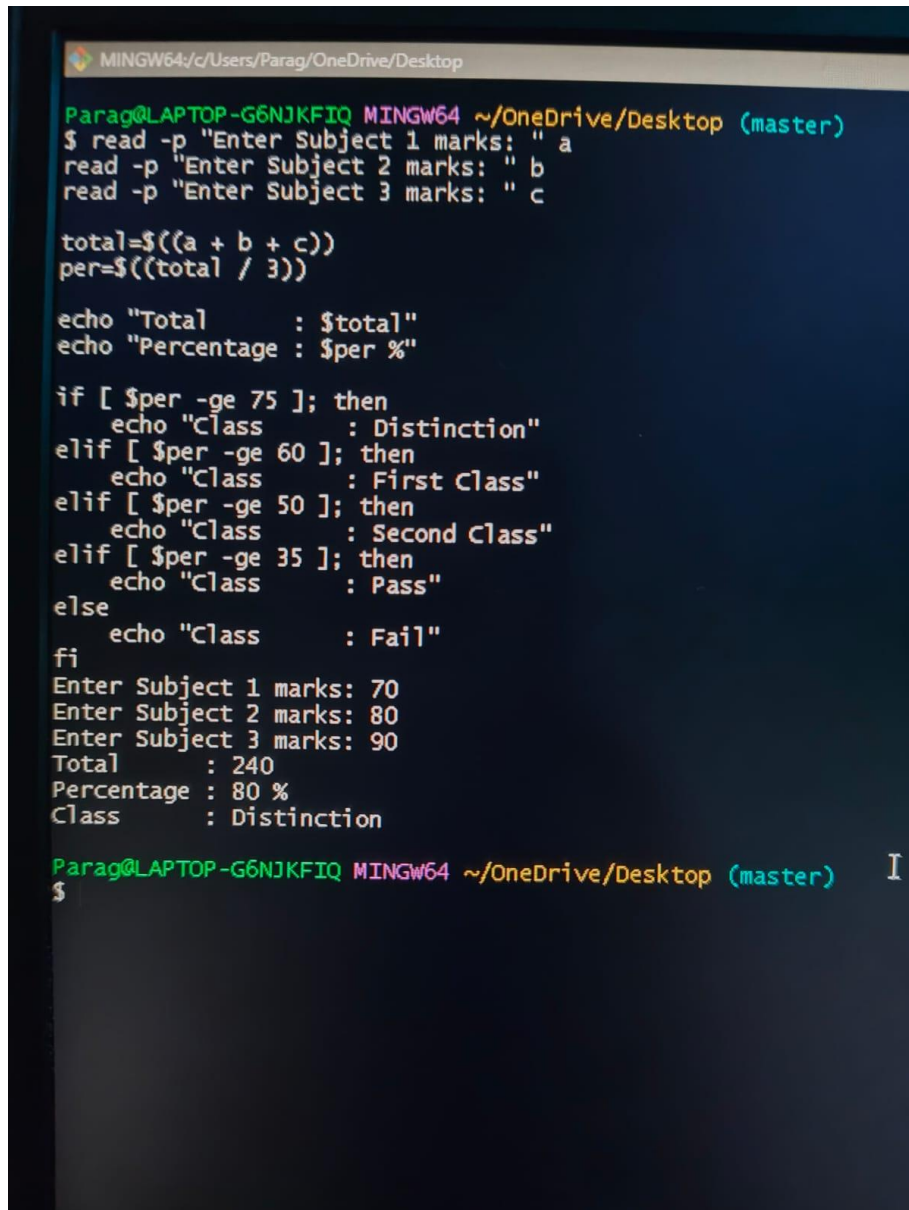


PracticalNo : 2

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/Parag/OneDrive/Desktop
Parag@LAPTOP-G6NJKFIQ MINGW64 ~/OneDrive/Desktop (master)
$ read -p "Enter Subject 1 marks: " a
read -p "Enter Subject 2 marks: " b
read -p "Enter Subject 3 marks: " c

total=$((a + b + c))
per=$((total / 3))

echo "Total      : $total"
echo "Percentage : $per %"

if [ $per -ge 75 ]; then
    echo "Class      : Distinction"
elif [ $per -ge 60 ]; then
    echo "Class      : First Class"
elif [ $per -ge 50 ]; then
    echo "Class      : Second Class"
elif [ $per -ge 35 ]; then
    echo "Class      : Pass"
else
    echo "Class      : Fail"
fi
Enter Subject 1 marks: 70
Enter Subject 2 marks: 80
Enter Subject 3 marks: 90
Total      : 240
Percentage : 80 %
Class      : Distinction

Parag@LAPTOP-G6NJKFIQ MINGW64 ~/OneDrive/Desktop (master) I
$
```

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

```
MINGW64/c/Users/Parag/OneDrive/Desktop
Parag@LAPTOP-G6NJKFIQ MINGW64 ~/OneDrive/Desktop (master)
$ 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 11:35:03 IST 2026

Parag@LAPTOP-G6NJKFIQ MINGW64 ~/OneDrive/Desktop (master)
$
```

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

