UMBC

AN HONORS UNIVERSITY IN MARYLAND

CMSC Spring 2016

Homework 1 – Constructing Algorithms

Assignment: Homework 1 – Constructing Algorithms
Due Date: Monday, February 8th, 2015 by 8:59:59 PM
Value: 40 points

Homework 1 is an exercise in thinking algorithmically, and consists of writing out simple algorithms in English. This homework is similar to the "Explaining a Peanut Butter & Jelly Sandwich to an Alien" exercise we did in class.

Instructions

For each question below, write out a series of *clear*, *ordered* steps that *fully* solve the problem given. Make sure to number each step, and to keep individual steps simple.

For example, the following is what we are looking for:

1. Get a bag of bread and open it.
2. Reach into the bag and remove two slices of bread.

Yes!

Something like this, however, is not acceptable and will lose you points:

Grab a bag of bread and take some out, then open the jar of jelly and stick the spoon in the jelly, and then put the jelly on the bread, and then put the knife in the peanut butter and put peanut butter on the other piece…

NO!

You should also avoid breaking steps into excessively small pieces:

1. Get a bag of bread.
2. Untwist the twist tie on the bag of bread.
3. Open the bag of bread.
4. Reach into the bag of bread.
5. Grab two slices of bread.
6. Remove the two slices of bread you grabbed.

NO!

When you are writing your solutions, you can use loops (which we discussed in class during Lecture 2). For example, if we wanted to make five peanut butter and jelly sandwiches, we might use this:

```
1.    For each sandwich:
2.    Get a bag of bread and open it.
   …
13.   Put the two pieces of bread together, jelly side to peanut butter side.
14.   Repeat from step 2 until you've made 5 sandwiches in total.
```

Homework 1 should be completed on the GL server. Use emacs to create and complete a hw1.txt file, and put your name, the date, and your section number at the top of the file. This will be the file that you submit. If you have already completed Lab 0, you should make sure to place the hw1.txt file inside a **`HW1`** folder (which is itself inside the **`Homeworks`**, which is inside the **`201`** directory).

Finally, some of the questions may use words you aren't very familiar with. If this is the case, please look at the Explanations section directly after the list of questions.

Example

An example question with answer is presented in the box below; this should give you a better idea of what type of formatting and language we expect to see in your answers. Like the example answer, your answers should be clear and concise, and the order in which the steps are taken should also be plainly stated.

Example Question

You are given a positive number greater than zero. You need to calculate the sum of all of the natural (whole) numbers up to and including that one. What steps do you take?

Example Answer

1. Obtain the positive number from the user.
2. Initialize a variable called “answer” to 0.
3. Until the positive number changes to zero:
4. Add the positive number to the variable “answer”
5. Subtract 1 from the positive number.
6. Go back to step 3, and check if the number is now 0.
7. Present the final “answer” to the user.

Some explanation: (you don’t need to include this in your answers)

The algorithm we’ve written here simply counts down from the given number, and adds the value to an answer. For example, if we were given the number 5, we would repeats steps 4 through 6 five times:

5+0 = 5   4+5 = 9   3+9 = 12   2+12 = 14   1+12 = 15

We also could have solved this by adding from 1 up to the number, rather than subtracting until we reached 0. There are many ways to solve these sorts of problems – coming up with your own solution is part of the fun!

Questions

Each question is worth 9 points. Following the directions is worth 4 points.

Question 1

You are given a positive number, and need to decide whether it is a prime number or not. What steps do you take?
(Hint: You need to write an algorithm that will work for any positive number. If you're having trouble getting started, think about how you would test if 11 is a prime number. What changes when you need to test if 12 is a prime number?

Question 2

You are given a word, and need to decide if it is a palindrome or not. What steps do you take?
(Hint: How do you keep track of which two letters in the word you are comparing? How do you know when you've checked the entire word?)

Question 3

You have a stack of one thousand notecards, and each notecard has a number on it. You need to find the card with the lowest number on it. What steps do you take?
(Hint: "Look through the cards and find the smaller number." is not a solution. How do you determine which of two numbers is lower? How do you keep track of (i.e., "store") the smallest number you've found so far?)

Question 4

You are given a positive number and need to compute its factorial. What steps do you take?

IMPORTANT: Make sure that you write all of your answers out in English! Using lots of mathematical equations, code, or pseudo code in your answer will lose you points.

