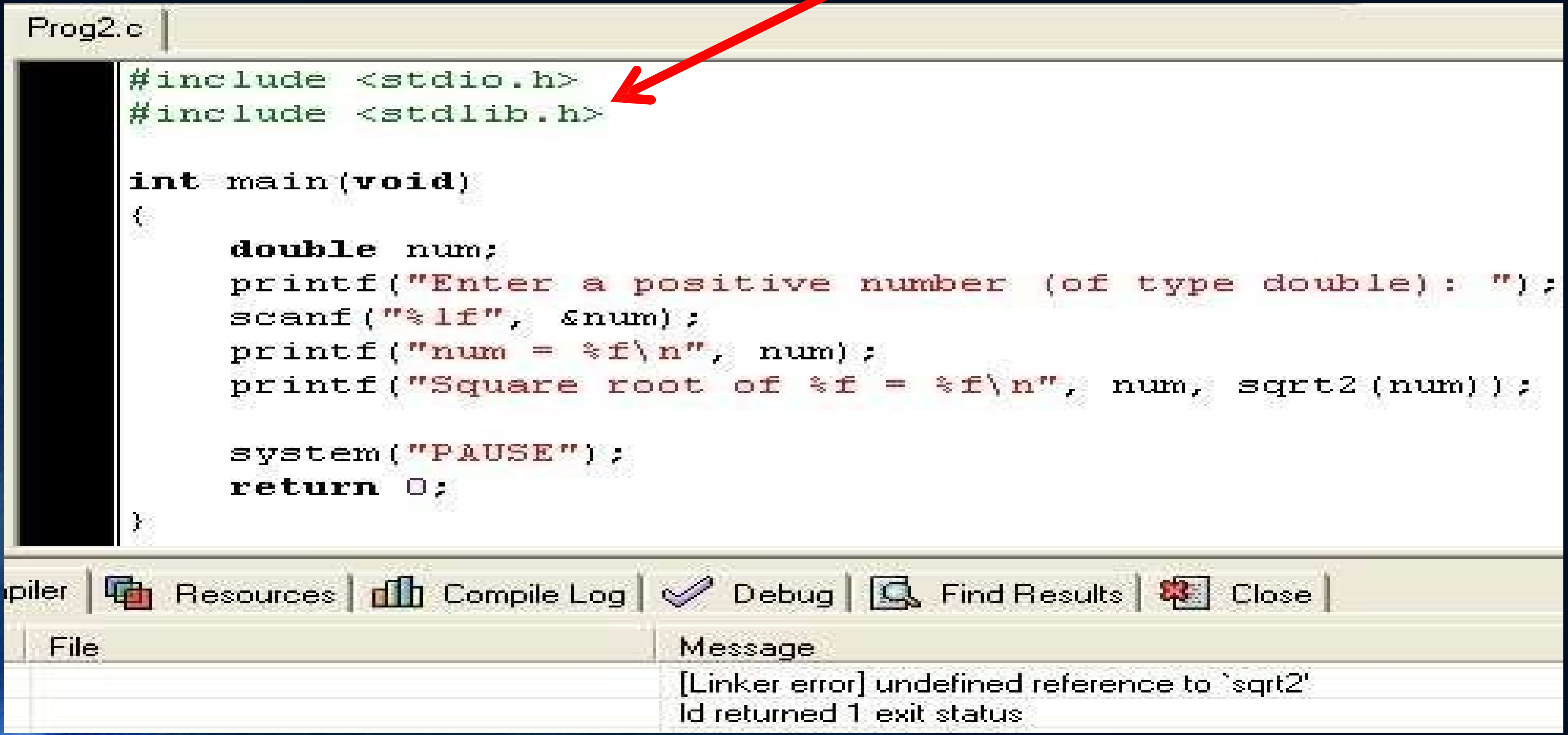


```
#include <stdio.h>

int Main(void)
{
    printf("Welcome to ICS 103: Programming in C\n");

    system("PAUSE");

     return 0;
}
```

```
Message
[Linker error] undefined reference to `WinMain@16'
ld returned 1 exit status
```

