

Bash Scripting Day 3 Assignment

Mohamed Emary

January 29, 2025

1 Assignment Questions

1.1 Question 1

Write a script called `mycase`, using the `case` utility to check the type of character entered by a user:

- a. Upper Case
- b. Lower Case
- c. Number
- d. Nothing

```
1  #!/usr/bin/bash
2
3  shopt -s extglob
4
5  echo "enter a char"
6  read ch
7
8  case $ch in
9    [a-z])
10     echo "lowercase"
11     ;;
12   [A-Z])
13     echo "uppercase"
14     ;;
15   [0-9])
16     echo "number"
17     ;;
18   *)
19     echo "Nothing"
20     ;;
21  esac
```

1.2 Question 2

Enhanced the previous script, by checking the type of string entered by a user:

- a. Upper Cases
- b. Lower Cases
- c. Numbers
- d. Mix
- e. Nothing

```
1  #!/bin/bash
2
3  shopt -s extglob
4
5  echo Please type your string:
6  read input
7
8  case $input in
9  +([a-z]))
10     echo Lower Case
11     ;;
12  +([A-Z]))
13     echo Upper Case
14     ;;
15  +([0-9]))
16     echo Number
17     ;;
18  +([0-9] | [a-z] | [A-Z]))
19     echo Mix
20     ;;
21  *)
22     echo Nothing
23     ;;
24  esac
```

1.3 Question 3

Write a script called mychmod using for utility to give execute permission to all files and directories in your home directory.

```
1  #!/bin/bash
2
3  echo "Press Enter to give execute permission to all files and directories
4  ↪ in your home directory"
5  echo "Or press Ctrl+C to cancel"
6  read -r
7
8  for item in ~/*; do
9      echo "$item"
10     chmod +x "$item"
11 done
```

Question 6

```
11 | echo "Finished"
```

1.4 Question 4

Write a script called `mybackup` using `for` utility to create a backup of only files in your home directory.

```
1  #!/bin/bash
2
3  echo "Press enter to start backup, or Ctrl+C to cancel"
4  read input
5  echo Backup started...
6
7  backup_folder="$HOME/home_backup"
8
9  mkdir "$backup_folder"
10
11 for item in "$HOME"/*; do
12     if [ -f "$item" ]; then
13         cp "$item" "$backup_folder"
14     fi
15 done
16
17 tar -cvf "$HOME/$backup_folder.tar" "$HOME/home_backup"
18 rm -r "$backup_folder"
19 echo "Backup completed: $backup_folder.tar"
```

1.5 Question 5

Write a script called `mymail` using `for` utility to send a mail to all users in the system.
Note: write the mail body in a file called `mtemplate`.

```
1  #!/usr/bin/bash
2
3  users=$(awk -F: '{print $1}' /etc/passwd)
4  for user in $users; do
5      echo "Sending mail to $user"
6      cat "mtemplate" | mail -s "Test Mail" "$user"
7  done
8
9  echo "Finished"
```

1.6 Question 6

Write a script called `chkmail` to check for new mails every 10 seconds.
Note: mails are saved in `/var/mail/username`.

Since all emails of a user are stored in a single file this script stores the current checksum of the mail file and compares it with the previous checksum. If the checksums are different, it means that the mail file has been updated (the user has new mail).

Question 7

```
1  #!/bin/bash
2
3  mail_file="/var/mail/$USER"
4  old_hash=""
5
6  while true; do
7      if [ -f "$mail_file" ]; then
8          current_hash=$(md5sum "$mail_file" | cut -d' ' -f1)
9
10         if [ -n "$old_hash" ] && [ "$current_hash" != "$old_hash" ]; then
11             echo "You have new mail!"
12         fi
13
14         old_hash=$current_hash
15     else
16         echo "Mail file does not exist"
17     fi
18     sleep 10
19 done
```

1.7 Question 7

What is the output of the following script

```
1  #!/bin/ksh
2
3  typeset -i n1
4  typeset -i n2
5
6  n1=1
7  n2=1
8
9  while test $n1 -eg $n2; do
10     n2=$n2+1
11     print $n1
12     if [ $n1 -gt $n2 ]; then
13         break
14     else
15         continue
16     fi
17     n1=$n1+1
18     print $n2
19 done
```

The script above will print 1 once then the loop condition will be false and the loop will exit.

1.8 Question 8

Create the following menu:

- a. Press 1 to `ls`
- b. Press 2 to `ls -a`
- c. Press 3 to `exit`

Using `select` utility then `while` utility.

```
1  #!/bin/bash
2
3  select option in "ls" "ls -a" "exit"; do
4      case $option in
5          "ls")
6              ls
7              ;;
8          "ls -a")
9              ls -a
10             ;;
11         "exit")
12             break
13             ;;
14         *)
15             echo Wrong option
16             ;;
17         esac
18     done
```

1.9 Question 9

Write a script called `myarr` that ask a user how many elements he wants to enter in an array, fill the array and then print it.

```
1  #! /bin/bash
2
3  echo "How many elements do you want to enter?"
4  read num
5
6  declare -a myarr
7  for ((i = 0; i < $num; i++)); do
8      echo "Array element:"
9      read user_input
10     $myarr[$i] = $user_input
11 done
```

1.10 Question 10

Write a script called `myavg` that calculate average of all numbers entered by a user.
Note: use arrays

```
1  #!/bin/bash
2
```

Question 11

```
3 echo "Enter the number of numbers:"
4 read num
5
6 declare -a myarr
7 sum=0
8
9 for ((i = 0; i < $num; i++)); do
10     echo "enter the array element"
11     read user_input
12
13     myarr[$i]=$user_input
14     sum=$((sum + user_input))
15 done
16
17 avg=$((echo "scale=2; $sum / $num" | bc))
18 echo "average is $avg"
```

1.11 Question 11

Write a function called `mysq` that calculate square if its argument.

```
1 #!/bin/bash
2
3 function square {
4     typeset -i sq
5     ((sq = $1 * $1))
6     echo $sq
7 }
8 echo "The square is: $(square $1)"
```