Students may use their text book, any hand written notes, any digital notes or programs written by the student, and the internet.

Wednesday, October 19th – 2016

6:00pm – 8:45pm – Be sure to attempt questions or programming tasks. Partial Credit will be awarded.

120 POints

**SHORT ANSWER SECTION**

1. (5 Points) Name 5 different code blocks we covered in class to this point.

**switch case**

**main**

**if**

**else**

**else if**

2. (10 points) a) Describe the purpose of a loops in programming.  
**Loops allow the programmer to perform a set of procedures multiple times.**

b) Compare and Contrast indefinite and definite loops. Describe why for loops are conventionally definite loops but can be structured as an indefinite loop as well. Describe how a while loop can be both definite or indefinite. Be as detailed as possible.

**Definite loops can be used if the programmer knows (or will know at runtime) the exact number of times a loop should be run. Indefinite loops are for the opposite case. For loops are typically descriptive of definite loops. For loops typically involve a count or counter and the loop will run until the count or counter has been filled. In this example, there are is a specific number of times the loop will run e.g. *for* all items in a list. For loops can also be indefinite. In C++ one form of the indefinite loop is *for ( ; ; )*.**

**While loops are typical of indefinite loops. They are used when the programmer is not sure how many times a loop will need to be run. They typically included a true/false condition as opposed to a counter, though counters can also be used *e.g. while (True)*. Though usually indefinite, an example of a definite while loop is while (x = 1; x < 100; x++).**

3. (5 Points) Please define source code, machine code, and compiler. Describe in your own words how the 3 are related to creating programs in C++.

***Source code* describes the lines of code a programmer wrote to create a piece of software. This code is what is compiled to create the application.**

***Machine Code* is the actual ones and zeros that correspond to the switches on a computer’s CPU. It is the low level and describes the instructions to the system’s processor.**

**A *compiler* is what turns the source code into an application. It takes all of the lines of the source code and translates it into machine code that the computer can run and an executable file for users.**

4. (10 Points) – a) Using || to connect multiple Boolean expressions, does it lessen or widen the scope of the Boolean Expression returning a true value? Explain in as much detail as possible.

**Using the pipes widens the scope of the Boolean. In the case of an OR, which is what the pipes represent, any of its components makes the statement true. The chance of the statement being true by adding ORs increases with each OR added to the statement.**

b) Using && to connect multiple Boolean expressions, does it lessen or widen the scope of the Boolean Expression returning a true value? Explain in as much detail as possible.

**Using two ampersands lessens the scope of the Boolean. It makes it more difficult for a TRUE value to be returned. In order for a statement involving AND, which is what the ampersand represents, to be TRUE, all of the items must be TRUE.**

5. (5 Points) Give a couple examples of C++ programming conventions, list how the conventions support programming. Give 3 examples of C++ Syntax rules.

**Conventions:**

1. ***variables should not have dual meanings*- Using this convention improves readability of code and lessens any confusion for any maintainers that may come after you.**
2. ***loop variables should be declared immediately before the condition*- This convention keeps code compartmentalized and less confused. If someone was reading the code from top to bottom, it may be daunting to see many variables declared at the beginning that won’t be used for lines of code.**
3. ***code should be laid out in blocks-*  Block layouts show, with tabs and line breaks, which code relates to which section.**

**Syntax:**

1. **Statements end with a semicolon (;)**
2. **Every opening symbol will have a closing symbol (), {}, etc.**
3. **A program shall contain a global function named main, which is the designated start of the program.**

6. (5 Points) Write an English sentence that has correct syntax but incorrect *semantics*.

**Wear is the buffalo?**

Write an English sentence that has correct semantics but incorrect *syntax*.

**What chips they got at the store?**

**SHORT ANSWER SECTION Cont…**

7. a) (10 points)

**Complete Logic Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **P** | **S** | **P && S** | **P || S** | (P&&S) && (P||S) | !S || False || !P |
| **FALSE** | **TRUE** | **False** | **True** | **False** | **True** |
| **FALSE** | **FALSE** | **False** | **False** | **False** | **True** |
| **TRUE** | **FALSE** | **False** | **True** | **False** | **True** |
| **TRUE** | **TRUE** | **True** | **True** | **True** | **False** |

**SHORT PROGRAMMING TASKS:**

(10 points) TASK 1: **Execute your program as a Text File and compile/run through the command prompt or terminal.**

Create a function called **info** that takes 4 arguments. 2 arguments to hold value for names, 2 arguments to hold values for age. Print the name and age together, for values passed from **main**.

