Assignment 2

1. Create two C files to print "**Hello World!**" in two different ways:
   a. Program containing normal statement terminator » HelloWorld1.c.
   b. Program without any statement terminator » HelloWorld2.c.

      Input :

             touch HelloWorld1.c; touch HelloWorld2.c

             gcc HelloWorld1.c –o HelloWorld1; gcc HelloWorld2.c –o HelloWorld2

             ./HelloWorld1; ./HelloWorld2

      Output :

```
[bhubonmondal@Bhubons-Mac-mini lc % touch HelloWorld1.c; touch HelloWorld2.c
```

```
lc — nano HelloWorld1.c — 43×24
 File: HelloWorld1.c              Modified

#include <stdio.h>
int main(){
   printf("Hello World!");
}
```

```
lc — nano HelloWorld2.c — 38×24
 File: HelloWorld2.c        Modified

#include <stdio.h>
int main(){
   if(printf("Hello World!")){}
}
```

```
bhubonmondal@Bhubons-Mac-mini lc % gcc HelloWorld1.c -o HelloWorld1; gcc HelloWorld2.
c -o HelloWorld2;
bhubonmondal@Bhubons-Mac-mini lc % ./HelloWorld1; ./HelloWorld2
Hello World!Hello World!%
```

<u>Write the processes for the following and paste the screenshot of the output:</u>

2. Display the contents of the files.

   Input : cat HelloWorld1.c; cat HelloWorld2.c

   Output :

```
bhubonmondal@Bhubons-Mac-mini lc % cat HelloWorld1.c; cat HelloWorld2.c
#include <stdio.h>
int main(){
   printf("Hello World!");
}
#include <stdio.h>
int main(){
   if(printf("Hello World!")){}
}
```

3. Concatenate the two files into a third file.

   Input : cat HelloWorld1.c HelloWorld2.c > HelloWorld3.c

   Output:

```
bhubonmondal@Bhubons-Mac-mini lc % cat HelloWorld1.c HelloWorld2.c > HelloWorld3.c
bhubonmondal@Bhubons-Mac-mini lc %
```

4. Compare the two C files.

   Input : cmp HelloWorld1.c HelloWorld2.c

Output:

```
bhubonmondal@Bhubons-Mac-mini lc % cmp HelloWorld1.c HelloWorld2.c
HelloWorld1.c HelloWorld2.c differ: char 35, line 3
```

5. Find what is common in two C files.

   Input : comm HelloWorld1.c HelloWorld2.c

   Output :

```
bhubonmondal@Bhubons-Mac-mini lc % comm HelloWorld1.c HelloWorld2.c
            #include <stdio.h>
            int main(){
        if(printf("Hello World!")){}
    printf("Hello World!");
            }
```

6. Find the difference in two C files.

   Input : diff HelloWorld1.c HelloWorld2.c

   Output :

```
bhubonmondal@Bhubons-Mac-mini lc % diff HelloWorld1.c HelloWorld2.c
3c3
<     printf("Hello World!");
---
>     if(printf("Hello World!")){}
```

7. Show the above file types.

   Input: file HelloWorld1.c; file HelloWorld2.c

   Output:

```
bhubonmondal@Bhubons-Mac-mini lc % file HelloWorld1.c; file HelloWorld2.c
HelloWorld1.c: c program text, ASCII text
HelloWorld2.c: c program text, ASCII text
```

8. Copy all the files to the home directory interactively.

   Input: cp *.c ~

   Output:

```
bhubonmondal@Bhubons-Mac-mini lc % cp *.c ~
```

9. Create a copy of the C file in `TestA-1`.

   Input : cp *.c mca2024/Bhubon_B_15/TestA/TestA-1

   Output :

```
bhubonmondal@Bhubons-Mac-mini ~ % cp *.c mca2024/Bhubon_B_15/TestA/TestA-1
bhubonmondal@Bhubons-Mac-mini ~ %
```

10. Copy the file to the home directory interactively.

    Input : cp –i *.c ~

    Output :

```
bhubonmondal@Bhubons-Mac-mini lc % cp -i *.c ~
```

11. Remove the directories `TestC & TestC-1`.

    Input : rmdir mca2024/Bhubon_B_15/TestC/TestC-1

    Output :

```
bhubonmondal@Bhubons-Mac-mini ~ % rmdir mca2024/Bhubon_B_15/TestC/TestC-1
```

12. Delete the file C file from TestA-1.

Input: rm –rf mca2024/Bhubon_B_15/TestA/TestA-1

Output:

```
[bhubonmondal@Bhubons-Mac-mini ~ % rm -rf mca2024/Bhubon_B_15/TestA/TestA-1
```

13. Rename the text file in the home directory.

    Input: mv HelloWorld1.c HelloWorld3.c

    Output:

```
[bhubonmondal@Bhubons-Mac-mini ~ % mv HelloWorld1.c HelloWorld3.c
```

Shell Script

14. Create a Shell Script for a menu-driven basic arithmetic calculator (Note: you must ensure that bc is installed in the UNIX system).