Explanations

Prime numbers

A prime number is a number that can only be evenly divided by itself and the number 1. For example, the number 5 is a prime number, because it cannot be easily divided by 2, 3, 4, or any numbers other than 1 and 5.
The number 12 is *not* a prime number, because 2 x 6 = 12, and also 3 x 4 = 12.
The number 13 is a prime number, because it cannot be divided by 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, or 12. It can only be divided by 1 and itself.

Palindromes

A palindrome is a word that is the same forward as it is backwards. For example, "Anna" is a palindrome. (The first and last letters are 'a' and the second and second-to-last letters are 'n'.) Other examples include words like "kayak" and "redder".
However, the word "moon" is *not* a palindrome. (The first letter of the word is 'm' while the last letter is 'n'.) The words "racecars" and "computer" are also not palindromes.

Factorials

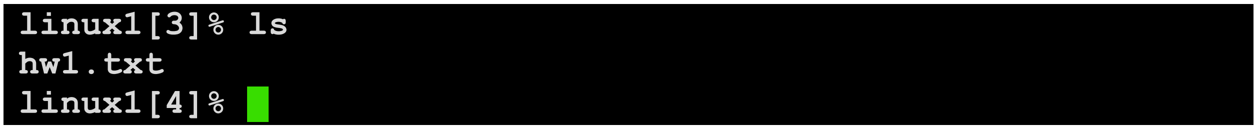
A factorial is the product of a whole number and all the numbers below it. In other words, if we have a whole number *n*, we need to multiply all the numbers from 1 to *n* together. For example, to calculate the factorial of the number 8 we multiply 1 x 2 x 3 x 4 x 5 x 6 x 7 x 8 to get 40,320 as our answer.
NOTE: The factorial of 0 is always 1, and a factorial cannot be computed for negative numbers.

Submitting

NOTE: How to submit is covered in Lab 0. If you have not completed Lab 0, you should do so before completing this part of the homework.

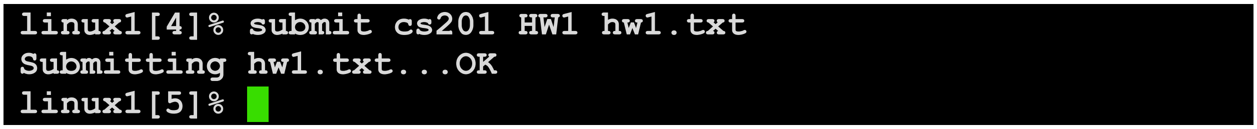
Once your hw1.txt file is complete, it is time to turn it in with the **`submit`** command.

You must be logged into your GL account, and you must be in the same directory as the hw1.txt file. To double check this, you can type **`ls`**.



```
linux1[3]% ls
hw1.txt
linux1[4]%
```

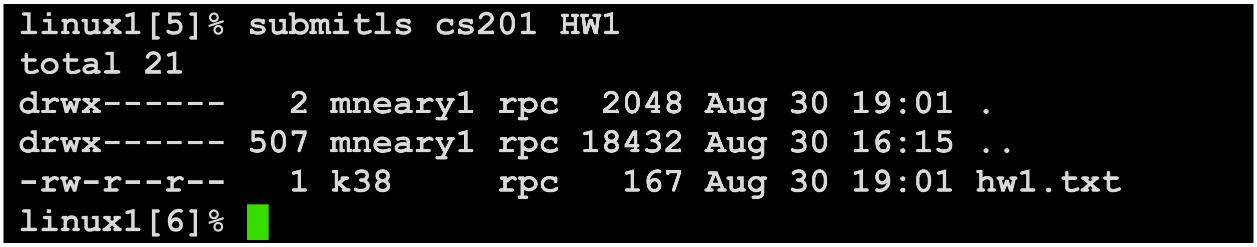
To submit your hw1.txt file, we use the **`submit`** command, where the class is **`cs201`**, and the assignment is **`HW1`**. Type in **`submit cs201 HW1 hw1.txt`** and press enter.



```
linux1[4]% submit cs201 HW1 hw1.txt
Submitting hw1.txt...OK
linux1[5]%
```

If you don't get a confirmation like the one above, check that you have not made any typos or errors in the command.

You can double-check that your homework was submitted by using the **`submitls`** command. Type in **`submitls cs201 HW1`** and hit enter. (You'll see your own username, of course.)



```
linux1[5]% submitls cs201 HW1
total 21
drwx------   2 mneary1 rpc  2048 Aug 30 19:01 .
drwx------ 507 mneary1 rpc 18432 Aug 30 16:15 ..
-rw-r--r--   1 k38     rpc   167 Aug 30 19:01 hw1.txt
linux1[6]%
```

And you're done!