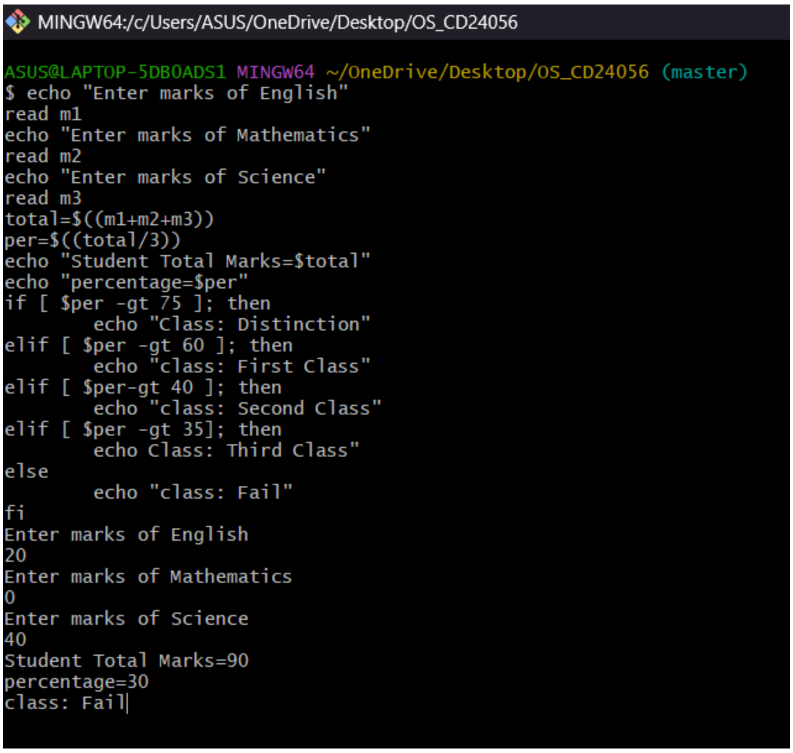
a] Write a shell script to generate mark-sheet of a student. Take 3 subjects, calculate and display total marks, percentage and Class obtained by the student.

Code:

```
echo "Enter marks of English"
read m1
echo "Enter marks of Mathematics"
read m2
echo "Enter marks of Science"
read m3
total=$((m1+m2+m3))
per=$((total/3))
echo "Student Total Marks=$total"
echo "percentage=$per"
if [ $per -gt 75 ]; then
        echo "Class: Distinction"
elif [ $per -gt 60 ]; then
        echo "class: First Class"
elif [ $per-gt 40 ]; then
        echo "class: Second Class"
elif [ $per -gt 35]; then
        echo Class: Third Class"
else
        echo "class: Fail"
fi
```

OUTPUT:



```
ASUS@LAPTOP-5DB0ADS1 MINGW64 ~/OneDrive/Desktop/OS_CD24056 (master)
$ echo "Enter marks of English"
read m1
echo "Enter marks of Mathematics"
read m2
echo "Enter marks of Science"
read m3
total=$((m1+m2+m3))
per=$((total/3))
echo "Student Total Marks=$total"
echo "percentage=$per"
if [ $per -gt 75 ]; then
        echo "Class: Distinction"
elif [ $per -gt 60 ]; then
        echo "class: First Class"
elif [ $per-gt 40 ]; then
        echo "class: Second Class"
elif [ $per -gt 35]; then
        echo Class: Third Class"
else
        echo "class: Fail"
fi
Enter marks of English
20
Enter marks of Mathematics
0
Enter marks of Science
40
Student Total Marks=90
percentage=30
class: Fail
```

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

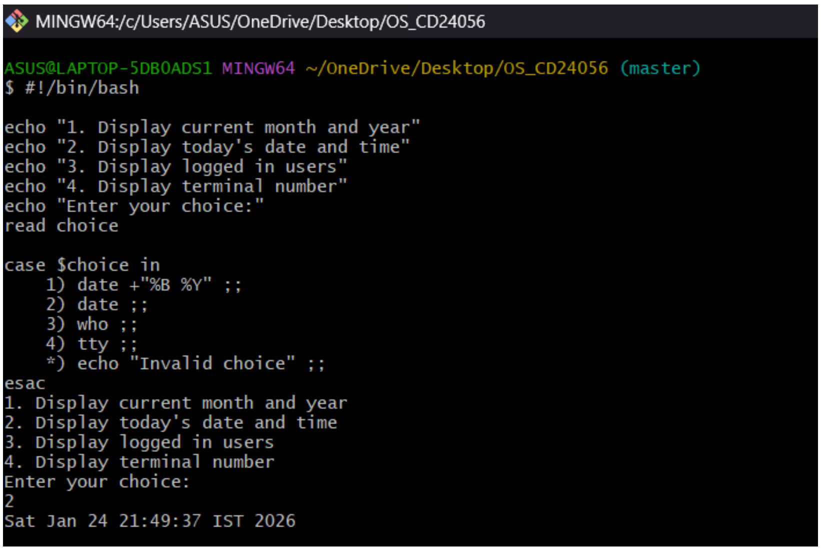
- Display calendar of current month.
- Display today’s date and time.
- Display usernames those are currently logged in the system.
- Display your terminal number.

