COP4521 Programming Assignment 1: Prime Triplet

Objectives:

- Practice problem solving using Python
- Experience with the performance of a Python program

Description:

A prime triplet is a set of three prime numbers where the smallest and the largest differ by 6. For example, (5, 7, 11) and (13, 17, 19) are prime triplets. In number theory, there is a prime triplet conjecture, which states that there are infinitely many prime triplets. In this assignment, you will write a Python program that prompts the user to input an integer number (positive or negative), and then display the smallest prime triplet whose smallest prime is no less than the number entered. Note that prime numbers are positive numbers (the smallest prime is 2).

Due time: September 4, 2024, 11:59pm.

Submission: Name your program lastname_firstinitial_primetriplet.py and submit on Canvas.

Grading (50 points total):

- Include basic header (template at the course website) for assignment and name your program lastname_firstinitial_primetriplet.py (10 points)
- Compute and output the prime triplet correctly in basic tests (25 points)
- Compute and output the prime triplet correctly in extreme tests (15 points)
- Programs with a runtime error for legitimate input (including 1000,000,000,000) will get no more than 5 points.
- -15 points for a program running for more than 30 seconds on linprog after input 1000000000000.
- +5 **extra** points for a program that gives the answer for 1000000000000 in less than 12 seconds on linprog (using the same logic to give answers for all other numbers). To get the extra points, you need to report the execution time of your program in the basic header. You can change the input statement (n=int(input("Enter a number: "))) to an assignment (n=1000000000000), measure the program execution time, and report the result.

Notes:

- Basic tests will use numbers no larger than 200 to check the correctness of the code. The following are some examples:

  <linprog2:945> python3 primetriplet1.py
  Enter a number: 5
  The smallest triplet larger than 5 is (5, 7, 11)
  <linprog2:946> python3 primetriplet1.py
  Enter a number: 10
  The smallest triplet larger than 10 is (11, 13, 17)

<linprog2:947> python3 primetriplet1.py
Enter a number: 20
The smallest triplet larger than 20 is (37, 41, 43)
<linprog2:948> python3 primetriplet1.py
Enter a number: 100
The smallest triplet larger than 100 is (101, 103, 107)
<linprog2:949> python3 primetriplet1.py
Enter a number: -1
The smallest triplet larger than -1 is (5, 7, 11)
<linprog2:950>

- Extreme tests will input some large numbers such as 1000000000000 (the numbers may be even larger than that).
- You can use the *time* command to measure the execution time of a program. In the following, time2.py is the same file as primetriplet1.py with the input statement changed to an assignment:

  <linprog3:533> time python3 time2.py
  The smallest triplet larger than 1000000000000 is (1000000005073, 1000000005077, 1000000005079)
  4.696u 0.009s 0:04.72 99.3% 0+0k 8+0io 0pf+0w
  <linprog3:534>

  You can copy and paste the timing result to the header of your program to claim the extra points.
- If you need to use the square root function, you can import the math module (*import math*), then *math.sqrt(n)* returns the square root of n.