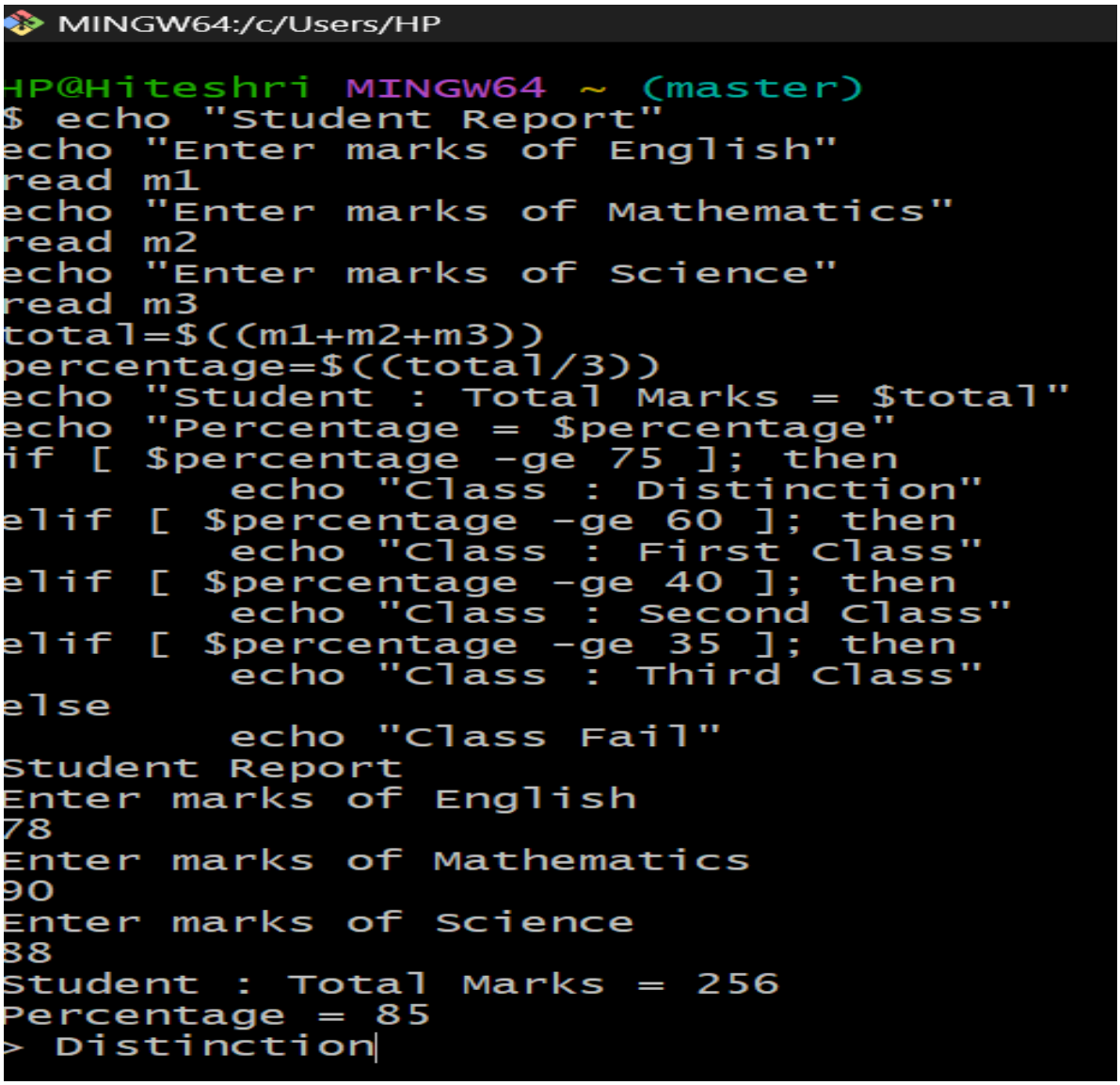
1. Write a shell script to generate mark- sheet of a student. Take 3 subjects, calculate and display total marks, percentage and Class obtained by the student.