CODE:

```
#!/bin/bash


echo "1. Display current month and year"

echo "2. Display today's date and time"

echo "3. Display logged in users"

echo "4. Display terminal number"

echo "Enter your choice:"

read choice
```

```
case $choice in
   1) date +"%B %Y" ;;
   2) date ;;
   3) who ;;
   4) tty ;;
   *) echo "Invalid choice" ;;
esac
```

OUTPUT:



```
ASUS@LAPTOP-5DB0ADS1 MINGW64 ~/OneDrive/Desktop/OS_CD24056 (master)
$ #!/bin/bash

echo "1. Display current month and year"
echo "2. Display today's date and time"
echo "3. Display logged in users"
echo "4. Display terminal number"
echo "Enter your choice:"
read choice

case $choice in
    1) date +"%B %Y" ;;
    2) date ;;
    3) who ;;
    4) tty ;;
    *) echo "Invalid choice" ;;
esac
1. Display current month and year
2. Display today's date and time
3. Display logged in users
4. Display terminal number
Enter your choice:
2
Sat Jan 24 21:49:37 IST 2026
```

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

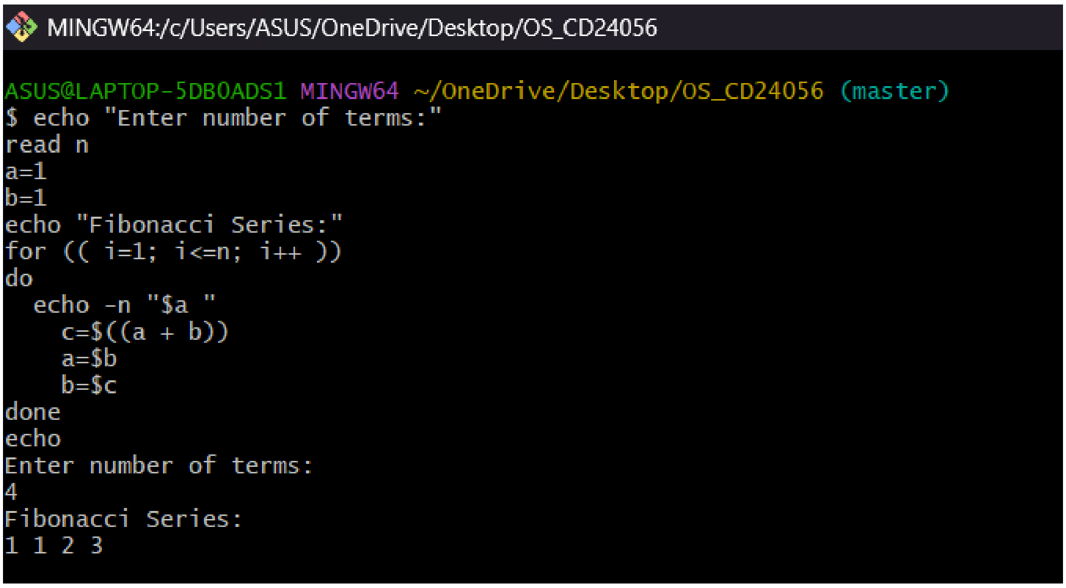
CODE:

```
echo "Enter number of terms:"
read n
a=1
b=1
echo "Fibonacci Series:"
for (( i=1; i<=n; i++ ))
do
 echo -n "$a "
   c=$((a + b))
   a=$b
   b=$c
```

done

echo

OUTPUT:



MINGW64:/c/Users/ASUS/OneDrive/Desktop/OS_CD24056

```
ASUS@LAPTOP-5DB0ADS1 MINGW64 ~/OneDrive/Desktop/OS_CD24056 (master)
$ echo "Enter number of terms:"
read n
a=1
b=1
echo "Fibonacci Series:"
for (( i=1; i<=n; i++ ))
do
  echo -n "$a "
    c=$((a + b))
    a=$b
    b=$c
done
echo
Enter number of terms:
4
Fibonacci Series:
1 1 2 3
```

