Practical No. :3

USN NO. :CM24120

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

script.sh

```
#!/bin/bash

echo "Enter Student Name:"
read name

echo "Enter Subject 1 Name:"
read sub1
echo "Enter marks for $sub1:"
read m1

echo "Enter Subject 2 Name:"
read sub2
echo "Enter marks for $sub2:"
read m2

echo "Enter Subject 3 Name:"
read sub3
echo "Enter marks for $sub3:"
read m3

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

if [ $percentage -ge 60 ]; then
    class="First Class"
elif [ $percentage -ge 50 ]; then
    class="Second Class"
elif [ $percentage -ge 40 ]; then
    class="Pass"
else
    class="Fail"
fi

echo "--------------------------------"
echo "          MARK SHEET"
echo "--------------------------------"
echo "Student Name : $name"
echo
echo "Subject    | Marks"
echo "$sub1        $m1"
echo "$sub2        $m2"
echo "$sub3        $m3"
```

STDIN

```
jay
math
67
```

Output:

```
Enter Student Name:
Enter Subject 1 Name:
Enter marks for math:
Enter Subject 2 Name:
Enter marks for phys:
Enter Subject 3 Name:
Enter marks for chem:
--------------------------------
          MARK SHEET
--------------------------------
Student Name : jay

Subject      Marks
math         67
phys         79
chem         82
--------------------------------
Total Marks  : 228
Percentage   : 76%
Class        : First 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

```
#!/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 "Enter your choice:"
    read choice

    case $choice in
        1)
            echo "Calendar of current month (simplified):"
            echo "Sun Mon Tue Wed Thu Fri Sat"
            for i in {1..30}     # assuming 30 days
            do
                printf "%2d " $i
                if (( i % 7 == 0 )); then
                    echo
                fi
            done
            echo
            ;;
        2)
            echo "Today's date and time:"
            date
            ;;
        3)
            echo "Currently logged-in users (simulated):"
            echo "$USER"      # $USER is the current username
            ;;
        4)
            echo "Terminal number (simulated):"
            echo "/dev/tty1"  # cannot detect real terminal in online compiler
            ;;
        5)
            echo "Exiting..."
            exit
```

STDIN

```
3
4
5
```

Output:

```
----------------------------
           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:
Calendar of current month (simplified):
Sun Mon Tue Wed Thu Fri Sat
 1  2  3  4  5  6  7
 8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30
----------------------------
           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:
Today's date and time:
Mon Jan 19 03:50:10 PM UTC 2026
----------------------------
```

```
    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 "Enter your choice:"
    read choice

    case $choice in
        1)
            echo "Calendar of current month (simplified):"
            echo "Sun Mon Tue Wed Thu Fri Sat"
            for i in {1..30}     # assuming 30 days
            do
                printf "%2d " $i
                if (( i % 7 == 0 )); then
                    echo
                fi
            done
            echo
            ;;
        2)
            echo "Today's date and time:"
            date
            ;;
        3)
            echo "Currently logged-in users (simulated):"
            echo "$USER"      # $USER is the current username
            ;;
        4)
            echo "Terminal number (simulated):"
            echo "/dev/tty1"  # cannot detect real terminal in online compiler
            ;;
        5)
            echo "Exiting..."
            exit
            ;;
        *)
            echo "Invalid choice. Please enter 1-5."
            ;;
    esac
done
```

```
3
4
5
----------------------------
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:
Currently logged-in users (simulated):

----------------------------
           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:
Terminal number (simulated):
/dev/tty1
----------------------------
           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:
Exiting...
```

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

```
#!/bin/bash

echo "Enter the number of terms (n):"
read n

# First two Fibonacci numbers
a=1
b=1

echo "Fibonacci Series:"

if [ $n -ge 1 ]; then
    echo -n "$a "
fi

if [ $n -ge 2 ]; then
    echo -n "$b "
fi

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

echo
```

```
STDIN
7
```

Output:

```
Enter the number of terms (n):
Fibonacci Series:
1 1 2 3 5 8 13
```

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

```
#!/bin/bash

echo "Enter the number of prime numbers to display:"
read n

count=0       # how many primes we have found
num=2         # number to check

echo "First $n prime numbers:"

while [ $count -lt $n ]
do
    is_prime=1    # assume num is prime

    # check if num is divisible by any number from 2 to sqrt(num)
    for ((i=2; i*i<=num; i++))
    do
        if [ $((num % i)) -eq 0 ]; then
            is_prime=0
            break
        fi
    done

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

    num=$((num + 1))
done

echo
```

```
STDIN
10
```

Output:

```
Enter the number of prime numbers to display:
First 10 prime numbers:
2 3 5 7 11 13 17 19 23 29
```

5. Write menu driven program for file handling activity Creation of file • Write content in the file Upend file content Delete file content

```bash
#!/bin/bash

while true
do
    echo "----------------------"
    echo "FILE HANDLING MENU"
    echo "----------------------"
    echo "1. Create file"
    echo "2. Write content to file"
    echo "3. Append content to file"
    echo "4. Delete file content"
    echo "5. Exit"
    echo "Enter choice:"
    read choice

    case $choice in
        1)
            echo "Enter filename:"
            read file
            touch "$file"
            echo "File '$file' created."
            ;;
        2)
            echo "Enter filename:"
            read file
            echo "Enter content:"
            read content
            echo "$content" > "$file"
            echo "Content written to '$file'."
            ;;
        3)
            echo "Enter filename:"
            read file
            echo "Enter content to append:"
            read content
            echo "$content" >> "$file"
            echo "Content appended to '$file'."
            ;;
        4)
            echo "Enter filename:"
            read file
            > "$file"
```