Create a **main** function that will input 2 values for name and 2 values for age. Pass the values into the function called **info**.

Print name and age together.

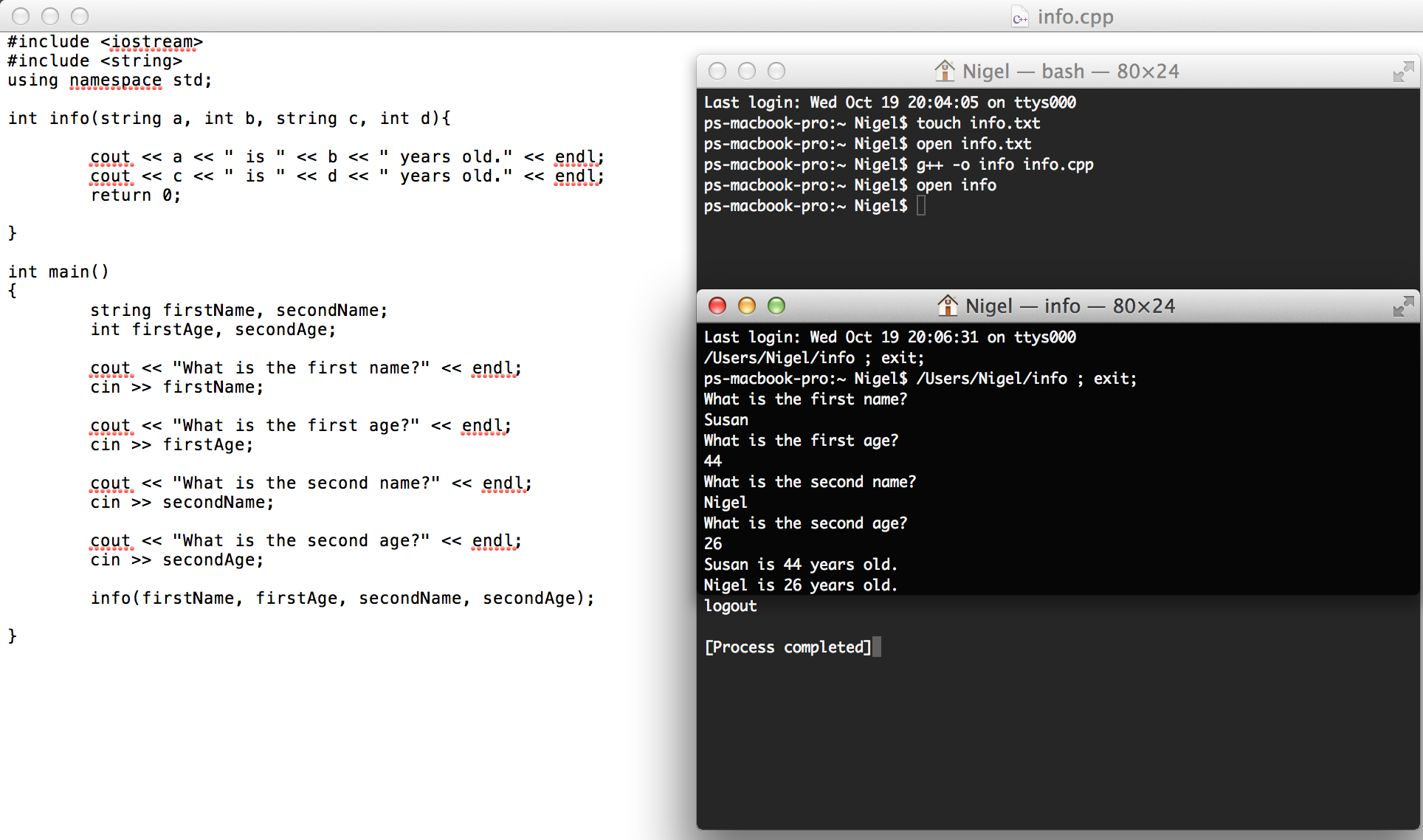
Example:

Input = susan.