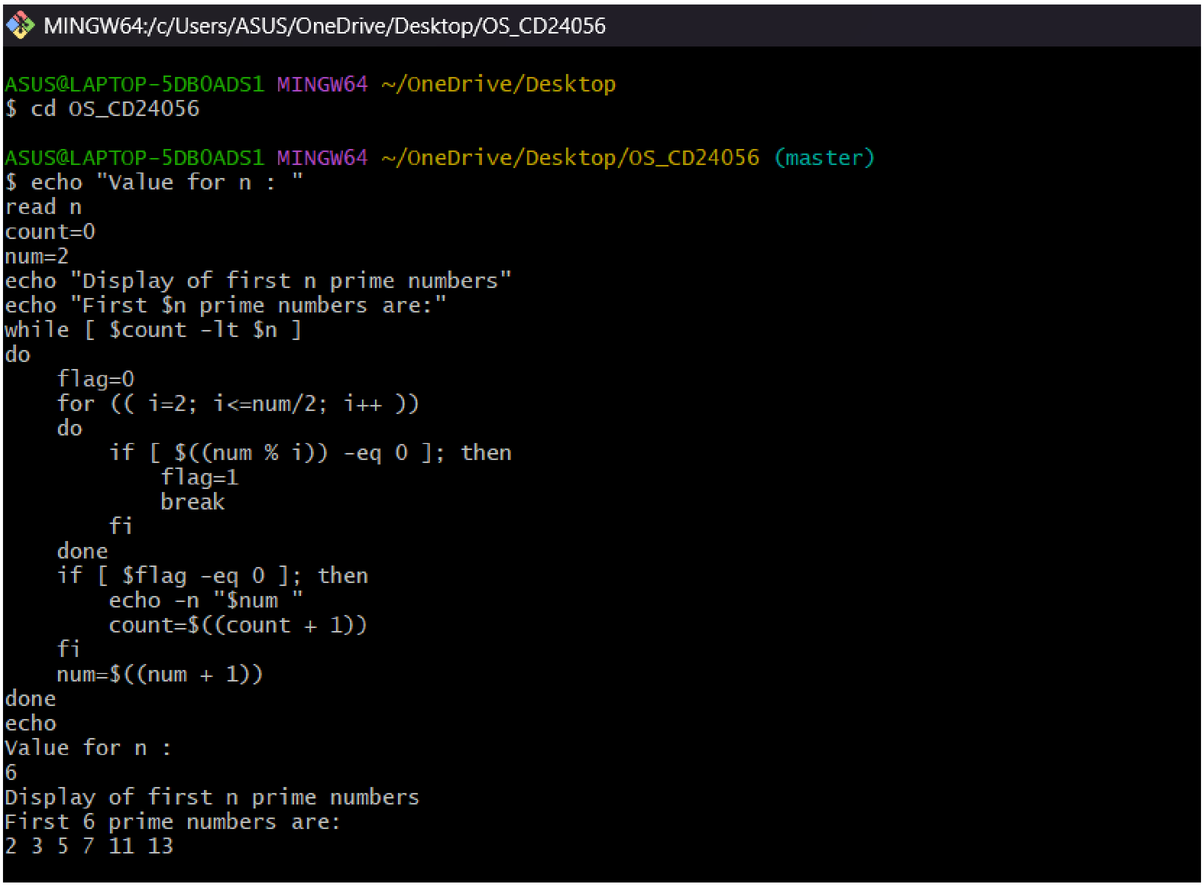
D] Write a shell script which will accept a number *n* and display first *n* prime numbers as output.

CODE:

```
echo "Value for n : "

read n

count=0

num=2

echo "Display of first n prime numbers"

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
```

OUTPUT:



```
MINGW64:/c/Users/ASUS/OneDrive/Desktop/OS_CD24056

ASUS@LAPTOP-5DB0ADS1 MINGW64 ~/OneDrive/Desktop
$ cd OS_CD24056

ASUS@LAPTOP-5DB0ADS1 MINGW64 ~/OneDrive/Desktop/OS_CD24056 (master)
$ echo "Value for n : "
read n
count=0
num=2
echo "Display of first n prime numbers"
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
Value for n :
6
Display of first n prime numbers
First 6 prime numbers are:
2 3 5 7 11 13
```