Examples

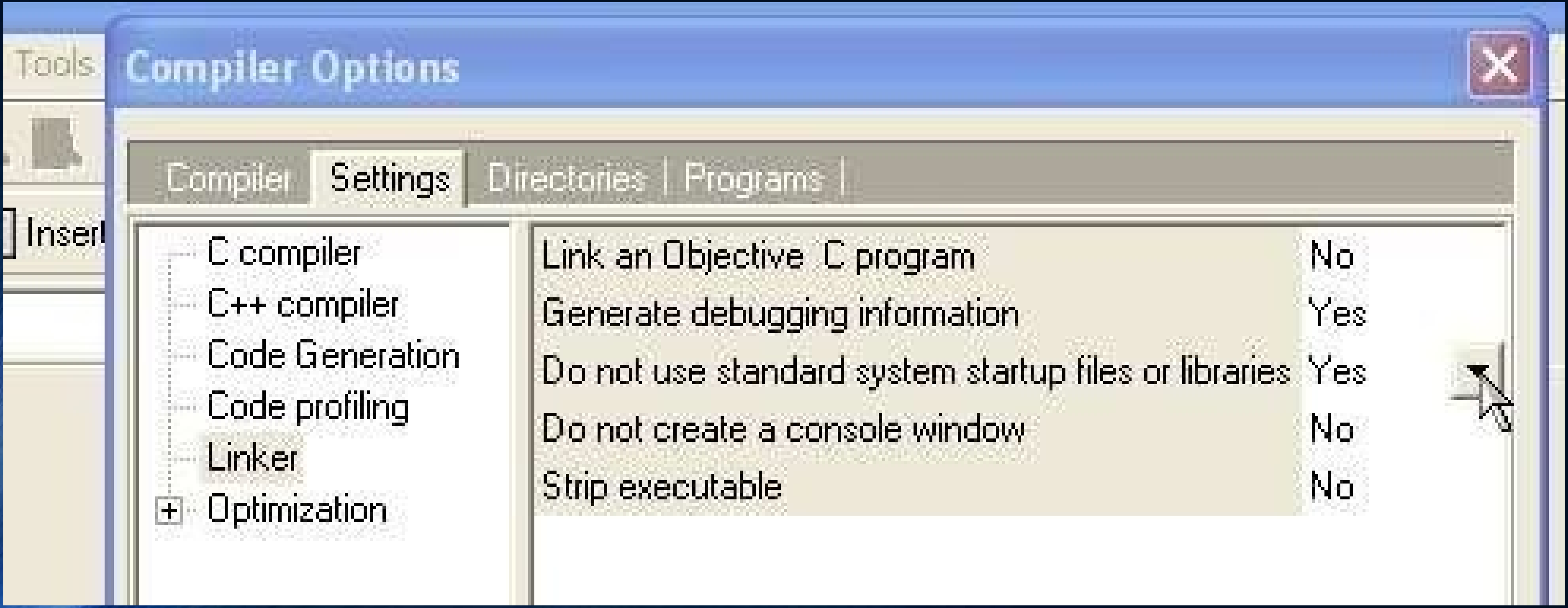
2. *Not including the header file for a function:*



```
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
      double num;
      printf("Enter a positive number (of type double): ");
      scanf("%lf", &num);
      printf("num = %f\n", num);
      printf("Square root of %f = %f\n", num, sqrt2(num));

      system("PAUSE");
      return 0;
}
```

