**Task 1**: Write a single shell script that creates four different files, while taking the names of all created files as input from the user. As the files contents, insert your name in the first file, registration number in the second and section details in the third. These should be followed by merging the contents of all three files into the fourth one.

**Code:-**

#!/bin/bash

echo "Enter Name of File1"

read file1

touch $file1.txt

echo "Enter name of file2"

read file2

touch $file2.txt

echo "Enter name of file3"

read file3

touch $file3.txt

echo "Enter name of file4"

read file4

touch $file4.txt

echo "Files created"

echo "Enter name your name in file 1"

read name

echo $name>>$file1.txt

echo "Enter reg no in file 2"

read reg

echo $reg>>$file2.txt

echo "Enter section in file 3"

read sec

echo $sec>>$file3.txt

cat $file1.txt>>$file4.txt

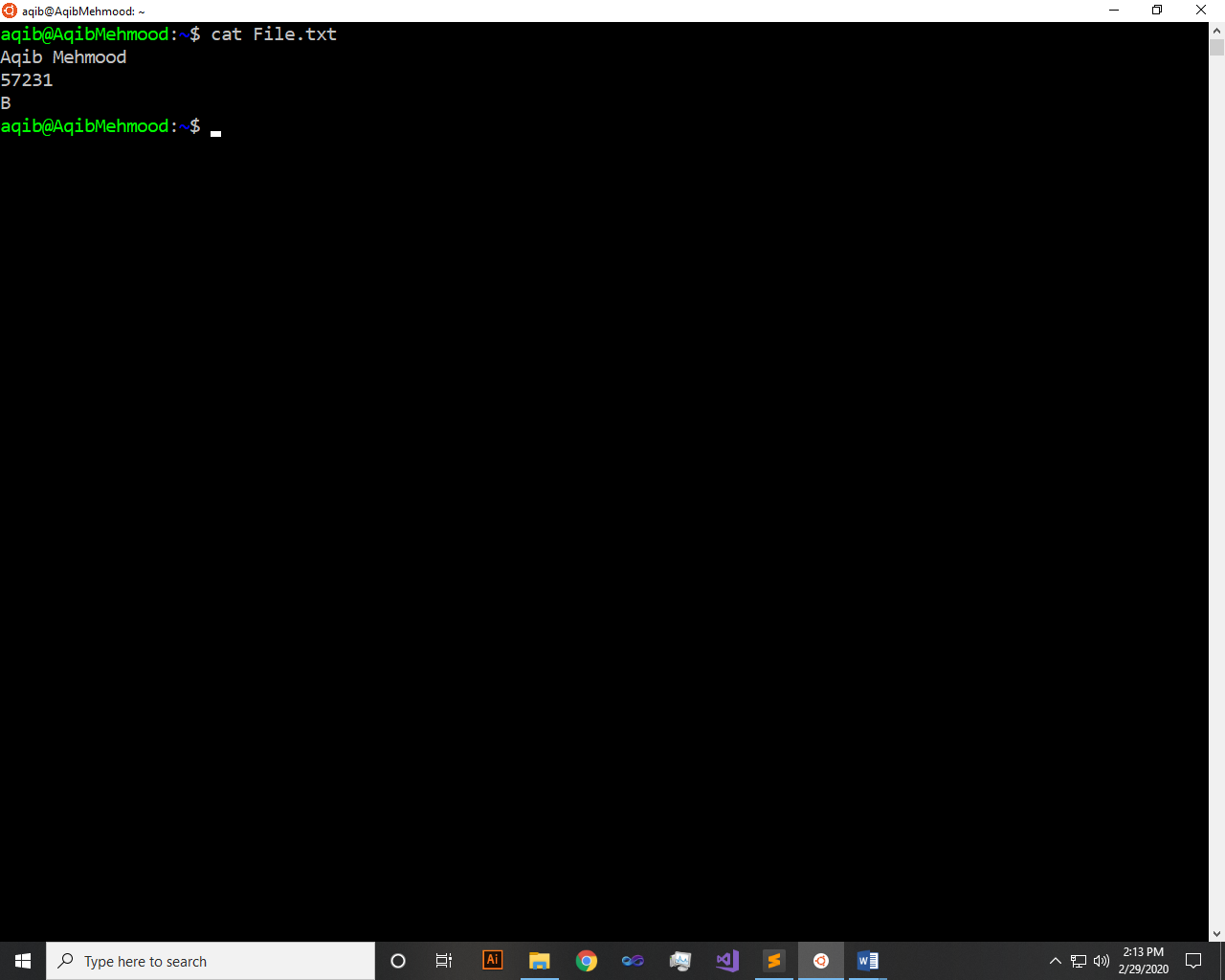
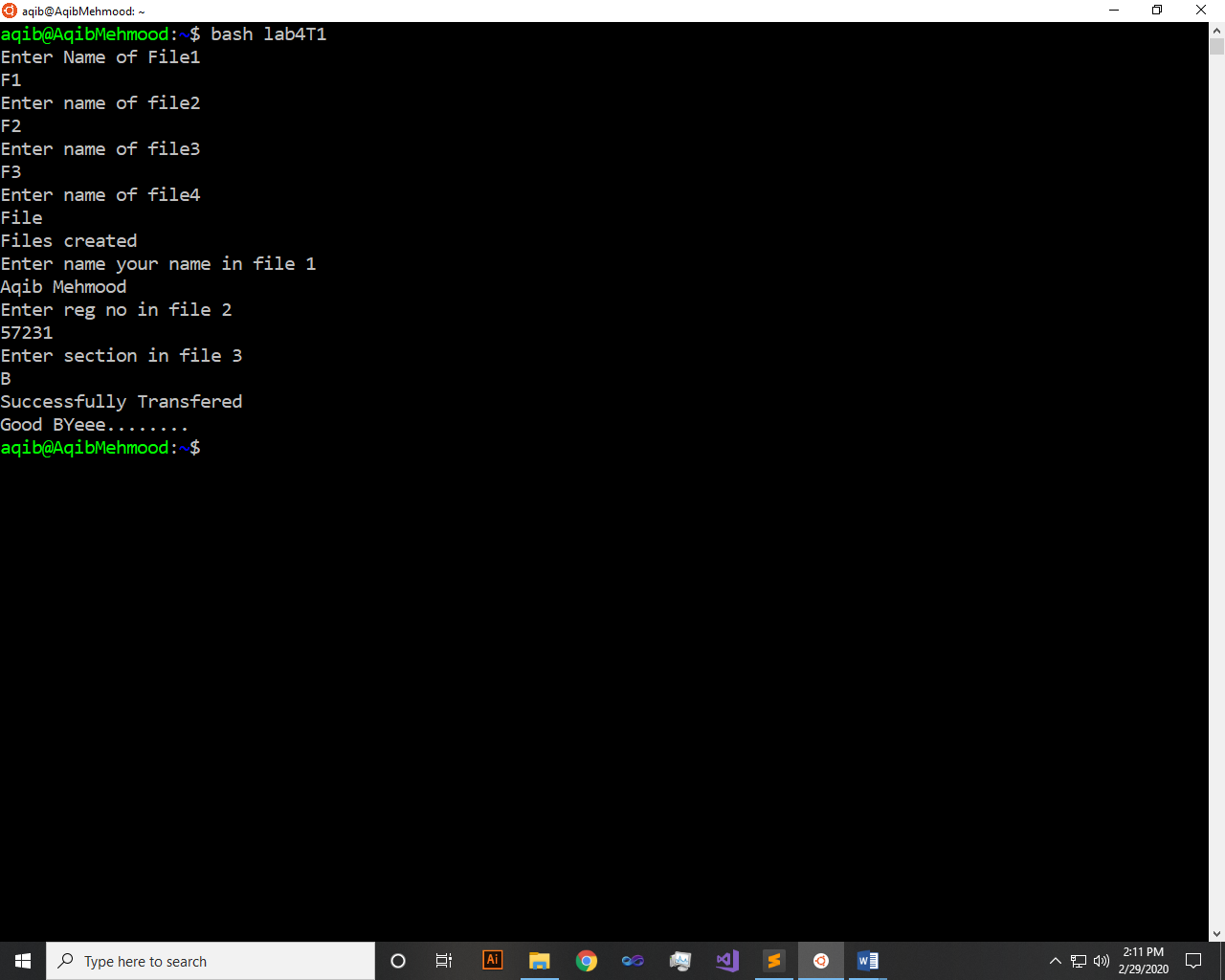
cat $file2.txt>>$file4.txt

cat $file3.txt>>$file4.txt

echo "Successfully Transfered"