Input = 44

Output = Susan is 44

(Attach snipping photo of source code and output. Output photo should be of your command prompt navigation and output in the command prompt.)



(20 points) TASK 2:) **(You may use your IDE or Text File to execute program.)**

Using &&, || operators) Write a program that prompts the user to enter an integer and determines whether it is divisible by 5 and 6, and whether it is divisible by 5 or 6.

Place the above code into a do-while / switch case menu.

Allow the user to choose which operation she will perform.

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**\* Main Menu: \***

**\* Enter # to run program or Quit \***

**\* 1)** Is Input divisible by 5 and 6 **\***

**\* 2)** Is Input divisible by 5 or 6 **\***

**\* 3) Quit \***

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

1)

Enter an integer:

10 // Users enters 10

Is 10 divisible by 5 and 6? No

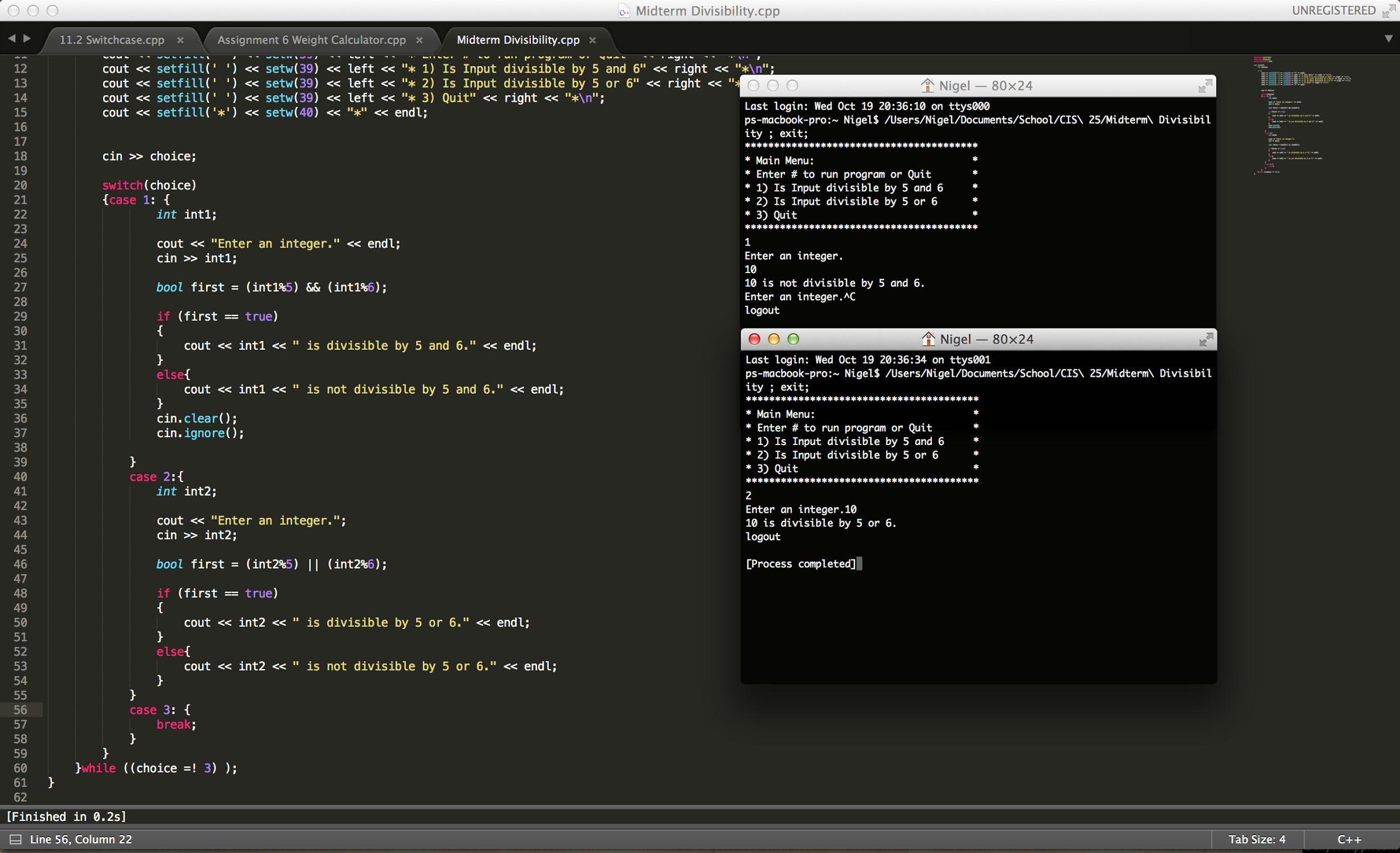
2)

