

Due 2:00PM, Thursday, March 5th, 2009

---

Name \_\_\_\_\_

CSE Login \_\_\_\_\_

**Instructions** Follow instructions *carefully*, failure to do so may result in points being deducted. For written questions: type your answers; for programming code, use a monotype font (like `Courier New`). Print your answers hardcopy and staple this cover page to the front of your assignment. Be sure to write your name on the front.

For each of the programs, place your source code into files named `homework03program01.c`, `homework03program02.c`, etc. and turn them in using the webhandin available on the course webpage. You do not need to print out your source code for the programs.

You may discuss problems with your classmates, but all work must be your own. The CSE academic dishonesty policy is in effect (see [http://www.cse.unl.edu/undergrads/academic\\_integrity.php](http://www.cse.unl.edu/undergrads/academic_integrity.php)).

Question	Points	Score
1	10	
2	10	
3	20	
4	20	
5	20	
6	20	
Total:	100	

1. 10 points For each of the following, write an equivalent compound assignment operator.
- (a) `x = x + 2;`
  - (b) `z = z + r * m;`
  - (c) `m = m * y + 1;`
  - (d) `x = x - (a + b - c);`
  - (e) `total = 5 * total;`

2. 10 points Write a `for` loop *and* a `while` loop to compute the following: given two integers, `x`, `y` compute the sum of all integers between `x` and `y` not including `x` and `y` (for example, if `x = 5` and `y = 9`, the result would be 21). If `x, y` are such that there are no integers between them, output zero.

3. 20 points **Program 1**

Write a program that asks the user to enter a positive integer. If the input is invalid, keep asking for valid input until the user enters a positive integer. Once the user has input a positive integer `n`, output a multiplication table for values up to `n`. For example, if `n = 4`, your output should look something like this:

1	1	2	3	4	
2	1	1	2	3	4
3	2	2	4	6	8
4	3	3	6	9	12
5	4	4	8	12	16

Be sure to format your output so that the columns line-up nicely.

4. 20 points **Program 2**

Write a program that determines how long it will take a towns population to reach a certain number. Your program will ask the user for two values: a starting population and an ending population. Assuming that the population increases by 10 percent each year, your program should use a loop to determine how many years it will take for the population to surpass the specified ending population. Output this result to the user.

5. 20 points **Program 3**

The ancient Greek mathematician Euclid developed a method for finding the greatest common divisor of two integers, `a` and `b`. His method is as follows:

1. If the remainder of `a/b` is 0 then `b` is the greatest common divisor.
2. If it is not 0, then find the remainder `r` of `a/b` and assign `b` to `a` and the remainder `r` to `b`.
3. Return to step (1) and repeat the process.

Write a program that uses a function to perform this procedure. Display the two integers and the greatest common divisor.

6. 20 points **Program 4**

The standard math library provides a function to compute the square root of a number. In this exercise, we'll be writing our own function. Given a number  $n$ , the square root can be approximated by repeated calculation using the formula,

$$x_i = \frac{1}{2} \left( x_{i-1} + \frac{n}{x_{i-1}} \right)$$

where  $x_1 = 1.0$ . We continue to compute  $x_i$  until  $|x_i - x_{i-1}| \leq 0.005$ . Write a function to interpolate the square root according to this method. Write a program to read an input from the user, then output the value, the square root using the standard math library function and the square root using your function.