

Project 2

Cornucopia of Functions

Ece 131 - Fall 2017

Design Doc Due - Tuesday **10/17/2017** by 11:59 PM

Project Due - Tuesday **10/24/2017** by 11:59 PM

Note: Start early! - this project will be challenging and take some time and effort. Seek out help during office hours/by email if you need help. Also there is a tutor on Fridays and Saturdays, check Learn for the details.

Ethics Reminder:

- Everyone needs to submit their own work!
- In no circumstances is it O.K. to look at someone else's code, or look for code in Internet, or copy any portion of it. THIS IS CHEATING!
- Looking online for answers is also cheating. If you can find it, I promise you, we will find that solution too, and YOU WILL GET A ZERO.
- You may discuss algorithms at a high level of how you solved a problem, but you may not share code - including verbally.
- Ask the TAs or Professor if you have any questions or concerns with regards to the ethics policy, which is also clearly stated in the syllabus.

Overview: Extensive practice using functions and loops. We have provided you with a starter file with several empty function bodies. Your job is to implement these functions correctly. You will also implement an interactive mode where a user can tell your program which of these functions they want to execute. The program will continually ask for new input, until the user selects "quit".

Design Document:

Your design document is due Tuesday 10/17/2017. This should include inputs and outputs for each function, along with the algorithm for each. There are five functions, but we are giving you 2 of the algorithms, so you need to turn in, the algorithms for three functions and the algorithm you will use for the interactive() mode. These algorithms that you turn in, must match the implementation that you develop later.

Functions Overview

- a. **float cuboid_volume(float length, float breadth, float height):** Calculates the volume of a cuboid given length, breadth, and height. You should **return 0** if any of the inputs are negative.

- b. **void Print_prime_numbers(int start, int end, char delim):** * Prints all prime numbers between start and end (excluding start and end)
delim is the character to put after every prime number - for example:
if delim = ',' the output look like: 3,5,7,11 with a ',' in between each
if delim = '\n' the output looks like:

3
5
6
11

This function shouldn't return anything. Here is an algorithm you **MUST USE** for this function:

1. Loop over every value from start + 1 to end
 2. For each of those values, check if it's prime (you could write an isprime() function that returns true if the number is prime, false if not)
 - a. To check if the number n is prime, do the following:
 - b. Initialize a boolean variable "isprime" to true
 - c. Loop from 2 up to and including n/2 using variable i
 - d. If the remainder of n divided by i is 0, then the number is not prime. Set a variable "isprime" to false (or return false if using a function)
 - e. If the loop exits and isprime is false, the number is not prime, otherwise the number must be prime (return true if using a function)
- c. **bool co_primes(int num1, int num2):** Determines whether num1(≥ 0), num2(≥ 0) are co-primes or not. if yes return true, otherwise return false
- i. You **MUST FOLLOW** this algorithm for calculating the co-primes:
 - ii. Choose the smaller number num1 or num2 - call it *min*
 - iii. Loop from 1 up to and including *min* using variable *i*
 - iv. If the remainder of *num1* divided by *i* is 0 **AND** the remainder of *num2* divided by *i* is 0
 1. If *i* is greater than 1, then return false
 - v. If the loops finished, that means we did not find a common factor greater than 1, so return true

- d. **void print_design(int rows):**

Prints a pretty number triangle

when rows is 4 you should print:

1
1 2
1 2
1

when rows is 7 you should print:

```
1
1 2
1 2 3
1 2 3 4
1 2 3
1 2
1
```

e. `int power(int n, int d)`: Computes n to the power of d

Interactive Mode:

You will also implement the `interactive()` function

`void interactive()`

This function will be an **infinite** loop. It will continuously take an integer input from the user (using `scanf`). You should write a switch statement that switches based on this number and calls the appropriate function. The appropriate actions are as follows:

1. Print the prime numbers from a 'start' value to 'end' value (get from user using `scanf`)
2. Checks if two values are co-primes (get these from user using `scanf`)
3. Volume of a cuboid (get width, breadth, and height from user using `scanf`)
4. print the design with a given number of rows (get from user using `scanf`)
5. Calculates n to the power of d (get n and d from user using `scanf`)
6. Quit the program (break out of the infinite loop)

Procedure:

2. Download the starter file from Learn - `p2.c`
3. Download the automatic grading script from Learn `p2_grade.pyc`
4. Try compiling the starter file and running the test script:
5. `$ gcc p2.c -o p2.o`
6. `$ python p2_grade.pyc p2.o`
7. You should see the series of tests run - and (most) of them should Fail at first. It should also show you the expected result. Note that you'll need to match this exactly in order to pass.
8. As you develop your solution, use this test script to check your work. Note that just because you "Pass" all tests, does not guarantee your code is correct. You should verify this by writing your own tests as well.
9. Develop your solution
 - a. Your job is to fill in any functions in `p2.c` that have a "Implement Me" comment
 - b. DO NOT MODIFY `main()` or `run_tests()`

Submission details:

- Submit your design document in pdf format to Learn by 10/17/2017
- Rename your source file as p2-<NETID>.c where NETID is your own UNM netID, and submit it to Learn.
- Submit the source file (p2-<netId>.c) with your solution to Learn by 10/24/2017

Note: these dates have varied from the dates in the original calendar, which will be updated soon.