Enter an integer:

10 // Users enters 10

Is 10 divisible by 5 or 6? Yes

(Attach snipping photo of source code and output.)



**(You may use your IDE or Text File to execute program.)**

(30 points) TASK 3: Use differentloops to **print the odd / negative numbers -1 to -101.** All programs will print the same output in the same order.

1. Using a **for** loop that increments the loop control variable by 2 each iteration
2. Using a **for** loop whose loop control variable goes from 0 to 50.
3. Using a **for** loop whose loop control variable goes from 100 down to 0.
4. Using an infinite **for** loop with no conditional expression and exiting the loop with a **break** statement.
5. Using a **while** loop.
6. Using a **do-while** loop.

There should be a minimum 6 different Snipping photos. One photo for each program A – F.

Make sure to capture both source code and output for all 6.

-1

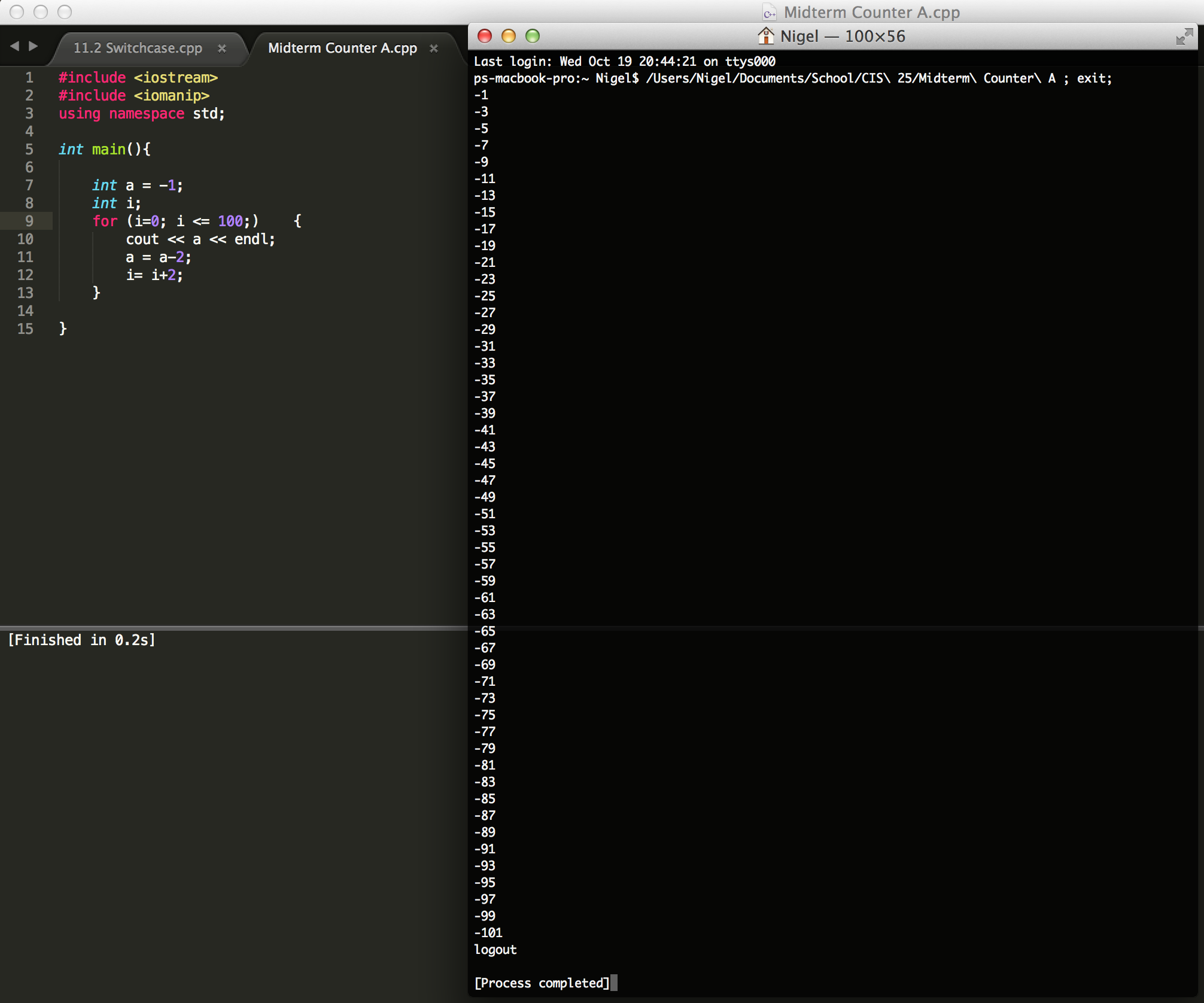
