

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/OSLab_CD24023
Class: Third Class

MAYUR@LAPTOP-5DSA01RI MINGW64 /c/OSLab_CD24023 (main)
$ echo "Enter marks of English"
read m1

echo "Enter marks of Maths"
read m2

echo "Enter marks of Science"
read m3

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

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

if [ $percentage -ge 75 ]; then
    echo "Class: Distinction"
elif [ $percentage -ge 60 ]; then
    echo "Class: First class"
elif [ $percentage -ge 40 ]; then
    echo "Class: Second class"
elif [ $percentage -ge 35 ]; then
    echo "Class: Third class"
else
    echo "Class: Fail"
fi
Enter marks of English
50
Enter marks of Maths
55
Enter marks of Science
56
Student: Total Marks = 161
Percentage = 53
Class: Second class
```

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/TOM/OneDrive/Desktop/OS_CD24016
FOIN@LAPTOP-MC73RDEH MINGW64 ~/OneDrive/Desktop/OS_CD24016 (main)
$ #!/bin/bash

while true
do
    echo "-----"
    echo "      MENU"
    echo "-----"
    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: "
    read choice
    case $choice in
        1) cal ;;
        2) date ;;
        3) who ;;
        4) tty ;;
        5) echo "Exiting program..." exit ;;
        *) echo "Invalid choice! Please try again." ;;
    esac
done
-----
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
-----
Enter your choice: 2
Sat Jan 24 21:25:11 IST 2026
-----
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
-----
Enter your choice: |
```

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

```
MINGW64:/c/OSLab_CD24023/Shell script/menu_script_git/menu_script_git/menu_script_git
Fibonacci series:
1 2 3 5 8 13 21 34 55 89 144 233 377 610 987
MAYUR@LAPTOP-5DSA01RI MINGW64 /c/OSLab_CD24023/Shell script/menu_script_git/menu_script_git/menu_script_git (master)
$ #!/bin/bash

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

echo
Enter how many prime numbers you want:
13
First 13 prime numbers are:
2 3 5 7 11 13 17 19 23 29 31 37 41
```

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

```
MINGW64:/c/OSLab_CD24023/Shell script/menu_script_git/menu_script_git/menu_script_git
Fibonacci series:
1 2 3 5 8 13 21 34 55 89 144 233 377 610 987
MAYUR@LAPTOP-5DSA01RI MINGW64 /c/OSLab_CD24023/Shell script/menu_script_git/menu_script_git/menu_script_git (master)
$ #!/bin/bash

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

echo
Enter how many prime numbers you want:
13
First 13 prime numbers are:
2 3 5 7 11 13 17 19 23 29 31 37 41
```

## 5. Write menu driven program for file handling activity

- Creation of file
- Write content in the file
- Upend file content
- Delete file content

```
MAYUR@LAPTOP-5DSA01RI MINGW64 /c/OSLab_CD24023/Shell script (main)
#!/bin/bash

while true
do
    echo "-----"
    echo "      MENU"
    echo "-----"
    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: "
    read choice

    case $choice in
        1)
            cal
            ;;
        2)
            date
            ;;
        3)
            who
            ;;
        4)
            tty
            ;;
        5)
            echo "Exiting program..."
            exit
            ;;
        *)
            echo "Invalid choice! Please try again."
            ;;
    esac
done

5)   echo "Exiting program..."  
exit  
:::  
*)  
ESAC ::ho "Invalid choice! Try again."  
r file name: MAYUR  
-----  
- FILE HANDLING MENU  
-----  
. Create file  
. Write content in file  
. Append file content  
. Delete file content  
. Exit  
-----  
nter your choice: 2  
ile created successfully.  
024023  
nter your choice: 2  
-----  
- FILE HANDLING MENU  
-----  
. Create file  
. Write content in file  
. Append file content  
. Delete file content  
. Exit  
-----  
nter your choice: 2  
-----  
- FILE HANDLING MENU  
-----  
. Create file  
. Write content in file  
. Append file content  
. Delete file content  
. Exit  
-----  
nter your choice: []
```