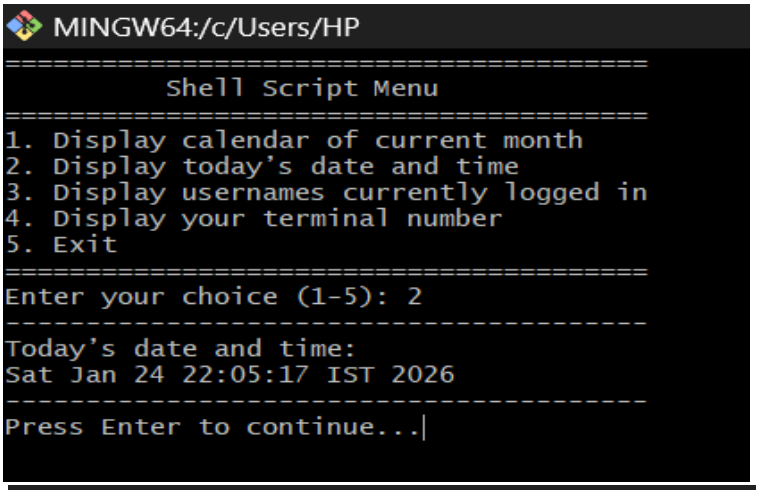
2. Write a menu driven shell script which will print the following menu and execute the given task.
   - Display calendar of current month
   - Display today's date and time
   - Display usernames those are currently logged in the system
   - Display your terminal number

