

Problem Set 5

Problem - 5.1

Write a Python program that reads a positive integer **n** and prints the prime factors of **n**.

Input: 24

Output: 2, 3

Input: 150

Output: 2, 3, 5

Problem - 5.2

Write a Python program that reads a positive integer $n > 10$ and prints all the primes whose decimal representations when reversed are also primes.

Input: 50

Output: 11 13 17 31 37

Continued Fraction

A rational number $\frac{p}{q}$ can be represented as a simple continued fraction (CF).

$$\frac{45}{37} = 1 + \frac{1}{4 + \frac{1}{1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{2}}}}}$$

A simple CF can be stored as a list,
 $[1, 4, 1, 1, 1, 2]$.

Problem - 5.3

Write a Python program that reads a **continued fraction** as a list and computes the **numerator** and the **denominator**.

Input: [1,4,1,1,1,2]

Output: 45/37

Problem - 5.4

Write a **recursive function** `listFact(n, i)` that takes a non-negative integer `n` as the first parameter and `1` as the second parameter. It returns the list of all factors of `n`.

Input: 13

Output: [1, 13]

Input: 36

Output: [1, 2, 3, 4, 6, 9, 12, 18, 36]

Input: 24

Output: [1, 2, 3, 4, 6, 8, 12, 24]

Problem - 5.5

Write a **recursive function** `myMap(l, fun)` that takes a list of integers `l` and applies the function `fun` to each element of `l`, and returns a list of values.

As an example, if `fun` is the factorial function, `fact(n)` and the list is `[2, 5, 4]`, then the return value of `myMap()` is `[2, 120, 24]`. **Do not use `map()` function.**

Bubble Sort Algorithm

```
for i ← 0 to n - 2 do
  exchange = False
  for j ← n - 1 downto i + 1 do
    if (data[j-1] > data[j])
      data[j-1] ↔ data[j] # Exchange
      exchange = True
  endFor
  if (exchange == False) break
endFor
```


Problem - 5.6

Write a Python program to implement the bubble sort algorithm.