

# Assignment 12

-Snehal Deshmukh

1. Create a user case where an array that stores the first ten **prime numbers**. Iterate over the array and print out each element inside it

```
#!/bin/bash

# Initialize an empty array to store prime numbers
primes=()

# Loop through the first 10 positive integers
for num in {1..10}
do
    # Check if the current number is prime
    if [ $(factor $num | wc -w) -eq 2 ]
    then
        # If the number is prime, add it to the array
        primes+=($num)
    fi
done

# Loop through the array of prime numbers and print each element
for prime in "${primes[@]}"
do
    echo $prime
done

~
~
~
```

Here's an example Bash script that creates an array of the first ten prime numbers and iterates over it to print out each element:

```
demo@ubuntu:/home/snehal/loop$ ./for_loop.sh  
2  
3  
5  
7  
demo@ubuntu:/home/snehal/loop$
```

In this script, we're using a for loop to iterate through the first 10 positive integers (represented by the numbers 1 to 10).

2. Create a user case to print all the files and directory that exists under the `/etc` directory

```
demo@ubuntu:~/home/snehal/loop$ find /etc -print
/etc
/etc/debconf.conf
/etc/bluetooth
/etc/bluetooth/main.conf
/etc/bluetooth/input.conf
/etc/bluetooth/network.conf
/etc/python2.7
/etc/python2.7/sitecustomize.py
/etc/rc.local
/etc/shadow
/etc/kernel
/etc/kernel/postrm.d
/etc/kernel/postrm.d/initramfs-tools
/etc/kernel/postrm.d/zz-update-grub
/etc/kernel/preinst.d
/etc/kernel/preinst.d/intel-microcode
/etc/kernel/postinst.d
/etc/kernel/postinst.d/update-notifier
/etc/kernel/postinst.d/initramfs-tools
/etc/kernel/postinst.d/unattended-upgrades
/etc/kernel/postinst.d/xx-update-initrd-links
/etc/kernel/postinst.d/apt-auto-removal
/etc/kernel/postinst.d/zz-update-grub
/etc/kernel/install.d
/etc/hp
/etc/hp/hplip.conf
/etc/sysctl.d
/etc/sysctl.d/10-console-messages.conf
/etc/sysctl.d/10-zero-page.conf
/etc/sysctl.d/10-kernel-hardening.conf
/etc/sysctl.d/README
/etc/sysctl.d/99-sysctl.conf
/etc/sysctl.d/10-ptrace.conf
```

3. Create a user case to print all **7 days** of a week by passing values as arguments and iterating for loop over these values

```
File Edit View Search Terminal Help
#!/bin/bash

# Define an array of days of the week
days=("Sunday" "Monday" "Tuesday" "Wednesday" "Thursday" "Friday" "Saturday")

# Loop through the command line arguments and print out the corresponding days
for day_num in "$@"
do
    # Subtract 1 from the day number to convert to a zero-indexed array index
    index=$((day_num - 1))

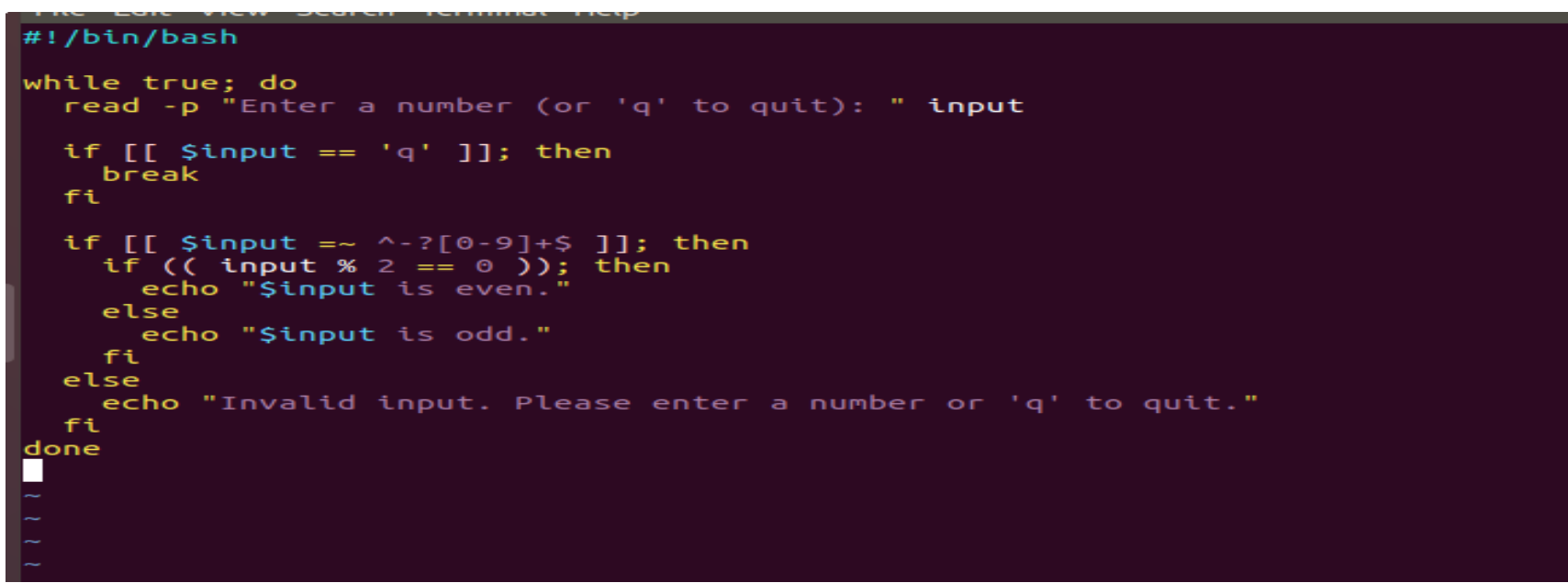
    # Check that the index is within the range of the array
    if [ $index -ge 0 ] && [ $index -lt ${#days[@]} ]
    then
        # If the index is valid, print out the corresponding day
        echo ${days[$index]}
    else
        # If the index is out of range, print an error message
        echo "Invalid day number: $day_num"
    fi
done
~
~
~
~
~
```