```
MINGW64/c/Users/Parag/OneDrive/Desktop
Parag@LAPTOP-G6NJKFIQ MINGW64 ~/OneDrive/Desktop (master)
$ echo "Enter the value of n:"
read n

a=1
b=1

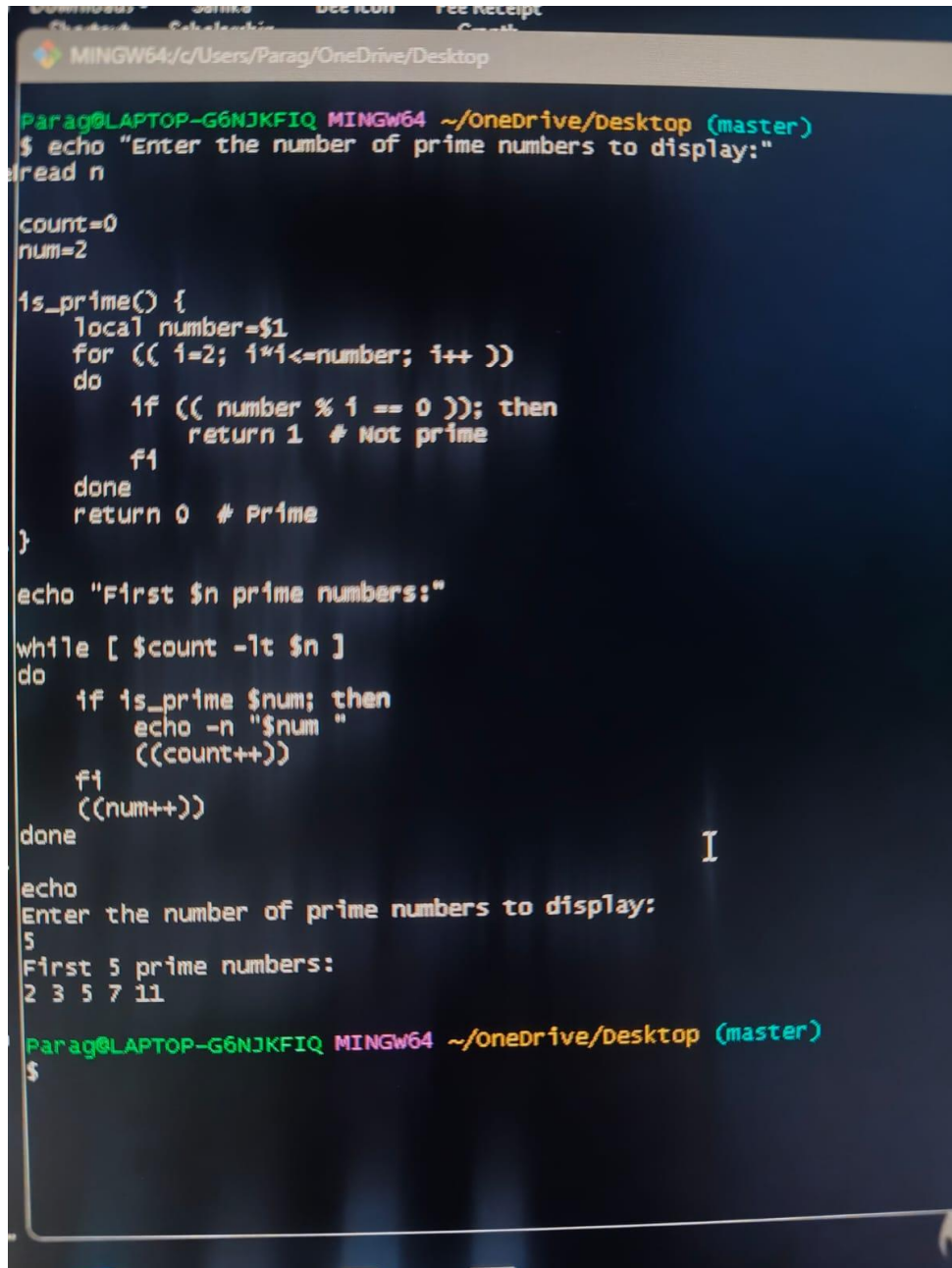
echo "Fibonacci series:"
echo -n "$a $b "

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

echo
Enter the value of n:
5
Fibonacci series:
1 1 2 3 5

Parag@LAPTOP-G6NJKFIQ MINGW64 ~/OneDrive/Desktop (master)
$
```

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



```
MINGW64/c/Users/Parag/OneDrive/Desktop
Parag@LAPTOP-G6NJKFIQ MINGW64 ~/OneDrive/Desktop (master)
$ 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:
5
First 5 prime numbers:
2 3 5 7 11

Parag@LAPTOP-G6NJKFIQ MINGW64 ~/OneDrive/Desktop (master)
$
```

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

```
MINGW64/c/Users/Parag/OneDrive/Desktop
Parag@LAPTOP-G6N3KFIQ MINGW64 ~/OneDrive/Desktop (master)
$ 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: myfile.txt
File 'myfile.txt' created.

Parag@LAPTOP-G6N3KFIQ MINGW64 ~/OneDrive/Desktop (master)
$
```