COSC 1336 Lab: 5

Relevant reading: Sections 5.1-5.3

Due: Oct. 16, 2:30 pm

(Late date: Oct. 23, 2:30 pm)

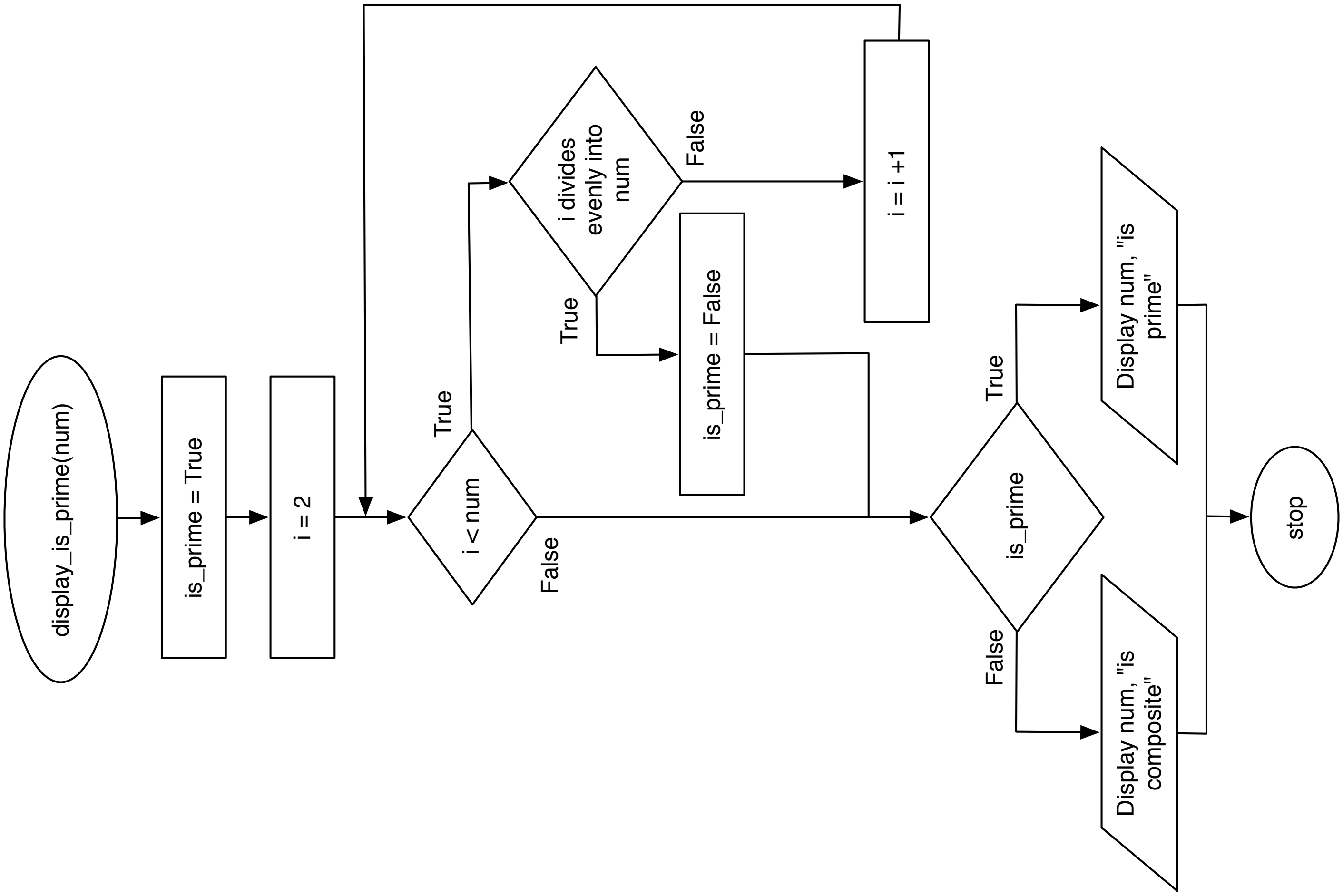
50 Points

Problem 1. [**8 points**] Download the file `tip_table_loop.py`. In it, you will find a version of the tip table program you wrote for lab 2 (though it displays the amount due for a few more tip rate values). For this problem, change the `main` function so that rather than calling the function `display_total_due` function multiple times, it uses a loop. You may choose whether you want to use a `while` loop or a `for` loop. I recommend that you choose the one that seems more challenging.

Problem 2. For this problem, you will create a program that allows the user to find out whether integers are prime or composite. A *prime* number is one that is evenly divisible only by 1 and itself (and a *composite* number is one that is not prime).

a. [**12 points**] Create a file called `is_prime.py`. In it, write a function called `display_is_prime` that uses a `for` loop and the `range` function to implement the flowchart shown below (shown sideways so it fits on the paper better). The purpose of the function is to determine whether or not the input integer is prime, and display an appropriate message. Test your function on several inputs to be sure it works correctly.

The next subproblem will use this function, so it is important that you get it working. Get help in lab, via email, at office hours, or from the tutor if you aren't able to get it working on your own.



b. **[8 points]** Now write a `main` function that uses a `while` loop and the function you wrote in part a to allow the user to enter integers, and tells the user whether the entered integer is prime or composite.

Problem 3. **[22 points]** For this problem, you will create a visual demonstration of the power of compound interest.

