

PRACTICAL 2

[Shell Scripting]

Q1. Write a shell script to generate mark- sheet of a student. Take 3 subjects, calculate and display total marks, percentage and Class obtained by the student.

```
MINGW64:/c/Users/PARAS/OneDrive/Desktop

PARAS@LAPTOP-DL5TDAQK MINGW64 ~/OneDrive/Desktop (main)
$ echo "Enter marks of OS"
read m1

echo "Enter marks of English"
read m2

echo "Enter marks of Maths"
read m3

total=$((m1 + m2 + m3))
percentage=$((total / 3))

echo "Total Marks: $total"
echo "Percentage: $percentage"

if [ $percentage -ge 75 ]; then
    echo "Class: First Class"
elif [ $percentage -ge 60 ]; then
    echo "Class: Second Class"
elif [ $percentage -ge 50 ]; then
    echo "Class: Pass"
else
    echo "Class: Fail"
fi
Enter marks of OS
70
Enter marks of English
65
Enter marks of Maths
90
Total Marks: 225
Percentage: 75
Class: First Class

PARAS@LAPTOP-DL5TDAQK MINGW64 ~/OneDrive/Desktop (main)
$ |
```

Q2. Write a menu driven shell script which will print the following menu and execute the given task.

1. Display calendar of current month
2. Display today's date and time
3. Display usernames those are currently logged in the system
4. Display your terminal number

```
MINGW64/c/Users/PARAS/OneDrive/Desktop

PARAS@LAPTOP-DL5TDAQK MINGW64 ~/OneDrive/Desktop (main)
$ while true
do
    clear

    echo "SYSTEM INFORMATION MENU (Git Bash)"
    echo "1. Display calendar of current month"
    echo "2. Display today's date and time"
    echo "3. Display usernames currently logged in"
    echo "4. Display your terminal number"
    echo "5. Exit"
    echo "=====
    echo -n "Enter your choice (1-5): "
    read choice

    case $choice in
    1)
        echo "Calendar of current month:"
        cal
        echo
        read -p "Press Enter to continue..."
        ;;
    2)
        echo "Today's date and time:"
        date
        echo
        read -p "Press Enter to continue..."
        ;;
    3)
        echo "Users currently logged in:"
        who
        echo
        read -p "Press Enter to continue..."
        ;;
    4)
        echo "Your terminal number:"
        tty
        echo
        read -p "Press Enter to continue..."
        ;;
    5)
        echo "Exiting program..."
        exit
        ;;
    *)
        echo "Invalid choice! Please enter 1-5."
        read -p "Press Enter to continue..."
        ;;
    esac
done
```

```
MINGW64/c/Users/PARAS/OneDrive/Desktop
SYSTEM INFORMATION MENU (Git Bash)
1. Display calendar of current month
2. Display today's date and time
3. Display usernames currently logged in
4. Display your terminal number
5. Exit
=====
Enter your choice (1-5): 1
Calendar of current month:
bash: cal: command not found

Press Enter to continue...|
```

```
MINGW64/c/Users/PARAS/OneDrive/Desktop
SYSTEM INFORMATION MENU (Git Bash)
1. Display calendar of current month
2. Display today's date and time
3. Display usernames currently logged in
4. Display your terminal number
5. Exit
=====
Enter your choice (1-5): 2
Today's date and time:
Sat Jan 24 21:18:07 IST 2026

Press Enter to continue...|
```

```
MINGW64/c/Users/PARAS/OneDrive/Desktop
SYSTEM INFORMATION MENU (Git Bash)
1. Display calendar of current month
2. Display today's date and time
3. Display usernames currently logged in
4. Display your terminal number
5. Exit
=====
Enter your choice (1-5): 3
Users currently logged in:

Press Enter to continue...|
```

```
MINGW64/c/Users/PARAS/OneDrive/Desktop
SYSTEM INFORMATION MENU (Git Bash)
1. Display calendar of current month
2. Display today's date and time
3. Display usernames currently logged in
4. Display your terminal number
5. Exit
=====
Enter your choice (1-5): 4
Your terminal number:
/dev/pty0

Press Enter to continue...|
```

