Computer Engineering Dept.
Cairo University
Spring 2021

Operating Systems
Advanced Bash Scripting
Part 2 Requirements

Cairo University

Requirement 1: Find Primes

You're required to write a function **FindPrime** that takes two parameters as follows:

1. The first **parameter** is a positive integer **N** representing the size of the prime number to be extracted.
2. The second **parameter** is a numeric string **X** which will be searched. The digits of X take only values from [1-9].

The function should find the number of all prime numbers of size **N consecutive digits** contained within X. It should also return the largest value of these prime numbers.

If no prime numbers of size N exist within X, the function should echo 0 for both entries.

In case one parameter only is specified for the function, then N should search for N = 3 by default.

Notes:

- You do not need to perform any checks concerning the validity of the input. All the test cases are valid integers for X and valid sizes for N.

Example for running the script (Assume the script is called "1.sh")

```
$ source 1.sh
$ FindPrime 2 2329971
4 97
$ FindPrime 2 222324
1 23
$ FindPrime 68271982
3 827
$ FindPrime 3 2368224
0 0
```

Explanation:

- The first input string "2329971" can be decomposed into the following numbers consisting of two digits "23", "32", "29", "99", "97", and "71" when we slide over the string from the beginning to the end with a sliding window of size = 2.
- The extracted numbers include only four prime numbers "23", "29", "97", and "71".
- Therefore, the first entry in the output is 4 (indicating the number of prime numbers) and the second entry is 97 (indicating the largest number of these primes).

Requirement 2: Check Pattern

You're required to write a function **CheckPattern** that takes a string as an input and prints 1 if the string consists only of a repeated pattern (one substring occurring **more than** once) and 0 otherwise.

The function will always take only one parameter representing the string to be tested.

Your code should handle mixed lower and capital letters, you should solve this using string processing, however there's a faster way you already know from the last tutorial as you can convert them all uppercase or alternatively lower case.

Notes:

- You do not need to perform any checks concerning the validity of the input. All the test cases are valid strings.

Example for running the script (Assume the script is called "2.sh")

```
$ source 2.sh
$ CheckPattern boyboyboyboy
1
$ CheckPattern tomatotomato
1
$ CheckPattern tomtomtomato
0
$ CheckPattern ThisPatternThisPatternthispattern
1
$ CheckPattern aaaaaaaaaaaa
1
$ CheckPattern aaaaaabbbbbb
0
$ CheckPattern THISpatternTHATpatternTHISpattern
0
```

Hints:

- The source command (the first command in the two snippet examples) can be used to load any functions into the current terminal so you can use the function directly as shown.
- If you performed any changes in the script, you need to re-run the source command again.
- Don't forget to put #! bin/bash at the beginning of your scripts
- Don't forget to make your script executable