PRACTICAL-3

1. 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/Admin/Desktop/OS_CD24043

```
echo "Enter the name of the student: "
read student_name

echo "Enter marks for Subject 1: "
read subject1

echo "Enter marks for Subject 2: "
read subject2

echo "Enter marks for Subject 3: "
read subject3

total_marks=$((subject1 + subject2 + subject3))

max_marks=100
max_total_marks=$((max_marks * 3))

percentage=$(echo "scale=2; ($total_marks / $max_total_marks) * 100" | bc)

if (( $(echo "$percentage >= 60" | bc -l) )); then
    class="First Class"
elif (( $(echo "$percentage >= 50" | bc -l) )); then
    class="Second Class"
elif (( $(echo "$percentage >= 40" | bc -l) )); then
    class="Third Class"
else
    class="Fail"
fi

echo "Total Marks: $total_marks/$max_total_marks"
echo "Percentage: $percentage%"
echo "Class Obtained: $class"
Enter the name of the student:
sanket
Enter marks for Subject 1:
35
Enter marks for Subject 2:
35
Enter marks for Subject 3:
35
bash: bc: command not found
bash: bc: command not found
bash: bc: command not found
bash: bc: command not found
Total Marks: 105/300
Percentage: %
Class Obtained: Fail
```

2. Write a menu driven shell script which will print the following menu and execute the given task.
Display calendar of current month

Display today's date and time

Display usernames those are currently logged in the system

Display your terminal number

MINGW64:/c/Users/Admin/Desktop/OS_CD24043

```
Student@DESKTOP-JJ8CFNB MINGW64 ~/Desktop/OS_CD24043
$ #!/bin/bash

while true; do
    clear
    echo "------------------------ Menu ------------------------"
    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 "Please choose an option (1-5): "
    read choice

    case $choice in
        1)
            echo "Displaying calendar for the current month:"
            cal
            read -p "Press Enter to continue..."
            ;;
        2)
            echo "Today's date and time:"
            date
            read -p "Press Enter to continue..."
            ;;
        3)
            echo "Users currently logged in:"
            who
            read -p "Press Enter to continue..."
            ;;
        4)
            echo "Your terminal number is: $(tty)"
            read -p "Press Enter to continue..."
            ;;
        5)
            echo "Exiting... Goodbye!"
            exit 0
            ;;
        *)
            echo "Invalid choice! Please choose a valid option (1-5)."
            read -p "Press Enter to continue..."
            ;;
    esac
done
```

MINGW64:/c/Users/Admin/Desktop/OS_CD24043

```
------------------------ Menu ------------------------
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
------------------------------------------------------
Please choose an option (1-5):
2
Today's date and time:
Tue Jan 27 15:56:35 IST 2026
Press Enter to continue...
```

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

MINGW64:/c/Users/Admin/Desktop/OS_CD24043

```
Student@DESKTOP-JJ8CFNB MINGW64 ~/Desktop/OS_CD24043
$ #!/bin/bash

echo "Enter the value of n (number of Fibonacci numbers to generate):"
read n

# Initialize the first two Fibonacci numbers
a=0
b=1

echo "Fibonacci sequence:"
for (( i=1; i<=n; i++ ))
do
    echo -n "$b "
    # Update the Fibonacci sequence
    next=$((a + b))
    a=$b
    b=$next
done
echo
Enter the value of n (number of Fibonacci numbers to generate):
5
Fibonacci sequence:
1 1 2 3 5
```

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

MINGW64:/c/Users/Admin/Desktop/OS_CD24043

```
Student@DESKTOP-JJ8CFNB MINGW64 ~/Desktop/OS_CD24043
$ #!/bin/bash

echo "Enter the value of n (number of prime numbers to generate):"
read n

echo "Enter the value of b (upper limit for checking prime numbers):"
read b

count=0
num=2

echo "Prime numbers up to $b are:"

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

    if [ $is_prime -eq 1 ]; then
        if [ $num -le $b ]; then
            echo -n "$num "
            ((count++))
        fi
    fi
    ((num++))
done

echo
Enter the value of n (number of prime numbers to generate):
5
Enter the value of b (upper limit for checking prime numbers):
5
Prime numbers up to 5 are:
2 3 5
```

