COS 350 Systems Programming
2024 Spring
Assignment 4

Q1: Bash programming practice. You can paste your code and a screenshot of your execution result in. (10 points, 5 points for code, 5 points for screenshot)
Parse through an Apache access log file (download 'access.log' from BrightSpace), count the number of requests per HTTP status code, and output a summary. This can be useful for web server administrators to quickly assess the distribution of response codes their server is producing, which in turn can help identify issues like frequent 404 (Not Found) or 500 (Internal Server Error) responses.

A typical log entry looks something like this:

```
127.0.0.1 - - [24/Mar/2021:13:15:42 -0400] "GET /index.html HTTP/1.1" 200 612 "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)"
```

In this entry, '200' is the HTTP status code, which indicates a successful response.

To run the script for parsing Apache access logs, you must provide one argument: the filename of the log file you wish to analyze. This means the script should be executed in the following manner:

```
./Script.sh log_file_name.log
```

Upon successful execution, the script will produce output summarizing the count of HTTP status codes found in the log file, formatted as follows:

```
Status Code 200: 2 times
Status Code 404: 1 times
Status Code 500: 1 times
……
```

Q2: Regular Expressions exercises. (40 points, 10 points each.)

A) Validate usernames starting with a letter and consisting of letters, digits, or underscores, 5 to 12 characters in length.

B) Match hexadecimal numbers that start with 0x or 0X. After the 0x or 0X prefix, this regex should match one or more hexadecimal characters (digits and letters A-F, case-insensitive).

C) Match passwords that are at least 8 characters long and contain a mix of uppercase and lowercase letters, numbers, and special characters ('!@#$%^&*()').

D) Match file names that do not end in specific extensions (e.g., excluding .exe and .tmp files).

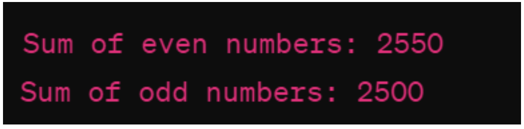
Q3: Testing Bash scripts. You can paste your script and a screenshot of your execution result. (30 points)
Download the script named 'COS350_A4_Q4.sh' from BrightSpace. This script includes two important functions: "deleteFiles" and "appendToFile".

A) Use either shunit2 or Bats to create a script specifically for testing the two functions mentioned above. Make sure your test script checks if these functions are working correctly. In your test script, include 'setup' and 'teardown' functions. Use 'setup' to set things up before your tests run (like creating files or folders needed for the test). Use 'teardown' to clean everything up after the tests finish. (20 points, 15 points for the testing script, 5 points for the screenshot.)

B) Utilize the ShellCheck to review this script. (10 points, just provide the screenshot.)

Q4: C programming language practice. You can paste your code and a screenshot of your execution result. (20 points, 10 points each.)

A) Write a C program that calculates and prints the sum of all even numbers and the sum of all odd numbers from 1 to 100. Your output should be similar to the following figure. (5 points for the code, 5 points for the screenshot.)



```
Sum of even numbers: 2550
Sum of odd numbers: 2500
```

B) Implement a C program that calculates the Nth Fibonacci number using recursion. The Fibonacci sequence starts with two numbers 1 and 1, and each subsequent number is the sum of the previous two. The program should prompt the user to enter a non-negative integer N and then print the Nth Fibonacci number. (5 points for the code, 5 points for the screenshot.)