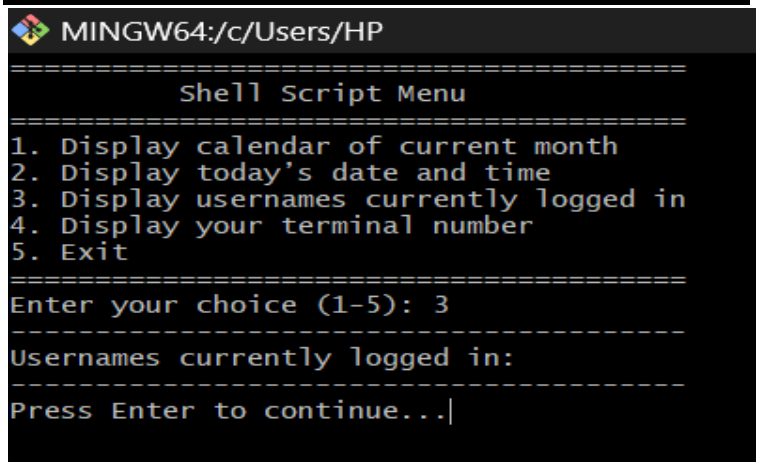
MINGW64:/c/Users/HP

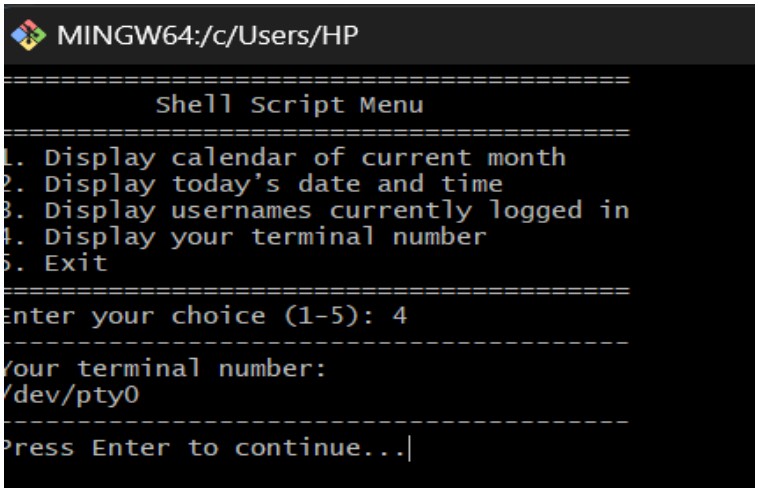
```
HP@Hiteshri MINGW64 ~ (master)
$ display_menu() {
    clear
    echo "========================================"
    echo "          Shell Script Menu"
    echo "========================================"
    echo "1. Display calendar of current month"
    echo "2. Display today's date and time"
    echo "3. Display usernames currently logged in"
    echo "4. Display your terminal number"
    echo "5. Exit"
    echo "========================================"
    echo -n "Enter your choice (1-5): "
}
while true; do
    display_menu
    read choice

    case $choice in
        1)
            echo "----------------------------------------"
            echo "Calendar of current month:"
            cal
            echo "----------------------------------------"
            read -p "Press Enter to continue..."
            ;;
        2)
            echo "----------------------------------------"
            echo "Today's date and time:"
            date
            echo "----------------------------------------"
            read -p "Press Enter to continue..."
            ;;
        3)
            echo "----------------------------------------"
            echo "Usernames currently logged in:"
            who | awk '{print $1}' | sort | uniq
            echo "----------------------------------------"
            read -p "Press Enter to continue..."
            ;;
        4)
            echo "----------------------------------------"
            echo "Your terminal number:"
            tty
            echo "----------------------------------------"
            read -p "Press Enter to continue..."
            ;;
        5)
            echo "Exiting script. Goodbye!"
            exit 0
            ;;
        *)
            echo "Invalid choice. Please enter a number between 1 and 5."
            read -p "Press Enter to continue..."
            ;;
    esac
done
```

MINGW64:/c/Users/HP

```
========================================
          Shell Script 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 (1-5): 1
----------------------------------------
Calendar of current month:
bash: cal: command not found
----------------------------------------
Press Enter to continue...
```