For each argument, we're subtracting 1 from the day number to convert it to a zero-indexed array index. We're then checking that the resulting index is within the range of the days array using an if statement. If the index is valid, we print out the corresponding day using the echo command. If the index is out of range, we print an error message.

```
demo@ubuntu:/home/snehal/loop$ ./days.sh 1 2 3 4 5 6 7
Sunday
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
demo@ubuntu:/home/snehal/loop$ ./days.sh 1
Sunday
demo@ubuntu:/home/snehal/loop$ ./days.sh 2
Monday
demo@ubuntu:/home/snehal/loop$ ./days.sh 8
Invalid day number: 8
demo@ubuntu:/home/snehal/loop$
```

This will print out the names of all 7 days of the week. If you pass in any invalid day numbers (i.e. numbers outside the range of 1 to 7), the script will print an error message instead of the corresponding day.

4. Creating a numbered menu to allow a user to select a number. Once a number is selected by the user display whether the number is even or odd.

A terminal window with a dark purple background and light green text. The window title is "File Edit View Search Terminal Help". The script content is as follows:

```
#!/bin/bash

while true; do
    read -p "Enter a number (or 'q' to quit): " input

    if [[ $input == 'q' ]]; then
        break
    fi

    if [[ $input =~ ^-?[0-9]+$ ]]; then
        if (( input % 2 == 0 )); then
            echo "$input is even."
        else
            echo "$input is odd."
        fi
    else
        echo "Invalid input. Please enter a number or 'q' to quit."
    fi
done
```

The cursor is at the end of the last line of the script.

In this script, the while loop prompts the user for input using the read command. If the user enters "q", the script breaks out of the loop and exits.

```
demo@ubuntu:/home/snehal/loop$ ./even_odd.sh
Enter a number (or 'q' to quit): 2
2 is even.
Enter a number (or 'q' to quit): 7
7 is odd.
Enter a number (or 'q' to quit): q
demo@ubuntu:/home/snehal/loop$
```

The script checks whether the input is a valid integer using a regular expression check: `[[ $input =~ ^-?[0-9]+$ ]]`. If the input is a valid integer, the script checks whether it's even or odd using the modulus operator `%`. If the input is not a valid integer, the script prints an error message and continues the loop.



5. Create a user case of **simple calculator** that prompt the user for input and performs basic arithmetic operations like **addition, subtraction, multiplication, and division.**

```
echo "Welcome to the calculator script!"

while true; do
    read -p "Please enter the first number: " num1
    read -p "Please enter the operator (+, -, *, /): " operator
    read -p "Please enter the second number: " num2

    case $operator in
        +)
            result=$((num1 + num2))
            ;;
        -)
            result=$((num1 - num2))
            ;;
        *)
            if [[ $operator == "*" ]]; then
                result=$((num1 * num2))
            elif [[ $operator == "/" ]]; then
                result=$((num1 / num2))
            else
                echo "Invalid operator, please try again."
                continue
            fi
            ;;
    esac

    echo "The result is: $result"

    read -p "Would you like to perform another calculation? (Y/N): " choice
    case $choice in
        [Yy]*)
            continue
            ;;
        [Nn]*)
            echo "Goodbye!"
            exit 0
            ;;
        *)
            echo "Invalid choice, exiting."
            exit 1
            ;;
    esac
done

"calculator.sh" 46L, 907C
```

```
demo@ubuntu:/home/snehal/loop$ ./calculator.sh
Welcome to the calculator script!
Please enter the first number: 34
Please enter the operator (+, -, *, /): +
Please enter the second number: 34
The result is: 68
Would you like to perform another calculation? (Y/N): y
Please enter the first number: 45
Please enter the operator (+, -, *, /): -
Please enter the second number: 23
The result is: 22
Would you like to perform another calculation? (Y/N): y
Please enter the first number: 23
Please enter the operator (+, -, *, /): *
Please enter the second number: 17
The result is: 391
Would you like to perform another calculation? (Y/N): y
Please enter the first number: 45
Please enter the operator (+, -, *, /): /
Please enter the second number: 7
The result is: 6
Would you like to perform another calculation? (Y/N): n
Goodbye!
demo@ubuntu:/home/snehal/loop$
```