SP-Homework-1

Name: Suvonkulov Abdulaziz   ID:220456

Question 1

Question 1

Write a shell script that takes a filename as an argument and uses `grep` to search for a specific pattern within the file. Implement error handling using `trap` to catch any errors that may occur during execution.

Shell Script Code

```
# error handling
trap ’echo␣"unexpected␣error"␣exit’ ERR
#checking
if [ $# -ne 2]; then
    echo "Usage:␣$0␣<filename>␣<pattern>"
    exit 1
fi
# check the readable or exist
if [ ! -r "$1" ]; then
    echo "Error:␣$1␣not␣found␣or␣not␣readable"
    exit 1
fi
# searcing the pattern
grep -n "$2" "$1" || echo "No␣matches"
```

Terminal Output

```
felixnowman@Felixs-MacBook-Air HM-1 % chmod +x Q1.sh
felixnowman@Felixs-MacBook-Air HM-1 % ./Q1.sh sample.txt "hello"
./Q1.sh: line 4: [: missing `]'
Error:␣sample.txt␣not␣found␣or␣not␣readable
felixnowman@Felixs-MacBook-Air␣HM-1␣%␣vim␣sample.txt
felixnowman@Felixs-MacBook-Air␣HM-1␣%␣cat␣sample.txt
Hello
hello
1233
4231
#!@#
@@
#!3213
GSH
System␣programing`
felixnowman@Felixs-MacBook-Air␣HM-1␣%␣./Q1.sh␣sample.txt␣"hello"
./Q1.sh:␣line␣4:␣[:␣missing␣`]'
2:hello
felixnowman@Felixs-MacBook-Air HM-1 %
```

Question 2

Question 2

Create a shell script that demonstrates the usage of `set`, `unset`, and `shift` commands. Your script should receive command-line arguments, set some variables, unset them, and then shift the arguments to demonstrate the change.

Shell Script Code

```
echo "before␣set:␣$1=$1,␣$2=$2,␣$3=$3"

# using 'set' to change positional parameters
set s1 s2 s3
echo "After␣set:␣$1=$1,␣$2=$2,␣$3=$3"

# using 'unset' to remove a variable
var="Hello"
echo "Before␣unset:␣var=$var"
unset var
echo "After␣unset:␣var=${var:-Unset}"

# using 'shift' to move positional parameters
shift
echo "After␣shift:␣$1=$1,␣$2=$2,␣$3=$3"
```

Terminal Output

```
felixnowman@Felixs-MacBook-Air HM-1 % chmod +x Q2.sh
felixnowman@Felixs-MacBook-Air HM-1 % ./Q2.sh h1 h2 h3
before set: $1=h1, $2=h2, $3=h3
After set: $1=s1, $2=s2, $3=s3
Before unset: var=Hello
After unset: var=Unset
After shift: $1=s2, $2=s3, $3=
felixnowman@Felixs-MacBook-Air HM-1 %
```

Question 3

Question 3

Write a shell script that uses a `for` loop to iterate over a list of filenames in a directory obtained using the `find` command. Display each filename found.

Shell Script Code

```
#save directory in dir, by defolt currect directory
DIR=${1:-.}
echo "Searching␣file␣in␣$DIR"

#using find to get file from directory

for file in $(find "$DIR" -type f); do
    echo "Found␣file:␣$file"
done
```

Terminal Output

```
felixnowman@Felixs-MacBook-Air HM-1 % chmod +x Q3.sh
felixnowman@Felixs-MacBook-Air HM-1 % ./Q3.sh
Searching file in .
Found file: ./Q3.sh
Found file: ./Q2.sh
Found file: ./Q1.sh
Found file: ./sample.txt
felixnowman@Felixs-MacBook-Air HM-1 % ./Q3.sh ../
Searching file in ../
Found file: ..//HM-1/Q3.sh
Found file: ..//HM-1/Q2.sh
Found file: ..//HM-1/Q1.sh
Found file: ..//HM-1/sample.txt
Found file: ..//LAB-1/LAB-1-SP.pdf
Found file: ..//.DS_Store
Found file: ..//LAB-2/lab2_Task1.pdf
Found file: ..//LAB-2/codes/abzal.c
Found file: ..//LAB-2/codes/Abdurashid.c
Found file: ..//LAB-2/codes/output
Found file: ..//LAB-2/codes/lib.h
Found file: ..//LAB-2/codes/lab2-Task-2.c
Found file: ..//LAB-2/codes/Abzal.o
Found file: ..//LAB-2/codes/lab-2_Task1.c
Found file: ..//LAB-2/codes/Abdurashid.o
Found file: ..//LAB-2/codes/libfoo.a
Found file: ..//LAB-5/t1output
Found file: ..//LAB-5/task3.c
Found file: ..//LAB-5/output
Found file: ..//LAB-5/t2output
Found file: ..//LAB-5/lab5
Found file: ..//LAB-5/task2.c
Found file: ..//LAB-5/task1.c
Found file: ..//LAB-5/Untitled.pdf
felixnowman@Felixs-MacBook-Air HM-1 %
```

Question 4

Question 4

Develop a shell script that utilizes a `while` loop and `case` statement to continuously prompt the user for input until they enter a specific keyword to exit. The script should display a menu with options for the user to choose from using `echo` and `read`.