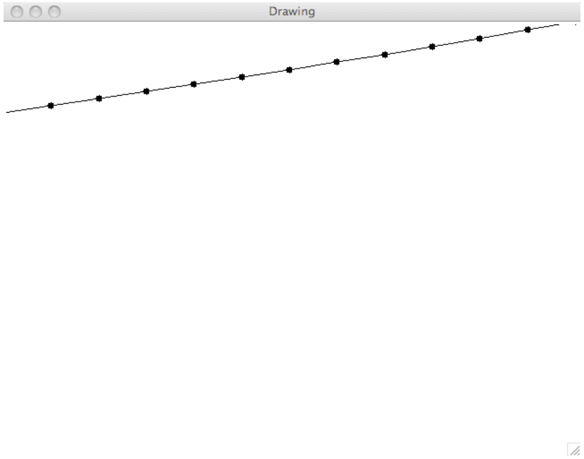
a. **[12 points]** First, in a file called `interest_loop.py`, write a function called `balance_display_loop` that receives a starting bank account balance, a monthly interest rate, and a number of months over which to accrue interest. This function should use a loop to display the bank account balance at the end of each month. Then write a `main` function that reads in the balance, interest rate, and number of months from the user, and calls the function.

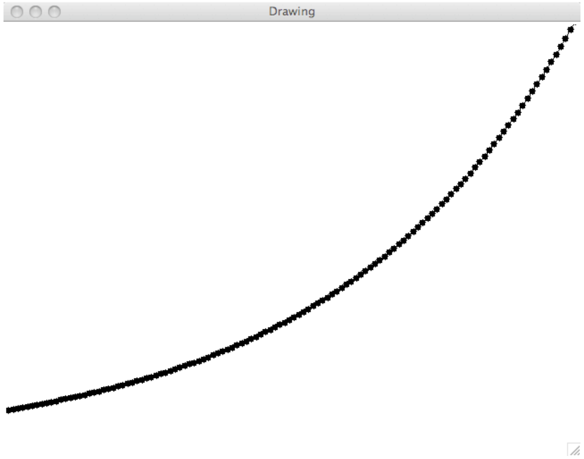
Here is a transcript of what running the program might look like:

```
Enter the starting account balance: $100
Enter the monthly interest rate (where 1.0 is 100%): 0.02
Enter the number of months: 3
Starting balance: $100.00
Balance after 1 months: $102.00
Balance after 2 months: $104.04
Balance after 3 months: $106.12
```

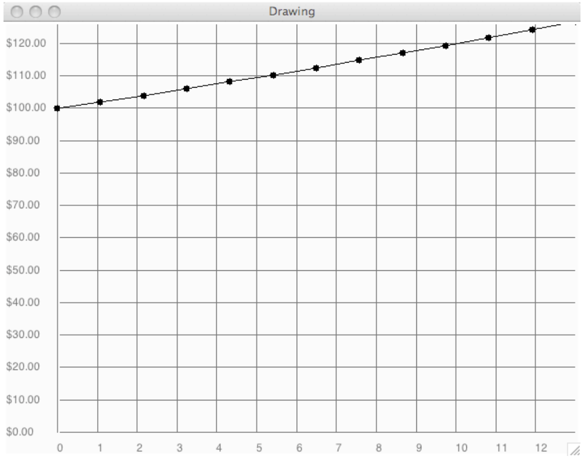
b. **[10 points]** Now create a graphical representation of the same data you calculated in the last problem. Download the file `interest_loop_graphics.py`, as well as the `draw_functions2.py` file, and put the two files in the same folder. The `draw_functions2.py` file has a new (and hopefully more robust) implementation of the same graphics functions you used in lab 2, with the addition of one more function: `draw_text`. You can find a description of all of the functions, including the new one, in the `graphics_functions.pdf` document that is posted in the "Resources" section of Blackboard.

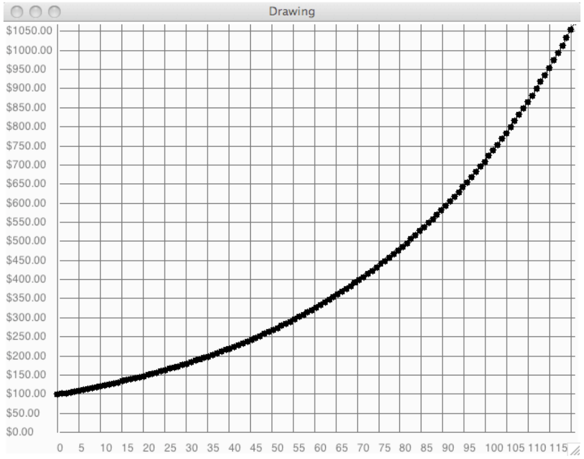
Modify the provided `balance_display_loop` (which receives the same arguments as in part a), so that instead of displaying the values on the screen, it creates a plot of the data. The simplest thing that will receive full credit should show a circle for each data point and connect each point with a line, as shown on the below. Both plots show an account that started at $100 with 2% interest. On the left is the plot for 12 months and on the right, 10 years (120 months).





For up to five points extra credit, you can create a fancier plot that includes labels and/or grid lines, as shown below. The hardest part of this is getting things to rescale as the numbers get bigger so that you can always read the labels.





For this lab, you should submit the following files:

- `tip_table_loop.py`
- `is_prime.py`
- `interest_loop.py` and `interest_loop_graphics.py`