-3

-5

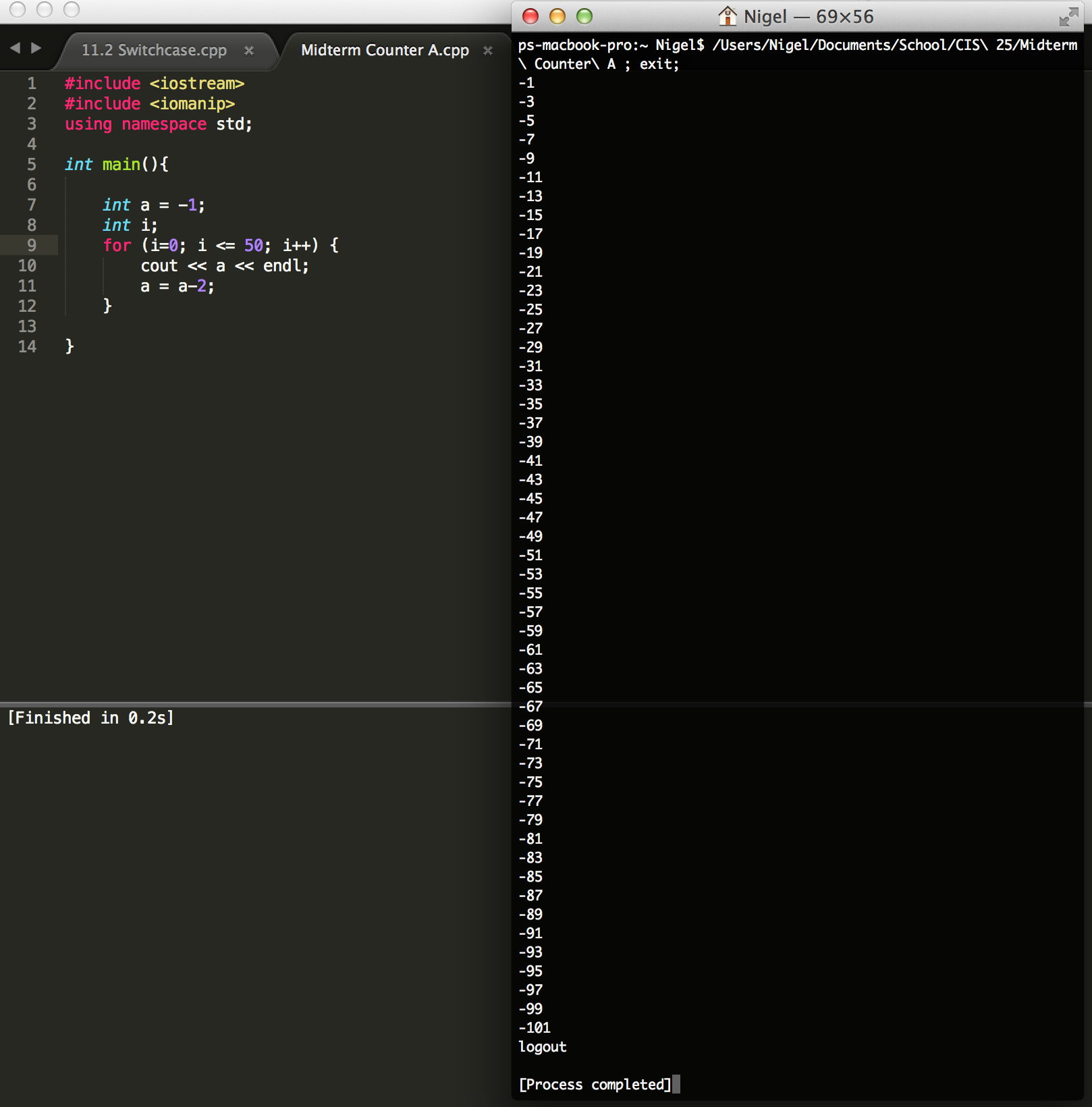
-7

…

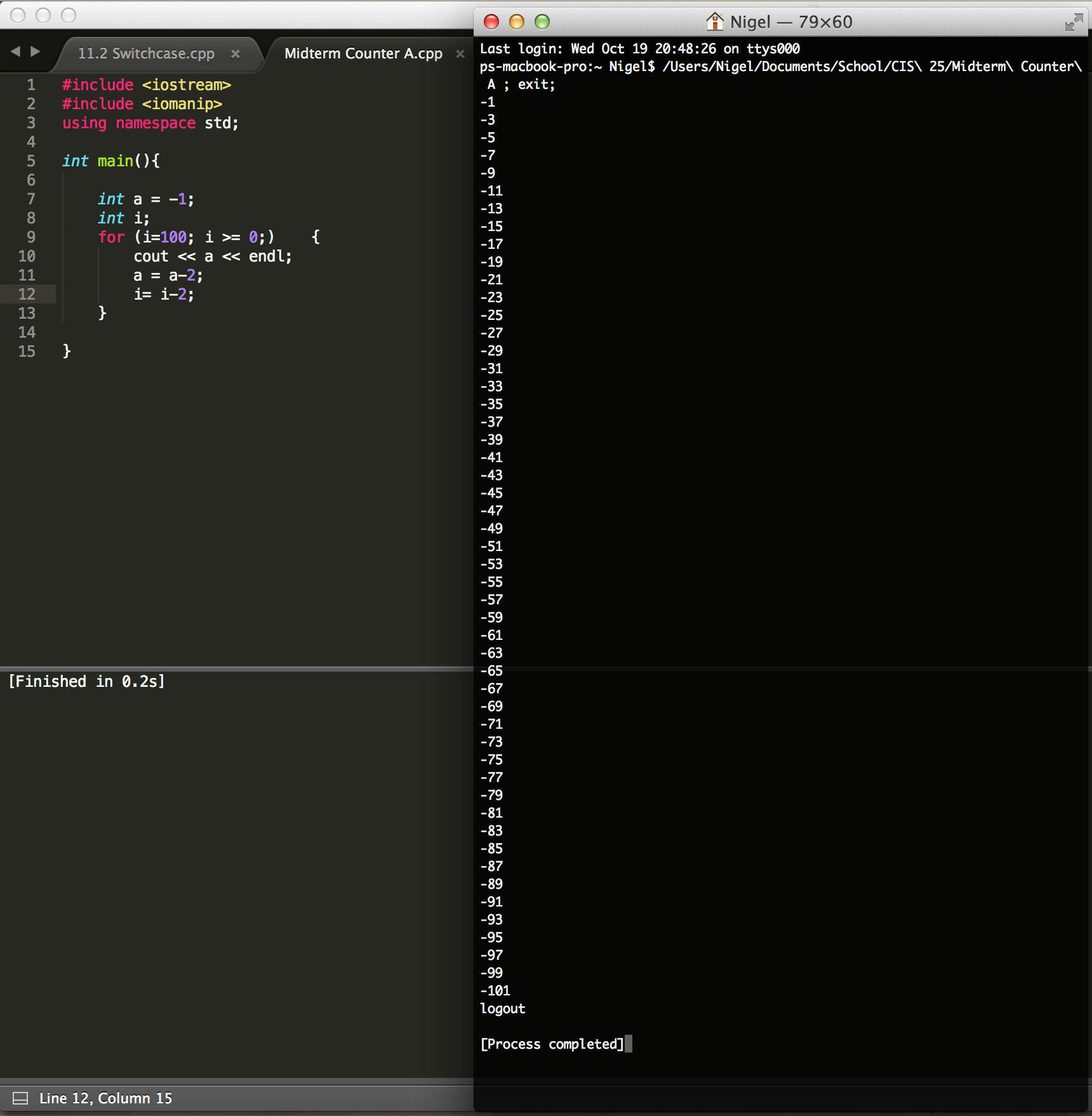
-101



^^^A



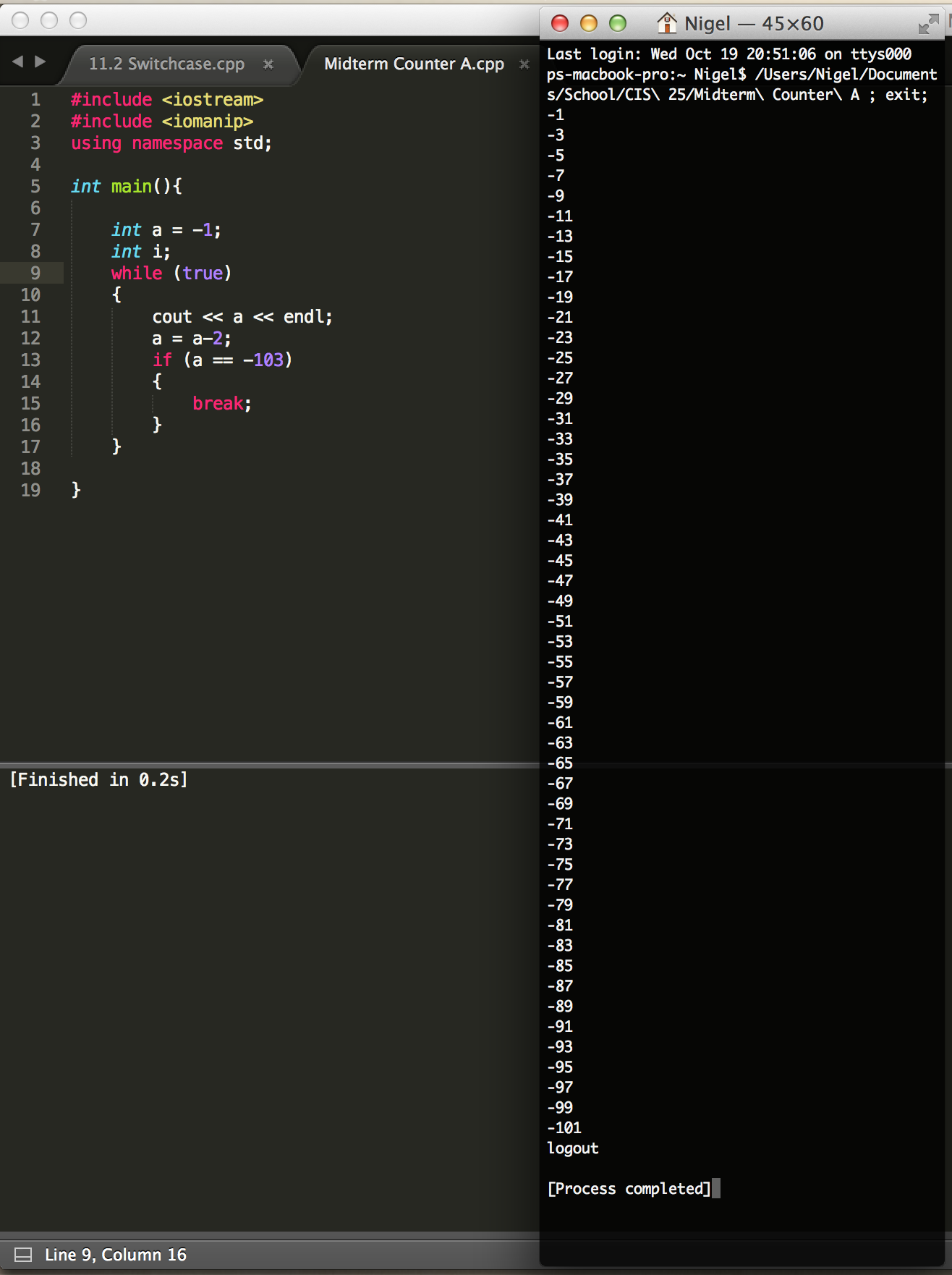
B ^^^



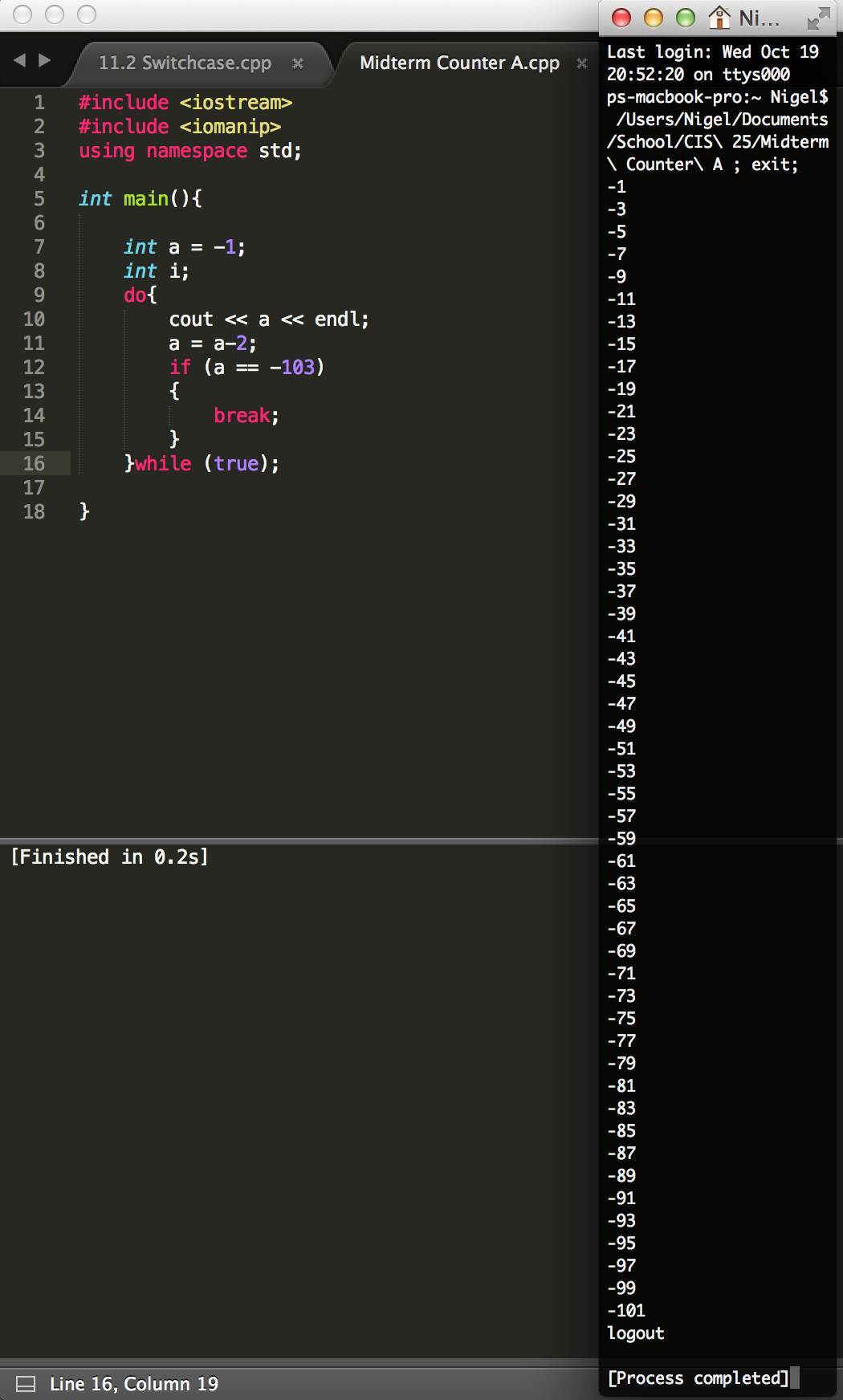
C^^^



D^^^



E^^^



F^^^