Shell Script Code

```
while true; do
    echo "Enter␣1␣to␣get␣currect␣date"
    echo "Enter␣2␣to␣get␣currect␣directory"
    echo "Enter␣0␣to␣exit"

    read -p "Enter␣your␣choise" choise

    case $choise in
    1) echo "Currect␣date␣%(date)"
    ;;
    2) echo "Currect␣directory"
       ls -l
    ;;
    0) echo "GoodBye"
       exit 0
    ;;
    esac
done
```

Terminal Output

```
felixnowman@Felixs-MacBook-Air HM-1 % chmod +x Q4.sh
felixnowman@Felixs-MacBook-Air HM-1 % ./Q4.sh
Enter 1 to get currect date
Enter 2 to get currect directory
Enter 0 to exit
Enter your choise 1
Currect date %(date)
Enter 1 to get currect date
Enter 2 to get currect directory
Enter 0 to exit
Enter your choise2
Currect directory
total 40
-rwxr-xr-x  1 felixnowman  staff  311 Feb 25 17:32 Q1.sh
-rwxr-xr-x  1 felixnowman  staff  368 Feb 26 10:12 Q2.sh
-rwxr-xr-x  1 felixnowman  staff  209 Feb 26 10:39 Q3.sh
-rwxr-xr-x  1 felixnowman  staff  292 Feb 26 11:13 Q4.sh
-rw-r--r--  1 felixnowman  staff   60 Feb 26 09:45 sample.txt
Enter 1 to get currect date
Enter 2 to get currect directory
Enter 0 to exit
Enter your choise0
GoodBye
felixnowman@Felixs-MacBook-Air HM-1 % vim Q4.sh
felixnowman@Felixs-MacBook-Air HM-1 % ./Q4.sh
Enter 1 to get currect date
Enter 2 to get currect directory
Enter 0 to exit
Enter your choise1
Currect date Wed Feb 26 11:14:59 +05 2025
Enter 1 to get currect date
Enter 2 to get currect directory
Enter 0 to exit
Enter your choise0
GoodBye
felixnowman@Felixs-MacBook-Air HM-1 %
```

Question 5

Question 5

Design a shell script that takes two numbers as positional parameters and calculates their sum, difference, product, and quotient using `expr`. Display the results with appropriate messages.

Shell Script Code

```
[ $# -ne 2 ] && echo "usage: $0 <num1> <num2>" && exit 1

num1=$1
num2=$2

echo "sum:␣$(expr␣$num1␣+␣$num2)"
echo "difference:␣$(expr␣$num1␣-␣$num2)"
echo "product:␣$(expr␣$num1␣\*␣$num2)"

if [ $num2 -ne 0 ]; then
    quotient=$(expr $num1 / $num2)
else
    quotient="Not␣divisible"
fi

echo "Division:␣$quotient"
```

Terminal Output

```
felixnowman@Felixs-MacBook-Air HM-1 % chmod +x Q5.sh
felixnowman@Felixs-MacBook-Air HM-1 % ./Q5.sh 5 10
sum: 15
difference: -5
product: 50
Division: 0
felixnowman@Felixs-MacBook-Air HM-1 % ./Q5.sh 15 10
sum: 25
difference: 5
product: 150
Division: 1
felixnowman@Felixs-MacBook-Air HM-1 %
```

Question 6

Question 6

Create a shell script that checks if a file exists. If the file exists, it prints a message saying "File already exists." If the file does not exist, it uses the `touch` command to create the file and prints a message saying "File created successfully." Utilize the AND list () to execute the touch command only if the file doesn't exist. Once the file is created use `find` command with any test and by any action.

Shell Script Code

```
#get input from user

read -p "Enter␣file␣name:␣" filename

#check file exits or not

if [ -f "$filename" ]; then
    echo "file␣exits"
else
    touch "$filename" && ehco "file␣created"
fi

#for find the  path of the file
find . -name "$filename" -type f -exec ls -l {} \;
```

Terminal Output

```
felixnowman@Felixs-MacBook-Air HM-1 % chmod +x Q6.sh
felixnowman@Felixs-MacBook-Air HM-1 % ./Q6.sh sample.txt
Enter file name: sample.txt
file exits
-rw-r--r--  1 felixnowman  staff  60 Feb 26 09:45 ./sample.txt
felixnowman@Felixs-MacBook-Air HM-1 % ./Q6.sh s
Enter file name: something.txt
./Q6.sh: line 10: ehco: command not found
-rw-r--r--  1 felixnowman  staff  0 Feb 28 10:56 ./something.txt
felixnowman@Felixs-MacBook-Air HM-1 % vim Q6.sh
felixnowman@Felixs-MacBook-Air HM-1 % ./Q6.sh
Enter file name: something1.txt
file created
-rw-r--r--  1 felixnowman  staff  0 Feb 28 10:56 ./something1.txt
felixnowman@Felixs-MacBook-Air HM-1 %
```

Question 7

Question 7

Create a script that accepts a mathematical expression from the user and evaluates it using ${(( ))} and `expr`.