    Input:

```
while true
do
    echo "1. Addition, 2. Subtraction, 3. Multiplication, 4. Division, 5. Exit"
    read -p "Enter your choice [1-5]: " choice

    if [ "$choice" -eq 1 ]; then
        read -p "Enter two numbers: " a b
        echo "$a + $b" | bc
    elif [ "$choice" -eq 2 ]; then
        read -p "Enter two numbers: " a b
        echo "$a - $b" | bc
    elif [ "$choice" -eq 3 ]; then
        read -p "Enter two numbers: " a b
        echo "$a * $b" | bc
    elif [ "$choice" -eq 4 ]; then
        read -p "Enter two numbers: " a b
        echo "scale=2; $a / $b" | bc
    elif [ "$choice" -eq 5 ]; then
        echo "Exiting..."
        exit 0
    else
        echo "Invalid choice, try again."
    fi
done
```

    Output:

```
[bhubonmondal@Bhubons-Mac-mini lc % nano calc.sh
[bhubonmondal@Bhubons-Mac-mini lc % chmod +x calc.sh
[bhubonmondal@Bhubons-Mac-mini lc % ./calc.sh
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Exit
Enter your choice [1-5]: 1
Enter two numbers: 5 6
11
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Exit
Enter your choice [1-5]: 3
Enter two numbers: 5 5
25
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Exit
Enter your choice [1-5]: █
```

15. Write the process to display the shell script file in a paginated manner using the 'more' and 'less' commands.

Input : more calc.sh

less calc.sh

Output:

```
[bhubonmondal@Bhubons-Mac-mini lc % more calc.sh
#!/bin/bash
while true
do
    echo "1. Addition"
    echo "2. Subtraction"
    echo "3. Multiplication"
    echo "4. Division"
    echo "5. Exit"
    read -p "Enter your choice [1-5]: " choice

    if [ "$choice" -eq 1 ]; then
        read -p "Enter two numbers: " a b
        echo "$a + $b" | bc

    elif [ "$choice" -eq 2 ]; then
        read -p "Enter two numbers: " a b
        echo "$a - $b" | bc

    elif [ "$choice" -eq 3 ]; then
        read -p "Enter two numbers: " a b
        echo "$a * $b" | bc

    elif [ "$choice" -eq 4 ]; then
        read -p "Enter two numbers: " a b
        echo "scale=2; $a / $b" | bc

    elif [ "$choice" -eq 5 ]; then
        echo "Exiting..."
        exit 0
```

```
#!/bin/bash
while true
do
    echo "1. Addition"
    echo "2. Subtraction"
    echo "3. Multiplication"
    echo "4. Division"
    echo "5. Exit"
    read -p "Enter your choice [1-5]: " choice

    if [ "$choice" -eq 1 ]; then
        read -p "Enter two numbers: " a b
        echo "$a + $b" | bc

    elif [ "$choice" -eq 2 ]; then
        read -p "Enter two numbers: " a b
        echo "$a - $b" | bc

    elif [ "$choice" -eq 3 ]; then
        read -p "Enter two numbers: " a b
        echo "$a * $b" | bc

    elif [ "$choice" -eq 4 ]; then
        read -p "Enter two numbers: " a b
        echo "scale=2; $a / $b" | bc

    elif [ "$choice" -eq 5 ]; then
        echo "Exiting..."
        exit 0
calc.sh█
```

16. Write a Shell Script to count the number of lines, words and characters in the above file and store the details in a file named stats.txt along with the file name and absolute path of the file.

Input:

```
file="calc.sh"
abs_path=$(realpath "$file")

lines=$(wc -l < "$file")
```

```
words=$(wc -w < "$file")
chars=$(wc -m < "$file")

echo "File: $abs_path"
echo "Number of lines: $lines"
echo "Number of words: $words"
echo "Number of characters: $chars"

{
    echo "File: $abs_path"
    echo "Number of lines: $lines"
    echo "Number of words: $words"
    echo "Number of characters: $chars"
} > stats.txt
```

Output:

```
[bhubonmondal@Bhubons-Mac-mini lc % nano abc.sh
[bhubonmondal@Bhubons-Mac-mini lc % chmod +x abc.sh
[bhubonmondal@Bhubons-Mac-mini lc % ./abc.sh
File: /Users/bhubonmondal/Desktop/lc/lc/calc.sh
Number of lines:        35
Number of words:       131
Number of characters:       766
```