5. Write menu driven program for file handling activity.

Creation of file

Write content in the file

Upend file content

Delete file content

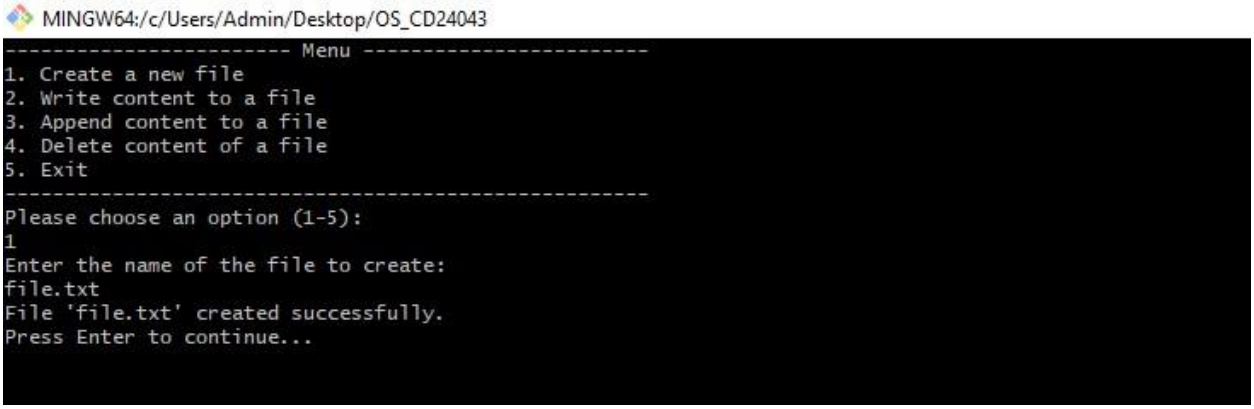
MINGW64:/c/Users/Admin/Desktop/OS_CD24043

```
while true; do
    clear
    echo "----------------------- Menu -----------------------"
    echo "1. Create a new file"
    echo "2. Write content to a file"
    echo "3. Append content to a file"
    echo "4. Delete content of a file"
    echo "5. Exit"
    echo "----------------------------------------------------"
    echo "Please choose an option (1-5):"
    read choice

    case $choice in
        1)
            echo "Enter the name of the file to create:"
            read filename
            if [ -e "$filename" ]; then
                echo "File already exists!"
            else
                touch "$filename"
                echo "File '$filename' created successfully."
            fi
            read -p "Press Enter to continue..."
            ;;

        2)
            echo "Enter the name of the file to write to:"
            read filename
            if [ ! -e "$filename" ]; then
                echo "File does not exist!"
            else
                echo "Enter the content you want to write (Ctrl+D to finish):"
                cat > "$filename"
                echo "Content written to '$filename'."
            fi
            read -p "Press Enter to continue..."
            ;;

        3)
            echo "Enter the name of the file to append to:"
            read filename
            if [ ! -e "$filename" ]; then
                echo "File does not exist!"
            else
                echo "Enter the content you want to append (Ctrl+D to finish):"
                cat >> "$filename"
                echo "Content appended to '$filename'."
            fi
done|esac    ;;ad -p "Press Enter to continue..."a valid option (1-5)."
```


MINGW64:/c/Users/Admin/Desktop/OS_CD24043
---------- Menu ----------
1. Create a new file
2. Write content to a file
3. Append content to a file
4. Delete content of a file
5. Exit

Please choose an option (1-5):
1
Enter the name of the file to create:
file.txt
File 'file.txt' created successfully.
Press Enter to continue...