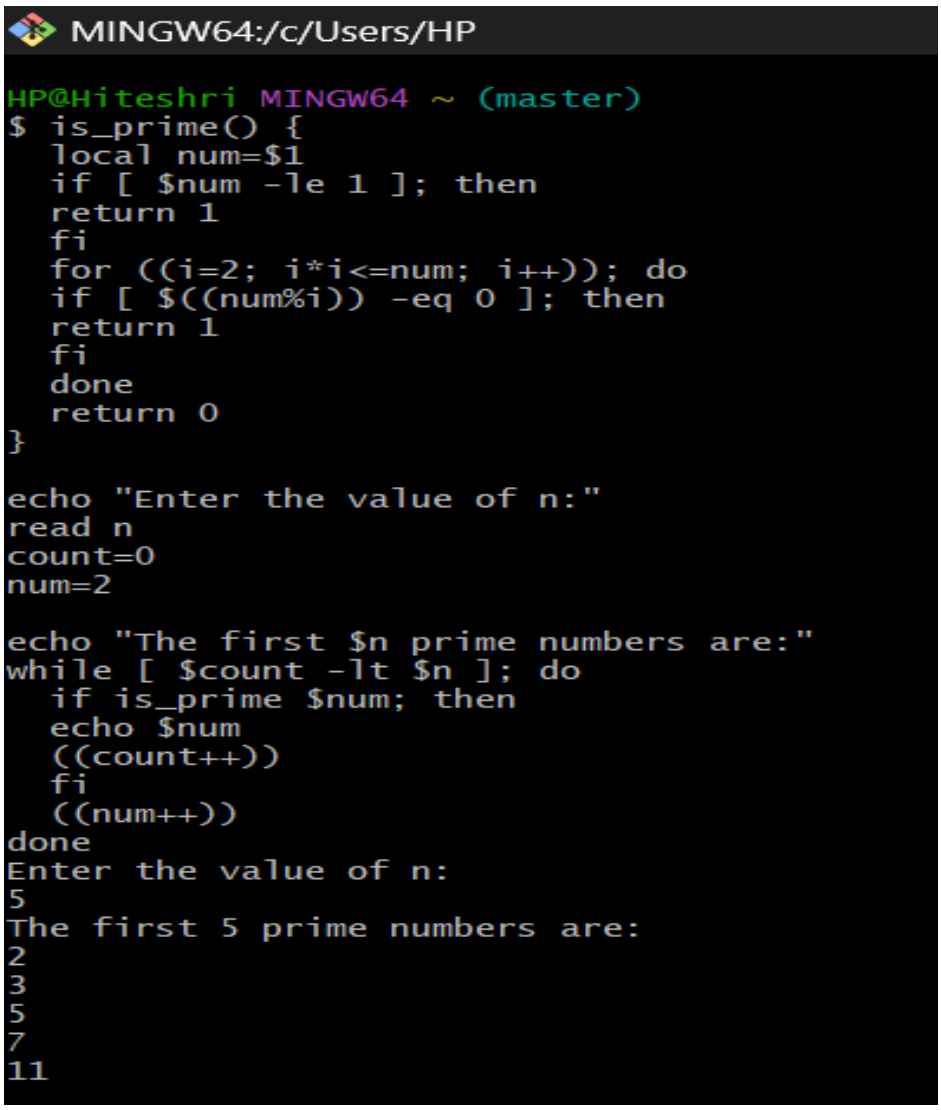




3. Write a shell script which will generate first n fibonnacci numbers like: 1, 1, 2, 3, 5, 8, 13.

```
HP@Hiteshri MINGW64 ~ (master)
$ N=7
a=1
b=1

echo "The Fibonacci series is : "

for (( i=0; i<N; i++ ))
do
    echo -n "$a "
    fn=$((a + b))
    a=$b
    b=$fn
done
The Fibonacci series is :
1 1 2 3 5 8 13
```

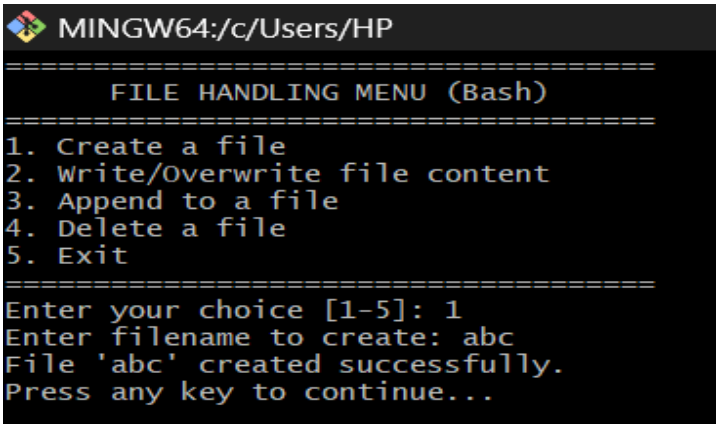
4. Write a shell script which will accept a number b and display first n prime numbers as output.