Shell Script Code

```
#get input from user

read -p "enter␣mathmatical␣expressin" exp

#evaluate using $(())

echo "result␣with␣((␣)):␣␣$((exp))"

#evaluate using expr

if result=$(expr $exp 2>/dev/null); then
    echo "result␣expr:␣$result"
else
    echo "invalid␣format,␣plese␣write␣like␣this␣'2␣+␣3'"
fi
```

Terminal Output

```
felixnowman@Felixs-MacBook-Air HM-1 % chmod +x Q7.sh
felixnowman@Felixs-MacBook-Air HM-1 % ./Q7.sh
enter mathmatical expressin 2+3
result with (( )):  5
result expr: 2+3
felixnowman@Felixs-MacBook-Air HM-1 % ./Q7.sh
enter mathmatical expressin 2 + 3
result with (( )):  5
result expr: 5
```

Question 8

Question 8

Implement a script that creates a backup of all `.txt` files in a directory, appending a timestamp to the backup name.

Shell Script Code

```
# create a direction
backup_dir="backup"
mkdir -p "$backup_dir"

# get current timestamp
timestamp=$(date -u +"%Y%m%d%H%M%S")

# backup all .txt files in the current directory
for file in *.txt; do
    if [ -f "$file" ]; then
        cp "$file" "$backup_dir/${file%.*}_$timestamp.txt"
        echo "Backup␣created:␣$backup_dir/${file%.*}_$timestamp.txt"
    fi
done

echo "done"
```

Terminal Output

```
felixnowman@Felixs-MacBook-Air HM-1 % chmod +x Q8.sh
felixnowman@Felixs-MacBook-Air HM-1 % ./Q8.sh
Backup created: backup/sample_20250301072636.txt
Backup created: backup/something_20250301072636.txt
Backup created: backup/something1_20250301072636.txt
done
```

Question 9

Question 9

Create a script that continuously monitors a given directory and notifies when a new file is added.

Shell Script Code

```
# directory to monitor
monitor_dir="."

# notify when a new file is added
#installed fswatch for notification
fswatch -0 "$monitor_dir" | while read -d "" new_file; do
    echo "New␣file␣added:␣$new_file"
done
```

Terminal Output

```
felixnowman@Felixs-MacBook-Air HM-1 % chmod +x Q9.sh
felixnowman@Felixs-MacBook-Air HM-1 % ./Q9.sh
New file added: /Users/felixnowman/Desktop/course-3/Spring/SP/HM
    -1/.test.txt.swx
New file added: /Users/felixnowman/Desktop/course-3/Spring/SP/HM
    -1/.test.txt.swp
New file added: /Users/felixnowman/Desktop/course-3/Spring/SP/HM
    -1/.test.txt.swp
New file added: /Users/felixnowman/Desktop/course-3/Spring/SP/HM
    -1/test.txt
New file added: /Users/felixnowman/Desktop/course-3/Spring/SP/HM
    -1/.test.txt.swp
^C%
felixnowman@Felixs-MacBook-Air HM-1 %
```

Question 10

Question 10

Write a script that extracts usernames from the `/etc/passwd` file and prints them in sorted order.

Shell Script Code

```
cut -d: -f1 /etc/passwd | sort
# cut extracts specific fields from a file.
# -d: sets : as the delimiter since /etc/passwd uses colons.
# -f1 selects the first field, which contains usernames.
# sort arranges them in alphabetical order.
```

Terminal Output

```
felixnowman@Felixs-MacBook-Air HM-1 % chmod +x Q10.sh
felixnowman@Felixs-MacBook-Air HM-1 % ./Q10.sh
#
#
# Note that this file is consulted directly only when the system
    is running
# Open Directory.
# Open Directory.
# See the opendirectoryd(8) man page for additional information
    about
# User Database
# in single-user mode.  At other times this information is
    provided by
##
##
_accessoryupdater
_amavisd
_analyticsd
_aonsensed
_appinstalld
_appleevents
_applepay
_appowner
_appserver
_appstore
_ard
_assetcache
_astris
_atsserver
_audiomxd
_avbdeviced
_avphidbridge
_backgroundassets
_biome
_calendar
_captiveagent
_ces
_clamav
_cmiodalassistants
_coreaudiod
_coremediaiod
... (and more lines)
```