```
STDIN
1
practical.pdf
2

Output:

----------------------
FILE HANDLING MENU
----------------------
1. Create file
2. Write content to file
3. Append content to file
4. Delete file content
5. Exit
Enter choice:
Enter filename:
File 'practical.pdf' created.
----------------------
FILE HANDLING MENU
----------------------
1. Create file
2. Write content to file
3. Append content to file
4. Delete file content
5. Exit
Enter choice:
Enter filename:
Enter content:
Content written to 'practical.pdf'.
----------------------
FILE HANDLING MENU
----------------------
1. Create file
2. Write content to file
```


```bash
#!/bin/bash

while true
do
    echo "----------------------"
    echo "FILE HANDLING MENU"
    echo "----------------------"
    echo "1. Create file"
    echo "2. Write content to file"
    echo "3. Append content to file"
    echo "4. Delete file content"
    echo "5. Exit"
    echo "Enter choice:"
    read choice

    case $choice in
        1)
            echo "Enter filename:"
            read file
            touch "$file"
            echo "File '$file' created."
            ;;
        2)
            echo "Enter filename:"
            read file
            echo "Enter content:"
            read content
            echo "$content" > "$file"
            echo "Content written to '$file'."
            ;;
        3)
            echo "Enter filename:"
            read file
            echo "Enter content to append:"
            read content
            echo "$content" >> "$file"
            echo "Content appended to '$file'."
            ;;
        4)
            echo "Enter filename:"
            read file
            > "$file"
```


```
STDIN
2
practical.pdf
hello world

Output:

----------------------
FILE HANDLING MENU
----------------------
1. Create file
2. Write content to file
3. Append content to file
4. Delete file content
5. Exit
Enter choice:
Enter filename:
File 'practical.pdf' created.
----------------------
FILE HANDLING MENU
----------------------
1. Create file
2. Write content to file
3. Append content to file
4. Delete file content
5. Exit
Enter choice:
Enter filename:
Enter content:
Content written to 'practical.pdf'.
----------------------
FILE HANDLING MENU
----------------------
1. Create file
2. Write content to file
```

```
    echo "5. Exit"
    echo "Enter choice:"
    read choice

    case $choice in
        1)
            echo "Enter filename:"
            read file
            touch "$file"
            echo "File '$file' created."
            ;;
        2)
            echo "Enter filename:"
            read file
            echo "Enter content:"
            read content
            echo "$content" > "$file"
            echo "Content written to '$file'."
            ;;
        3)
            echo "Enter filename:"
            read file
            echo "Enter content to append:"
            read content
            echo "$content" >> "$file"
            echo "Content appended to '$file'."
            ;;
        4)
            echo "Enter filename:"
            read file
            > "$file"
            echo "Content of '$file' deleted."
            ;;
        5)
            echo "Exiting..."
            exit
            ;;
        *)
            echo "Invalid choice"
            ;;
    esac
done
```

STDIN

```
3
practical.pdf
this is append
```

```
1. Create file
2. Write content to file
3. Append content to file
4. Delete file content
5. Exit
Enter choice:
Enter filename:
Enter content:
Content written to 'practical.pdf'.
----------------------
FILE HANDLING MENU
----------------------
1. Create file
2. Write content to file
3. Append content to file
4. Delete file content
5. Exit
Enter choice:
Enter filename:
Enter content to append:
Content appended to 'practical.pdf'.
----------------------
FILE HANDLING MENU
----------------------
1. Create file
2. Write content to file
3. Append content to file
4. Delete file content
5. Exit
```

```
    echo "5. Exit"
    echo "Enter choice:"
    read choice

    case $choice in
        1)
            echo "Enter filename:"
            read file
            touch "$file"
            echo "File '$file' created."
            ;;
        2)
            echo "Enter filename:"
            read file
            echo "Enter content:"
            read content
            echo "$content" > "$file"
            echo "Content written to '$file'."
            ;;
        3)
            echo "Enter filename:"
            read file
            echo "Enter content to append:"
            read content
            echo "$content" >> "$file"
            echo "Content appended to '$file'."
            ;;
        4)
            echo "Enter filename:"
            read file
            > "$file"
            echo "Content of '$file' deleted."
            ;;
        5)
            echo "Exiting..."
            exit
            ;;
        *)
            echo "Invalid choice"
            ;;
    esac
done
```

STDIN

```
4
practical.pdf
Content of 'practical.pdf' deleted.
```

```
4. Delete file content
5. Exit
Enter choice:
Enter filename:
Enter content:
Content written to 'practical.pdf'.
----------------------
FILE HANDLING MENU
----------------------
1. Create file
2. Write content to file
3. Append content to file
4. Delete file content
5. Exit
Enter choice:
Enter filename:
Enter content to append:
Content appended to 'practical.pdf'.
----------------------
FILE HANDLING MENU
----------------------
1. Create file
2. Write content to file
3. Append content to file
4. Delete file content
5. Exit
Enter choice:
Enter filename:
Content of 'practical.pdf' deleted.
```