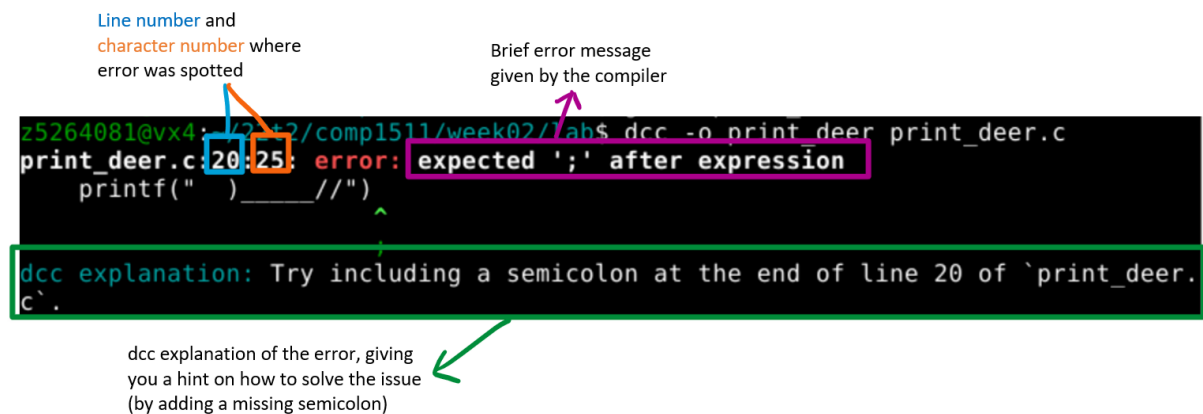


Guide to autotest reports and error messages

Error messages

- The best way to solve errors when you get an error is to **read the entire error message**. It will give you a lot of information about the error and often will straight up give you the solution to it.
- Look at the line number and character number where the computer detected the error. These come immediately after the file name, and have this format: <filename>:<line number>:<character number>. The line number is usually the most important of these – you can start your search for the bug from this line, rather than looking through the entire file.
- Read the dcc explanation – it's been made to give a more detailed explanation of the error and give hints on how to solve it.

You can see these different parts of the error message here:



The screenshot shows a terminal window with a compiler error. The error message is: `print_deer.c:20:25: error: expected ';' after expression`. The error is highlighted with a pink box. Below the error message, the dcc explanation is shown: `dcc explanation: Try including a semicolon at the end of line 20 of 'print_deer.c'.` The explanation is highlighted with a green box. Annotations with arrows point to specific parts of the error message: a blue arrow points to the line number '20', an orange arrow points to the character number '25', a pink arrow points to the error message 'expected ';' after expression', and a green arrow points to the dcc explanation.

```
z5264081@vx4: /2+2/comp1511/week02/7ab$ gcc -o print_deer print_deer.c
print_deer.c:20:25: error: expected ';' after expression
printf(" )____//")
                    ^
dcc explanation: Try including a semicolon at the end of line 20 of 'print_deer.c'.
```

Line number and character number where error was spotted

Brief error message given by the compiler

dcc explanation of the error, giving you a hint on how to solve the issue (by adding a missing semicolon)

Autotest Reports

- Again, your best course of action is to **read the entire autotest report** for any failed tests. There are some areas that can give you the best idea on why the test failed, but getting used to the structure of the complete autotest report will be very helpful. A basic autotest report will look something like this:

```
Test addition_3 (./addition) - failed (Incorrect output)
Your program produced this line of output:
Enter the number of students and tutors: 5 + 5 = 10

The correct 1 lines of output for this test were:
Please enter the number of students and tutors: 5 + 5 = 10

The difference between your output(-) and the correct output(+) is:
- Enter the number of students and tutors: 5 + 5 = 10
+ Please enter the number of students and tutors: 5 + 5 = 10

The input for this test was:
5 5

You can reproduce this test by executing these commands:
gcc -o addition addition.c
echo '5 5' | ./addition
```

- Look for the start of an individual test and read from there, otherwise it's easy to get overwhelmed by the wall of text. The first line of an individual test report will look like this:

```
Test addition_2 (./addition) - failed (Incorrect output)
```

You can see that this line includes the word “Test”, the name of the executable in brackets, and the word “failed” (followed by the reason the test failed).

- Look at the difference between your output and the correct output – often there is only a single word, or even just a full stop that is different!

```
Your program produced this line of output:
Enter the number of students and tutors: 5 + 5 = 10

The correct 1 lines of output for this test were:
Please enter the number of students and tutors: 5 + 5 = 10
```

Can you see the tiny difference that caused this test to fail?

- The “-” and “+” show which lines of your output are different from the correct output. The “-” means that you must remove this line for your output to be correct. The “+” means you must add this line for your output to be correct.

On some computers, the colours aren't working, so the entire autotest output will be in white text. You'll still be able to find all the same information, it might just be a bit more difficult to find the sections you want to look at (all the more reason to read through the entire test report!)