

# Homework 5

Colt Bradley

## 1 Introduction

The assignment was to create a program that prompted for a number, then determined all the prime numbers less than the given number and printed a statement.

## 2 Logic

My code prompts the user for an integer, then stores it as an integer. We then handle the case of 1, 2, and 3 separately using if statements to make it a bit easier. The bulk of the code uses two nested while loops, the first running through every integer between 4 (since we already checked 1-3) and the inputted integer. The second while loop checks if that number is divisible by any number smaller than it. If it is, a marker is added to a testing list. At the end of the loop, if that list is empty the number is added to a list of primes (which already contains 2 and 3). If the list isn't empty, it's cleared and we move on to the next number.

At the end, we have a list of primes called "primes." To see if our integer was prime, we check if it equals the last value of this list. If it does, we print "Congratulations! The number you chose was prime!" If it doesn't, we print "Sorry, the number you chose was not prime." Either way, we print our list of primes.

## 3 Python Code

```
#Colt Bradley
#1.26.16
#Lesson 5 Homework

#import essential modules
```

```

import numpy as n

#ask user for an integer
integer = raw_input("Give me any integer ")
integer = int(integer)

#sets up the list to hold prime numbers
primes = [2,3]

#first, we take care of special cases. 1, 2, and 3 are cases that if we take care
#of now it will be easier later
if integer == 1:
    print "Sorry, the number you chose was not prime."
elif integer == 2:
    print "Congratulations! The number you chose was prime!"
elif integer == 3:
    print primes
    print "Congratulations! The number you chose was prime!"

#here's the bluk of the code. Basically, the first while loop runs through every
#number up to and including whatever inputted number. The second while loop checks
#what numbers divide that number and append it to a list. Remember that empty
#lists are false, so if the list is false then we append our test number to our
#prime list and iterate. If true, we just interate. at the end, we check if the
#final value of the list is equal to the inputted value, and if so we declare
#that it is prime and print our list of primes.
else:
    r = 4
    while r <= integer+1:
        m = 2
        test = []
        while m < r:
            if (r%m) == 0:
                test.append(m)
                m = m+1
            else:
                m = m+1
        if not test:
            primes.append(r)
            r = r+1

```

```
        else: r = r+1
print primes
if primes[-1] == integer:
    print "Congratulations! The number you chose was prime!"
else:
    print "Sorry, the number you chose was not prime."
```