Free and Open Source Lab

Final Examination Report

Submitted By:

Mohammed Rabeeh
Roll No: 35
TVE18CS036
24 July 2020

Contents

1 Program 1

1.1 Problem Statement

If we list all the natural numbers below 10 that are multiples of 3 or 5, we get 3, 5, 6 and 9. The sum of these multiples is 23. Write a shell script to find the sum of all the multiples of 3 or 5 below an integer n (10 ¡= n ¡= 10000).

1.2 Theory

We have to the sum of all the numbers which are multiples of 3 or 5. A number is divisible by x if it gives us the remainder 0 when divided x. So to find the sum, we check if a number is divisible by 3 or 5. If it is, we add the number to our sum.

1.3 Implementation

The idea is to loop through all the numbers until n and take the total sum of all the number divisible by 3 or 5.

1. Read number n from the command line.
2. Initialise the sum to 0.
3. Check all number from 0 upto n (n not included).
   (a) For every number, check if the number is divisible by 3 or 5 using modulo operator.
   (b) If it is, add the number to the sum.
4. Print the sum.

1.4 Instructions for use

The source code can be found here. Click Me. The program file is named as MohammedRabeeh_TVE18CS036_program1.sh. To execute the file program, run the following commands

```
chmod +x MohammedRabeeh_TVE18CS036_program1.sh
./MohammedRabeeh_TVE18CS036_program1.sh n
```

where n, is the upper limit of the input. The output will be shown in the next line which is the sum of all multiples of 3 and 5.

1.5 Source Code

```
#!/bin/bash
# Program Name: MohammedRabeeh_TVE18CS036_1.sh
# Author: Mohammed Rabeeh
# Date: 24th July 2020
# OS: macOS Catalina 10.15.3
# Shell: zsh
# Function: To find the sum of all the multiples of 3 or 5 below an integer n
# (10 <= n <= 10000).
# Input: ./MohammedRabeeh_TVE18CS036_1.sh n

# Check if arguments are exists. If not, exit.
if [ -z "$1" ]; then
  echo "USAGE: $0 n"
  exit
fi

# Initialise sum to 0
sum=0
# Loop through all the numbers
for ((i = 0 ; i < $1 ; i++)); do
  # Check if the number is divisible by 3 or 5
  if [ $((i%3)) -eq 0 -o $((i%5)) -eq 0 ]; then
    # Add sum to total
    sum=$((sum + i))
  fi
done
# Print sum
echo $sum
```

1.6 Testing

The above program was tested with edge cases kept in mind. The cases that were tested

- No input

```
./MohammedRabeeh_TVE18CS036_program1.sh
USAGE: ./MohammedRabeeh_TVE18CS036_program1.sh n
```

- Proper Input

```
./MohammedRabeeh_TVE18CS036_program1.sh 10
23
```

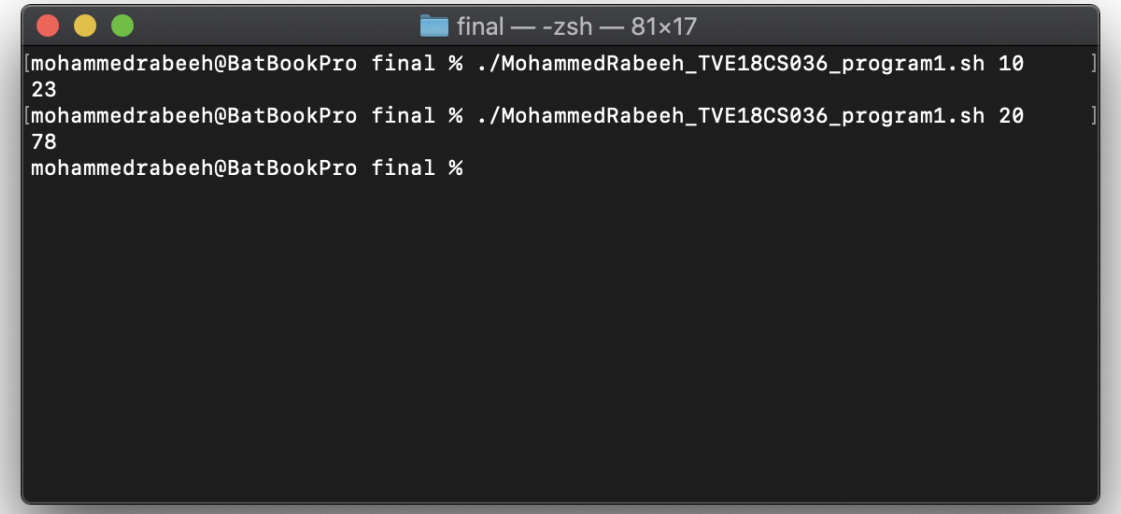
- Large Input

```
./MohammedRabeeh_TVE18CS036_program1.sh 100000
2333316668
```