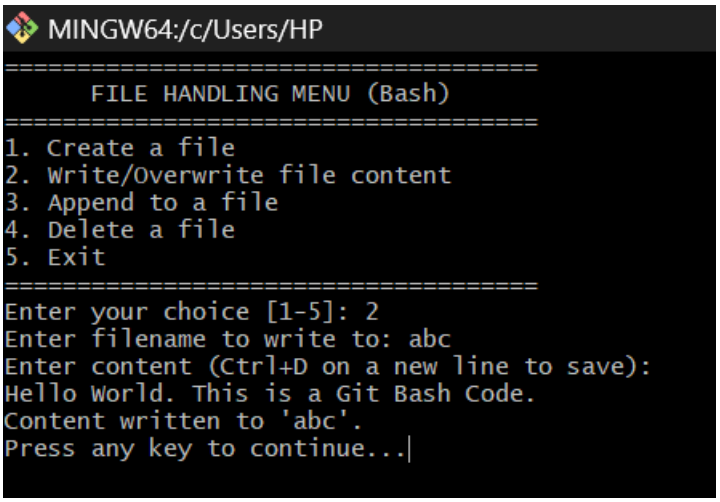


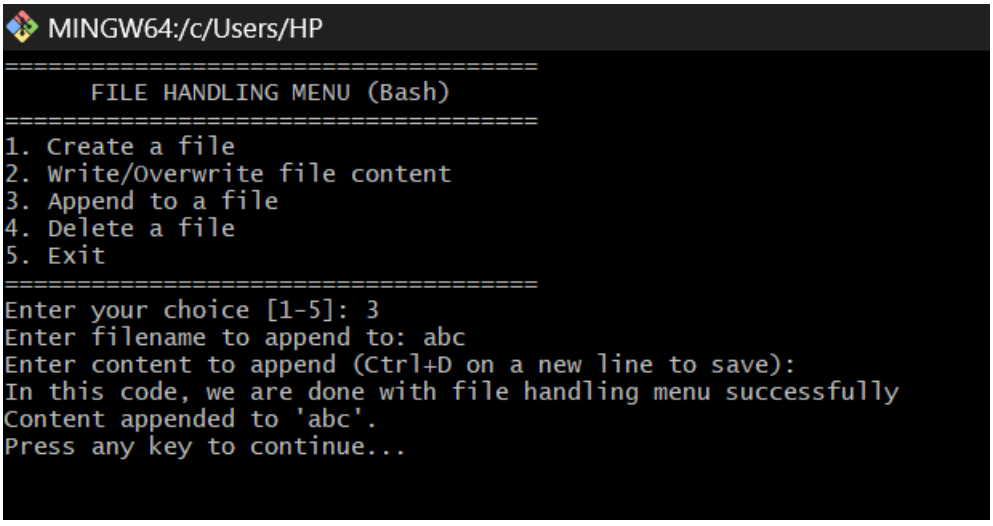
5. Write menu driven program for file handling activity
   - Creation of file
   - Write content in the file
   - Upend file content
   - Delete file content

