CS410P

ASSIGNMENT 2P

The purpose of this assignment is to give you practice with repetition using while, conditional statements, using lists and handling exceptions.

You can only use the "while" loop for repetition in your code even if you are familiar with other forms of repetition. You are also not allowed to use the "break" and "continue" statements in any programs.

You will need to write 2 separate programs as outlined below:

1) Program 1 to be submitted as **collatz.py**

   Consider this rule for transforming a positive integer **n**:

   if **n** is even, divide it by 2
   if not, multiply it by 3 and add 1

   The *Collatz conjecture* says that, no matter what integer you start with, applying this transformation over and over will eventually reach the value 1.

   Write a program **collatz.py** that prompts the user for a starting value **n**, applies the above transformation repeatedly, until **n** is 1.

   The program should simply *count the number of transformations* it took to reach 1 and print that count and the transformations. The program should stop after one positive number is computed for generating its Collatz conjecture and stop. If the input value is invalid, the user must be prompted repeatedly until one valid value is supplied.

   Requirement for Mimir: You must print every number computed in the transformation along with the total number of transformations (you can separate them by spaces or any delimiter as indicated by my sample output below).
   Examples:

   Sample output 1:

```
Enter a positive number: 11
11->34->17->52->26->13->40->20->10->5->16->8->4->2->1
14 transformations
```

Sample output 2:

```
Enter a positive number: 0
Please enter an integer value greater than zero

Enter a positive number: -10
Please enter an integer value greater than zero

Enter a positive number: abc
Please enter an integer value greater than zero

Enter a positive number: 18
18->9->28->14->7->22->11->34->17->52->26->13->40->20->10->5->16->8->4->2->1
20 transformations
```

You must use an exception handler to catch all invalid input that are not integers. In addition you must validate your input to make sure it's a positive number.

2) Program 2 to be submitted as **fibPrime.py**

This program will continuously prompt the user for a number (integers only) between 1 and 1000 and do the following for each number input by the user:

a) If the number is less than 1 or greater than 1000 you must display a message indicating that the input is invalid and continue to prompt the user
b) If the number entered is 1000, you should stop the program.
c) If the input is not an integer catch the exception and prompt the user again.
d) In all other cases do the following:
   - Generate the Fibonacci series of numbers until and including the number entered and store the series in a List (let's call this the *Fibonacci List*). Print the items in this list making sure that only a maximum of 10 numbers are printed on one line, followed by the total number of items.
     The Fibonacci Sequence is the series of numbers: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ... The next number is found by adding up the two numbers before it.
   - Generate the list of Prime Numbers from 1 to the number entered and store them in a List (let's call this the *PrimeNumber List*). Print the items in this list making sure that only a maximum of 10 numbers are printed on one line, followed by the total number of items.
     A prime number is a number greater than 1 and has no divisors other than 1 and the number itself
   - Create a third list where each number in the *Fibonacci List* is found in the *Prime Number List.* Print the items in this list, followed by the total number of items.

Requirement for Mimir: You must print the total counts as shown in the sample outputs (as an example below fib_count, prime_count, combined_count are 3 variables holding the count values):

```
print("Total Fibonacci:", fib_count)
print("Total Prime:", prime_count)
print("Total Combined:", combined_count)
```

Sample Output:

```
Enter a number between 1 and 1000: 500
You entered:  500
The Fibonacci Sequence less than  500  is:
    0     1     1     2     3     5     8    13    21    34
   55    89   144   233   377

Total Fibonacci: 15

The Prime Number List less than  500  is:
    2     3     5     7    11    13    17    19    23    29
   31    37    41    43    47    53    59    61    67    71
   73    79    83    89    97   101   103   107   109   113
  127   131   137   139   149   151   157   163   167   173
  179   181   191   193   197   199   211   223   227   229
  233   239   241   251   257   263   269   271   277   281
  283   293   307   311   313   317   331   337   347   349
  353   359   367   373   379   383   389   397   401   409
  419   421   431   433   439   443   449   457   461   463
  467   479   487   491   499

Total Prime: 95

The values in common between the Prime Numbers and Fibonacci Sequence
less than  500  are:
    2     3     5    13    89   233

Total Combined: 6


Enter a number between 1 and 1000: 1002
You entered:  1002
Invalid input.

Enter a number between 1 and 1000: -6
You entered:  -6
Invalid input.

Enter a number between 1 and 1000: 1000
You entered:  1000
```

Grade Key:

A	Collatz transformation accuracy	15
B	Collatz transformation Output Format	5
C	Collatz transformation exception handler & input validation	5
D	Fibonacci results accuracy	15
E	Fibonacci Output Format	5
F	Prime Number results accuracy	20
G	Prime Number Output Format	5
H	Creation of Lists to save numbers	5
I	Repetition using **while** only (no for loops), no break or continue statements used	15
J	Common List accuracy, Output Format	5
K	Fib/Prime exception handler and input validation	10
L	Fib/Prime number program terminates correctly only when 1000 is entered	5