Q3. Write a shell script which will generate first n fibonacci numbers like: 1, 1, 2, 3, 5, 13

```
MINGW64:/c/Users/PARAS/OneDrive/Desktop

PARAS@LAPTOP-DL5TDAQK MINGW64 ~/OneDrive/Desktop (main)
$ echo "Enter number of terms:"
read n

a=1
b=1

echo "Fibonacci Series:"
echo -n "$a $b "

count=2

while [ $count -lt $n ]
do
    c=$((a + b))

    # Skip printing 8
    if [ $c -ne 8 ]; then
        echo -n "$c "
        count=$((count + 1))
    fi

    a=$b
    b=$c
done
Enter number of terms:
6
Fibonacci Series:
1 1 2 3 5 13
PARAS@LAPTOP-DL5TDAQK MINGW64 ~/OneDrive/Desktop (main)
$
```

Q4. Write a shell script which will accept a number b and display first n prime numbers as output

```
MINGW64:/c/Users/PARAS/OneDrive/Desktop

PARAS@LAPTOP-DL5TDAQK MINGW64 ~/OneDrive/Desktop (main)
$ echo "Enter how many prime numbers you want:"
read n

count=0
num=2

echo "First $n prime numbers are:"

while [ $count -lt $n ]
do
    flag=0
    for (( i=2; i<=num/2; i++ ))
    do
        if [ $(num % i) -eq 0 ]; then
            flag=1
            break
        fi
    done

    if [ $flag -eq 0 ]; then
        echo -n "$num "
        count=$((count + 1))
    fi

    num=$((num + 1))
done
Enter how many prime numbers you want:
9
First 9 prime numbers are:
2 3 5 7 11 13 17 19 23
PARAS@LAPTOP-DL5TDAQK MINGW64 ~/OneDrive/Desktop (main)
$ |
```

Q5. Write menu driven program for file handling activity

1. Creation of file
2. Write content in the file
3. Upend file content
4. Delete file content

```
MINGW64/c/Users/PARAS/OneDrive/Desktop
PARAS@LAPTOP-DL5TDAQK MINGW64 ~/OneDrive/Desktop (main)
$ echo "Enter file name:"
read fname

while true
do
    echo "----- FILE MENU -----"
    echo "1. Create File"
    echo "2. Write Content in File"
    echo "3. Append File Content"
    echo "4. Delete File Content"
    echo "5. Exit"
    echo "-----"
    echo "Enter your choice:"
    read ch

    case $ch in
        1) touch $fname
            echo "File created successfully"
            ;;
        2) echo "Enter content (Press CTRL+D to save):"
            cat > $fname
            echo "Content written successfully"
            ;;
        3) echo "Enter content to append (Press CTRL+D to save):"
            cat >> $fname
            echo "Content appended successfully"
            ;;
        4) > $fname
            echo "File content deleted successfully"
            ;;
        5) echo "Exiting program..."
            exit
            ;;
        *) echo "Invalid choice"
            ;;
    esac
done
Enter file name:
os prac 2.txt
> done
```

```
----- FILE MENU -----
1. Create File
2. Write Content in File
3. Append File Content
4. Delete File Content
5. Exit
-----
Enter your choice:
1
File created successfully
----- FILE MENU -----
1. Create File
2. Write Content in File
3. Append File Content
4. Delete File Content
5. Exit
-----
Enter your choice:
2
Enter content (Press CTRL+D to save):
bash: $fname: ambiguous redirect
Content written successfully
----- FILE MENU -----
1. Create File
2. Write Content in File
3. Append File Content
4. Delete File Content
5. Exit
-----
Enter your choice:
3
Enter content to append (Press CTRL+D to save):
bash: $fname: ambiguous redirect
Content appended successfully
----- FILE MENU -----
1. Create File
2. Write Content in File
3. Append File Content
4. Delete File Content
5. Exit
-----
Enter your choice:
4
bash: $fname: ambiguous redirect
File content deleted successfully
----- FILE MENU -----
1. Create File
2. Write Content in File
3. Append File Content
4. Delete File Content
5. Exit
-----
Enter your choice:
```