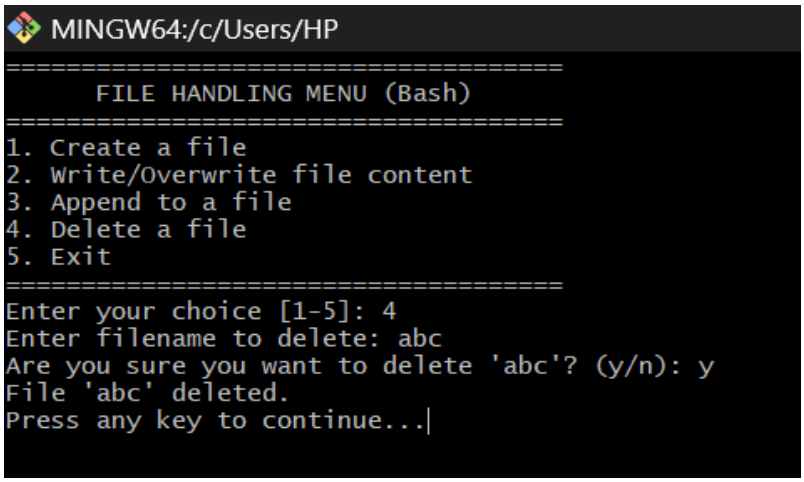
MINGW64:/c/Users/HP

```
$ clear_screen() {
    printf "\033c"
}
create_file() {
$ clear_screen() {
    printf "\033c"
}
create_file() {
    read -p "Enter filename to create: " filename
    if [ -e "$filename" ]; then
        echo "Error: File '$filename' already exists."
    else
        touch "$filename"
        echo "File '$filename' created successfully."
    fi
}
write_file() {
$ clear_screen() {
    printf "\033c"
}
create_file() {
    read -p "Enter filename to create: " filename
    if [ -e "$filename" ]; then
$ clear_screen() {
    printf "\033c"
}
create_file() {
    read -p "Enter filename to create: " filename
    if [ -e "$filename" ]; then
        echo "Error: File '$filename' already exists."
    else
        touch "$filename"
        echo "File '$filename' created successfully."
    fi
}
write_file() {
    read -p "Enter filename to write to: " filename
    if [ -f "$filename" ]; then
        echo "Enter content (Ctrl+D on a new line to save):"
        cat > "$filename"
        echo "Content written to '$filename'."
    else
        echo "Error: File '$filename' not found. Use option 1 to create it first."
    fi
}
append_file() {
    read -p "Enter filename to append to: " filename
    if [ -f "$filename" ]; then
        echo "Enter content to append (Ctrl+D on a new line to save):"
        cat >> "$filename"
        echo "Content appended to '$filename'."
    else
        echo "Error: File '$filename' not found."
    fi
}
delete_file() {
    read -p "Enter filename to delete: " filename
    if [ -f "$filename" ]; then
        read -p "Are you sure you want to delete '$filename'? (y/n): " confirm
        if [[ "$confirm" == [yY] || "$confirm" == [yY][eE][sS] ]]; then
            rm "$filename"
            echo "File '$filename' deleted."
        else
            echo "Deletion cancelled."
        fi
    else
        echo "Error: File '$filename' not found."
    fi
}
while true; do
    clear_screen
    echo "----------------------------------------"
    echo "       FILE HANDLING MENU (Bash)"
    echo "----------------------------------------"
    echo "1. Create a file"
    echo "2. Write/Overwrite file content"
    echo "3. Append to a file"
    echo "4. Delete a file"
    echo "5. Exit"
    echo "----------------------------------------"
    read -p "Enter your choice [1-5]: " choice

    case "$choice" in
        1)
            create_file
            read -n 1 -s -r -p "Press any key to continue..."
            ;;
        2)
            write_file
            read -n 1 -s -r -p "Press any key to continue..."
            ;;
        3)
            append_file
            read -n 1 -s -r -p "Press any key to continue..."
            ;;
        4)
            delete_file
            read -n 1 -s -r -p "Press any key to continue..."
            ;;
        5)
            echo "Exiting script. Goodbye!"
            exit 0
            ;;
        *)
            echo "Invalid choice. Please enter a number between 1 and 5."
            read -n 1 -s -r -p "Press any key to continue..."
            ;;
    esac
done
```


MINGW64:/c/Users/HP
FILE HANDLING MENU (Bash)
1. Create a file
2. Write/Overwrite file content
3. Append to a file
4. Delete a file
5. Exit
Enter your choice [1-5]: 1
Enter filename to create: abc
File 'abc' created successfully.
Press any key to continue...


MINGW64:/c/Users/HP
FILE HANDLING MENU (Bash)
1. Create a file
2. Write/Overwrite file content
3. Append to a file
4. Delete a file
5. Exit
Enter your choice [1-5]: 2
Enter filename to write to: abc
Enter content (Ctrl+D on a new line to save):
Hello World. This is a Git Bash Code.
Content written to 'abc'.
Press any key to continue...


MINGW64:/c/Users/HP
FILE HANDLING MENU (Bash)
1. Create a file
2. Write/Overwrite file content
3. Append to a file
4. Delete a file
5. Exit
Enter your choice [1-5]: 3
Enter filename to append to: abc
Enter content to append (Ctrl+D on a new line to save):
In this code, we are done with file handling menu successfully
Content appended to 'abc'.
Press any key to continue...


MINGW64:/c/Users/HP
FILE HANDLING MENU (Bash)
1. Create a file
2. Write/Overwrite file content
3. Append to a file
4. Delete a file
5. Exit
Enter your choice [1-5]: 4
Enter filename to delete: abc
Are you sure you want to delete 'abc'? (y/n): y
File 'abc' deleted.
Press any key to continue...