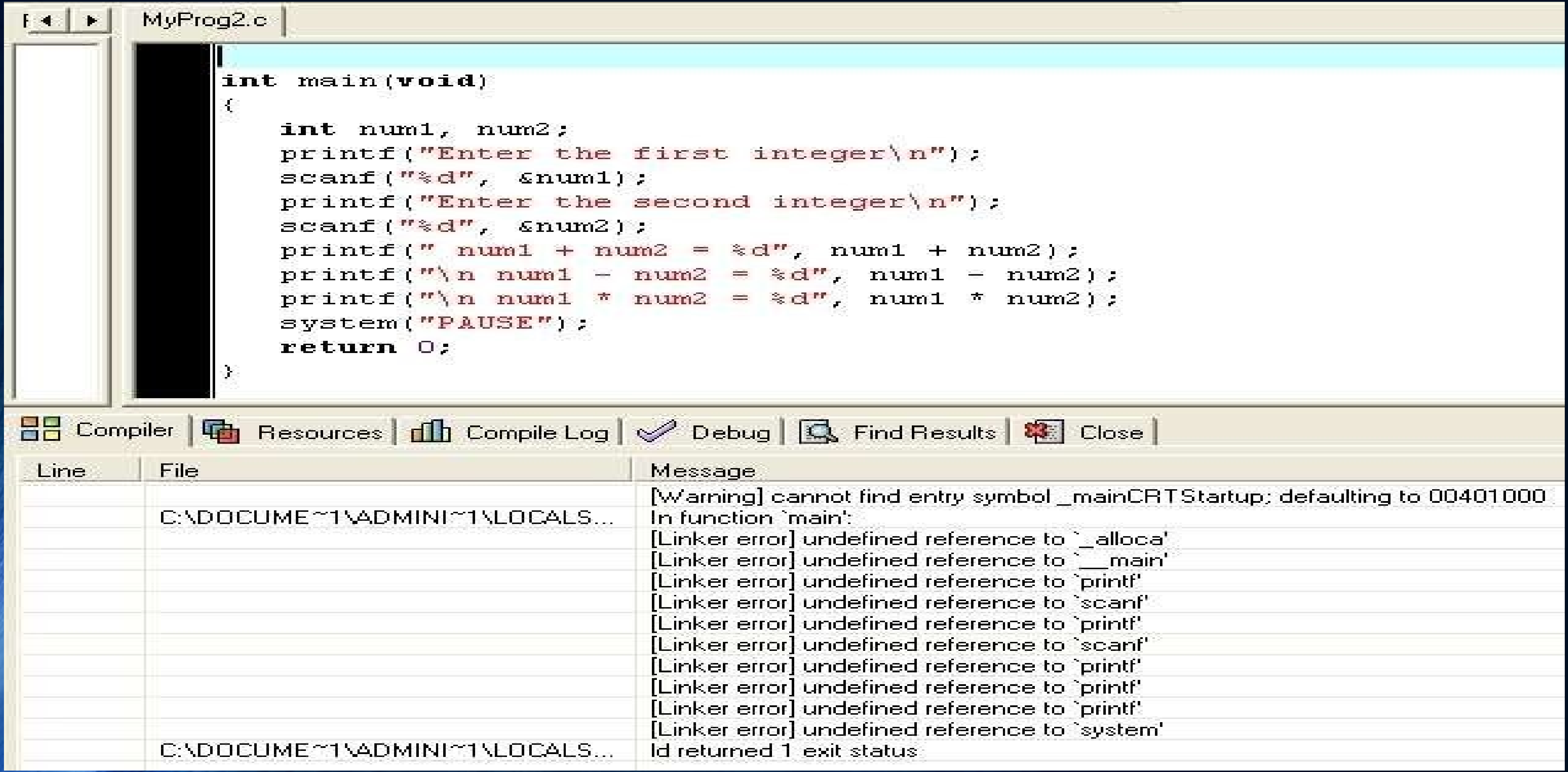
```
[Linker error] undefined reference to `sqrt2'
ld returned 1 exit status
```

*Dev C++ has an option that automatically includes standard header files. It is not recommended to use this option, as it is CPU-intensive. To turn off this option use the menus: **Tools Compiler Options Settings Linker** and set "Do not use standard system startup files or libraries" to Yes:*



