ULI101 Linux Command Practice – Advanced Bash Shell Scripting & Start-up Files

Instructions:

Write the answer to each question below the question in the space provided.
You can “wrap-around” the answer on separate lines if you need more space.

1. Write code for a Bash shell script will clear the screen and prompts the user for their **age**.

If the age entered is **less than 65**, then display a message that the person is NOT eligible to retire. If the age is **equal to 65**, then display a message that the person just turned 65 and can retire. If the age is **greater than 65**, then display the message that the user is over 65 and why have they not have already retired already?

->

```
#!/bin/bash
clear
echo -n "Enter your age: "
read age

if ! [[ "$age" =~ ^[0-9]+$ ]]; then
  echo "Error: Please enter a valid integer."
  exit 1
fi

if [ "$age" -lt 65 ]; then
  echo "You are NOT eligible to retire."
elif [ "$age" -eq 65 ]; then
  echo "Congratulations! You just turned 65 and can retire."
else
  echo "You are over 65. Why have you not retired already?"
fi
```

2. Add code to script in the previous question, to force the user to enter only an **integer** to provide error-checking for this shell script.

->

```
#!/bin/bash
clear
echo -n "Enter your age: "
read age

if ! [[ "$age" =~ ^[0-9]+$ ]]; then
  echo "Error: Please enter a valid integer."
  exit 1
fi
```

```
if [ "$age" -lt 65 ]; then
  echo "You are NOT eligible to retire."
elif [ "$age" -eq 65 ]; then
  echo "Congratulations! You just turned 65 and can retire."
else
  echo "You are over 65. Why have you not retired already?"
fi
```

3. Write code for a Bash shell script that will prompt the user for a **valid POSTAL CODE**.
A valid postal code consists of the following format: **`x#x  #x#`**
where **X** represents an upper or lowercase letter
and **#** represents a number from 0-9

Also VALID postal codes can consist of **no spaces** or **one or more spaces** in the format shown above.

If the user enters an INVALID postal code, indicate an error and allow the user to enter the VALID postal code. When the user enters a VALID postal code, then clear the screen and display the VALID postal code.

->

```
#!/bin/bash
clear
while :
do
  echo -n "Enter a valid postal code: "
  read postalCode

  if [[ "$postalCode" =~ ^[A-Za-z][0-9][A-Za-z] [0-9][A-Za-z][0-9]$ ]]; then
    clear
    echo "Valid Postal Code: $postalCode"
    break
  else
    echo "Error: Invalid postal code. Please enter a valid one."
  fi
done
```

4. Write code that works similar to the previous question, but have it read an input file called **unchecked-postalcodes.txt** and only save VALID postal codes to a file called: **valid-postalcodes.txt**

Design your Bash Shell script to only run if the user enters TWO ARGUMENTS:

unchecked-postalcodes.txt and **valid-postalcodes.txt**

Otherwise, display an **error message** and immediately **exit** your Bash Shell script with a **false** exit value.

->

```
#!/bin/bash
if [ "$#" -ne 2 ]; then
  echo "Error: Please provide two arguments - input and output files."
  exit 1
fi

input_file="$1"
output_file="$2"

while read -r postalCode; do
  if [[ "$postalCode" =~ ^[A-Za-z][0-9][A-Za-z] [0-9][A-Za-z][0-9]$ ]]; then
    echo "$postalCode" >> "$output_file"
  else
    echo "Error: Invalid postal code - $postalCode"
  fi
done < "$input_file"
```

5. What is the purpose of the **/etc/profile** startup file?
-> It is a system-wide shell profile file and runs at login and sets environment variables and executes commands for all users.

6. What is the purpose of the **/etc/bashrc** startup file?
-> It is a system-wide Bash configuration file and it is sourced by all Bash shells on the system.

7. What is the purpose of the **~/.bashrc** startup file?
-> It is the user-specific Bash configuration file and it is sourced for each interactive Bash shell session.

8. What is the purpose of the **~/.bash_profile** startup file?
-> It is the user-specific login script and it is executed when a user logs in.

9. What is the purpose of the **~/.bash_logout** file?
-> It is the user-specific logout script and it is executed when a user logs out.

10. Write code for the **~/.bashrc** file below to clear the screen, welcome the user by their username, and display a list of all users currently logged into your Matrix server. Insert blank lines between each of those elements.

->

```
# ~/.bashrc
clear
echo "Welcome, $USER!"
echo "Users currently logged in:"
who | awk '{print $1}' | sort | uniq
```

11. Write a command to run the recently created **~/.bashrc** startup file from the previous question without exiting and re-logging into your Matrix account.
-> source ~/.bashrc