1.7 Output (Text)

```
./MohammedRabeeh_TVE18CS036_program1.sh 10
23
./MohammedRabeeh_TVE18CS036_program1.sh 20
78
```

1.8 Output (Image)



2 Program 2

2.1 Problem Statement

Write a script to print name of the file that contains lines longer then n chars.

2.2 Theory

Given a list of files, we have to print the names of the files that has lines with no. of characters greater than n.

2.3 Implementation

We can use then grep utility to check the number of characters in a line. grep is a command-line utility for searching plain-text data sets for lines that match a regular expression. We can pass the condition of selecting lines having least n + 1 characters. We then count the number of lines which has been outputted from grep to determine if that file has lines with more than n characters.

2.4 Source Code

```
#!/bin/bash
# Program Name: MohammedRabeeh_TVE18CS036_program2.sh
# Author: Mohammed Rabeeh
# Date: 24th July 2020
# OS: macOS Catalina 10.15.3
# Shell: zsh
# Function: To print name of the file that contains lines longer then n
# chars.
# Input: ./MohammedRabeeh_TVE18CS036_program2.sh n

# Check for valid n
if [ -z "$1" ]; then
  echo "USAGE: $0 n"
  exit
fi
```

```
# Loop through all the files in the directory with txt extension
for filename in *.txt; do
    [ -e "$filename" ] || continue
    # grepping for lines with greater than n characters and
    # counting the lines
    c=‘grep ".\{$(($1 + 1))\}" $filename | wc -l‘
    # If such lines exists, then print the filename.
    if [ $c -ne 0 ]; then
      echo $filename
    fi
done
```

2.5 Instructions for use

The source code can be found here. Click Me. The program file is named as MohammedRabeeh_TVE18CS036_program2.sh. To execute the file program, run the following commands

```
chmod +x MohammedRabeeh_TVE18CS036_program1.sh
./MohammedRabeeh_TVE18CS036_program1.sh n
```

where n, is the required no. of characters in the line. The program will go through all the files with .txt extention in the current directory and print the file names according to the given n.

2.6 Input Files

fruit1.txt

```
Apples and Oranges
Bananas
```

fruit2.txt

```
Grapes
Kiwi
Tomato
```

2.7 Testing

The above program was tested with edge cases kept in mind. The cases that were tested

- No input

  ```
  ./MohammedRabeeh_TVE18CS036_program2.sh
  USAGE: ./MohammedRabeeh_TVE18CS036_program1.sh n
  ```

- Proper Input

  ```
  ./MohammedRabeeh_TVE18CS036_program2.sh 10
  fruit1.txt
  ```

- Proper Input 2

  ```
  ./MohammedRabeeh_TVE18CS036_program2.sh 5
  fruit1.txt
  fruit2.txt
  ```

- Large n (No files returned)

  ```
  ./MohammedRabeeh_TVE18CS036_program2.sh 100000
  ```

2.8 Output (Text)

```
./MohammedRabeeh_TVE18CS036_program2.sh 10
fruit1.txt
./MohammedRabeeh_TVE18CS036_program2.sh 5
fruit1.txt
fruit2.txt
./MohammedRabeeh_TVE18CS036_program2.sh 20
```

2.9 Output (Image)