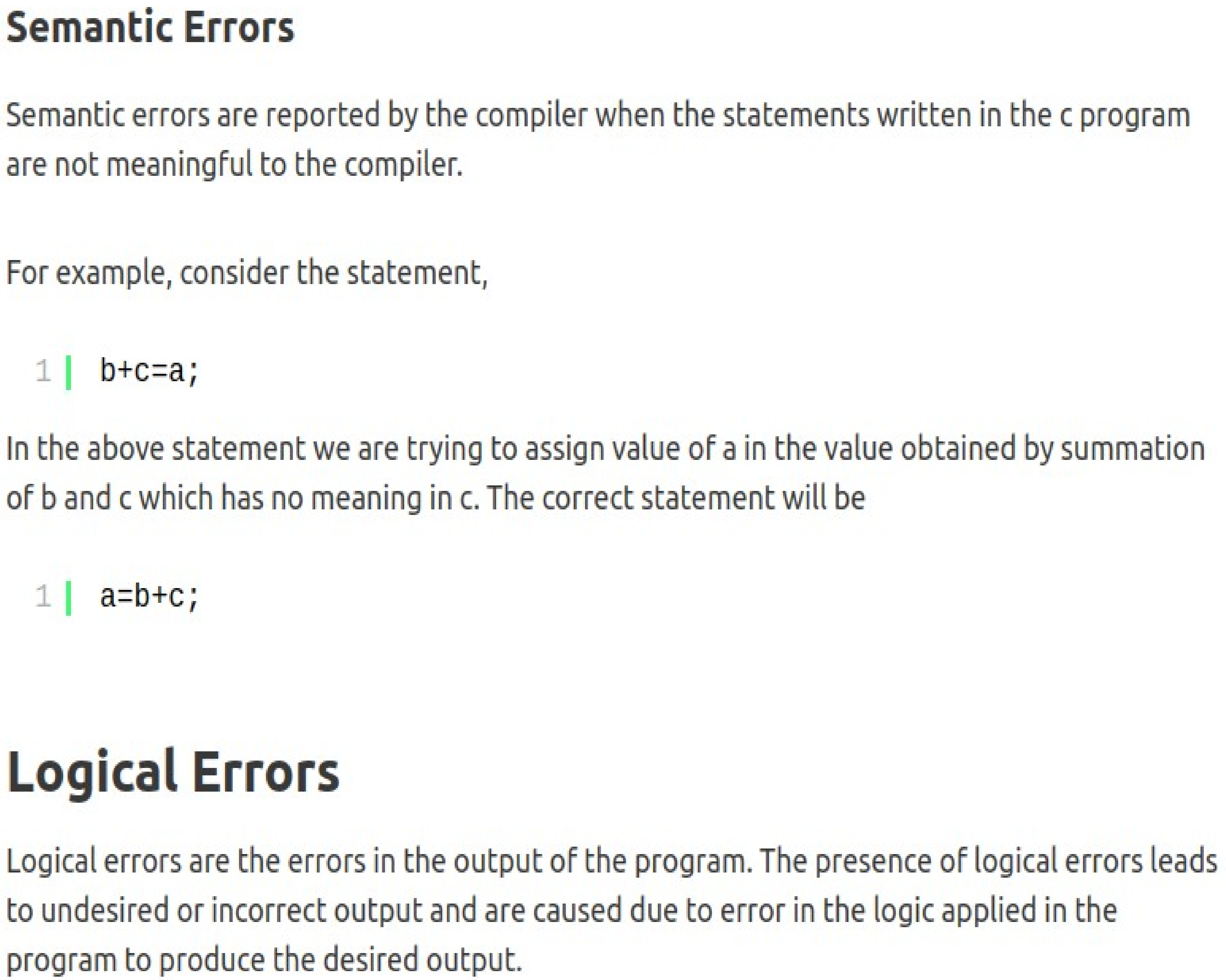
Link an Objective C program	No
Generate debugging information	Yes
Do not use standard system startup files or libraries	Yes
Do not create a console window	No
Strip executable	No

With the option turned off, not including a standard header file in your program will generate a linker error:



```
int main(void)
{
    int num1, num2;
    printf("Enter the first integer\n");
    scanf("%d", &num1);
    printf("Enter the second integer\n");
    scanf("%d", &num2);
    printf(" num1 + num2 = %d", num1 + num2);
    printf("\n num1 - num2 = %d", num1 - num2);
    printf("\n num1 * num2 = %d", num1 * num2);
    system("PAUSE");
    return 0;
}
```

Line	File	Message
		[Warning] cannot find entry symbol _mainCRTStartup; defaulting to 00401000
	C:\DOCUME~1\ADMINI~1\LOCALS...	In function `main':
		[Linker error] undefined reference to `_alloca'
		[Linker error] undefined reference to `__main'
		[Linker error] undefined reference to `printf'
		[Linker error] undefined reference to `scanf'
		[Linker error] undefined reference to `printf'
		[Linker error] undefined reference to `scanf'
		[Linker error] undefined reference to `printf'
		[Linker error] undefined reference to `printf'
		[Linker error] undefined reference to `printf'
		[Linker error] undefined reference to `system'
	C:\DOCUME~1\ADMINI~1\LOCALS...	ld returned 1 exit status



Semantic Errors

Semantic errors are reported by the compiler when the statements written in the c program are not meaningful to the compiler.

For example, consider the statement,

```
b+c=a;
```

In the above statement we are trying to assign value of a in the value obtained by summation of b and c which has no meaning in c. The correct statement will be

```
a=b+c;
```

Logical Errors

Logical errors are the errors in the output of the program. The presence of logical errors leads to undesired or incorrect output and are caused due to error in the logic applied in the program to produce the desired output.