

## PracticalNo : 2

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/sonus
sonus@DESKTOP-5EMSD6U MINGW64 ~
$ echo "===== MARK SHEET ====="

read -p "Enter marks for Subject 1: " s1
read -p "Enter marks for Subject 2: " s2
read -p "Enter marks for Subject 3: " s3

# Total marks
total=$((s1 + s2 + s3))

# Percentage (integer)
percentage=$((total / 3))

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

# Class calculation
if [ $percentage -ge 75 ]; then
    class="Distinction"
elif [ $percentage -ge 60 ]; then
    class="First Class"
elif [ $percentage -ge 50 ]; then
    class="Second Class"
elif [ $percentage -ge 35 ]; then
    class="Pass"
else
    class="Fail"
fi

echo "Class Obtained: $class"
echo "===== MARK SHEET ====="
Enter marks for Subject 1: 80
Enter marks for Subject 2: 90
Enter marks for Subject 3: 70
-----
Total Marks   : 240
Percentage    : 80 %
Class Obtained: Distinction
=====

sonus@DESKTOP-5EMSD6U MINGW64 ~
$ |
```

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

- I. Display calendar of current month
- II. Display today's date and time
- III. Display usernames those are currently logged in the system
- IV. Display your terminal number

```

sonus@DESKTOP-5EMSD6U MINGW64 ~
$ 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 "Enter your choice:"
read ch

if [ $ch -eq 1 ]; then
    cal
elif [ $ch -eq 2 ]; then
    date
elif [ $ch -eq 3 ]; then
    users
elif [ $ch -eq 4 ]; then
    tty
else
    echo "Invalid choice"
fi
1. Display calendar of current month
2. Display today's date and time
3. Display usernames currently logged in
4. Display your terminal number
Enter your choice:
2
Mon Jan 26 10:18:40 IST 2026

sonus@DESKTOP-5EMSD6U MINGW64 ~
$

```

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

```

sonus@DESKTOP-5EMSD6U MINGW64 ~
$ echo "Enter the value of n:"
read n

a=1
b=1

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

for (( i=3; i<=n; i++ ))
do
    c=$((a + b))
    echo -n "$c "
    a=$b
    b=$c
done

echo

Enter the value of n:
4
Fibonacci series:
1 1 2 3

sonus@DESKTOP-5EMSD6U MINGW64 ~
$ |

```

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

```
sonus@DESKTOP-5EMSD6U MINGW64 ~
$ echo "Enter the number of prime numbers to display:"
read n

count=0
num=2

is_prime() {
    local number=$1
    for (( i=2; i*i<=number; i++ ))
    do
        if (( number % i == 0 )); then
            return 1 # Not prime
        fi
    done
    return 0 # Prime
}

echo "First $n prime numbers:"
while [ $count -lt $n ]
do
    if is_prime $num; then
        echo -n "$num "
        ((count++))
    fi
    ((num++))
done

echo
Enter the number of prime numbers to display:
4
First 4 prime numbers:
2 3 5 7

sonus@DESKTOP-5EMSD6U MINGW64 ~
$ |
```



5. Write menu driven program for file handling activity

- I. Creation of file
- II. Write content in the file
- III. Upend file content
- IV. Delete file content

```
sonus@DESKTOP-5EMSD6U MINGW64 ~
$ echo "1. Create a file"
echo "2. Write content to a file"
echo "3. Append content to a file"
echo "4. Delete file content"
echo "Enter your choice:"
read choice

if [ $choice -eq 1 ]; then
    read -p "Enter file name: " file
    touch "$file"
    echo "File '$file' created."
elif [ $choice -eq 2 ]; then
    read -p "Enter file name: " file
    echo "Enter content (Ctrl+D to save):"
    cat > "$file"
    echo "Content written."
elif [ $choice -eq 3 ]; then
    read -p "Enter file name: " file
    echo "Enter content to append (Ctrl+D to save):"
    cat >> "$file"
    echo "Content appended."
elif [ $choice -eq 4 ]; then
    read -p "Enter file name: " file
    > "$file"
    echo "Content deleted."
else
    echo "Invalid choice!"
fi

1. Create a file
2. Write content to a file
3. Append content to a file
4. Delete file content
Enter your choice:
1
Enter file name: myfil.txt
File 'myfil.txt' created.

sonus@DESKTOP-5EMSD6U MINGW64 ~
$ |
```