e) Write menu driven program for file handling activity

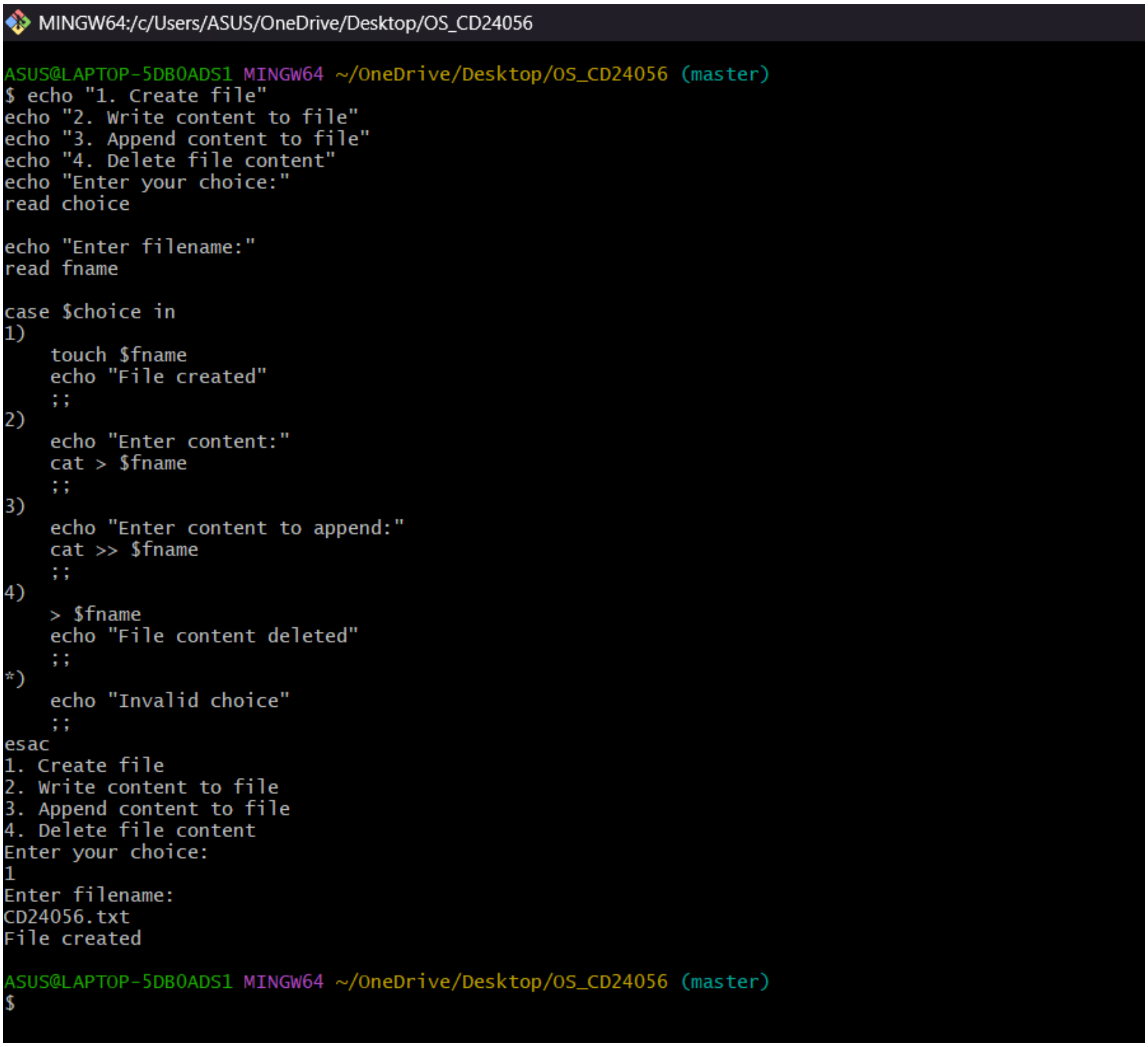
- Creation of file.
- Write content in the file.
- Upend file content.
- Delete file content.

CODE:

```
 echo "1. Create file"
echo "2. Write content to file"
echo "3. Append content to file"
echo "4. Delete file content"
echo "Enter your choice:"
read choice

echo "Enter filename:"
read fname

case $choice in
1)
   touch $fname
   echo "File created"
   ;;
2)
   echo "Enter content:"
   cat > $fname
   ;;
3)
   echo "Enter content to append:"
   cat >> $fname
   ;;
4)
   > $fname
```

```
    echo "File content deleted"
    ;;
*)
    echo "Invalid choice"
    ;;
esac
esac
```

OUTPUT:



MINGW64:/c/Users/ASUS/OneDrive/Desktop/OS_CD24056

```
ASUS@LAPTOP-5DB0ADS1 MINGW64 ~/OneDrive/Desktop/OS_CD24056 (master)
$ echo "1. Create file"
echo "2. Write content to file"
echo "3. Append content to file"
echo "4. Delete file content"
echo "Enter your choice:"
read choice

echo "Enter filename:"
read fname

case $choice in
1)
    touch $fname
    echo "File created"
    ;;
2)
    echo "Enter content:"
    cat > $fname
    ;;
3)
    echo "Enter content to append:"
    cat >> $fname
    ;;
4)
    > $fname
    echo "File content deleted"
    ;;
*)
    echo "Invalid choice"
    ;;
esac
1. Create file
2. Write content to file
3. Append content to file
4. Delete file content
Enter your choice:
1
Enter filename:
CD24056.txt
File created

ASUS@LAPTOP-5DB0ADS1 MINGW64 ~/OneDrive/Desktop/OS_CD24056 (master)
$
```