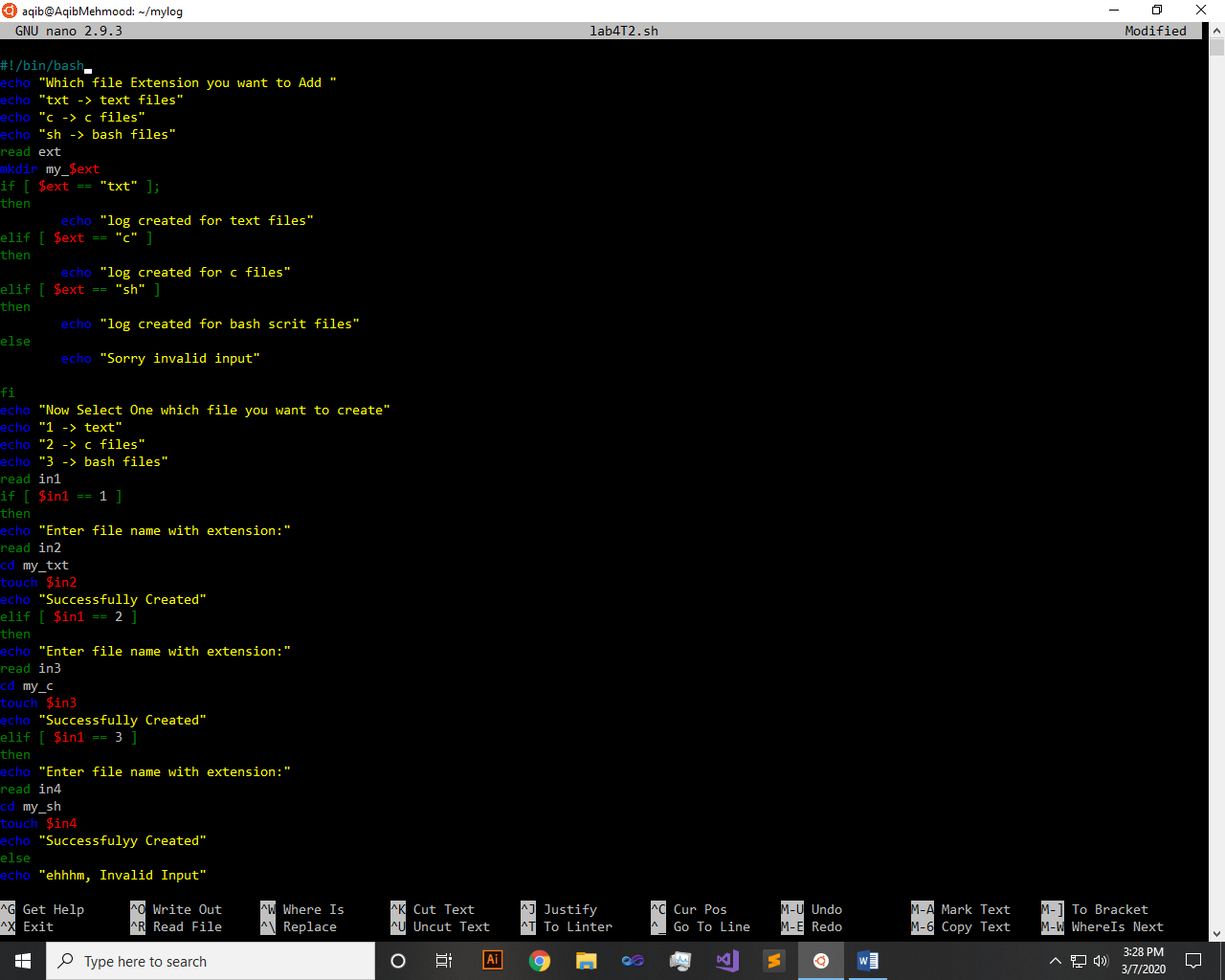
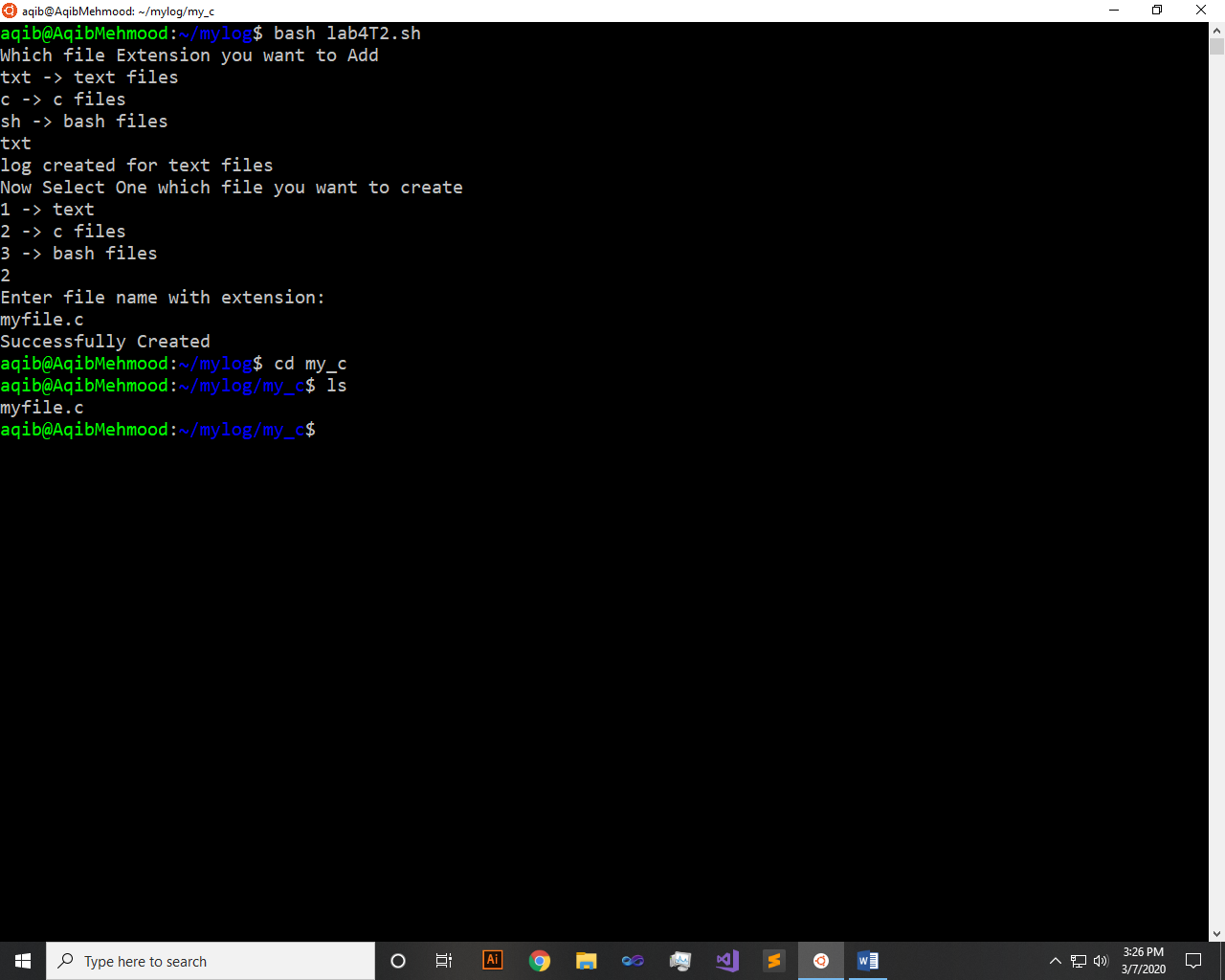
echo "Good BYeee........"

**Output:-**



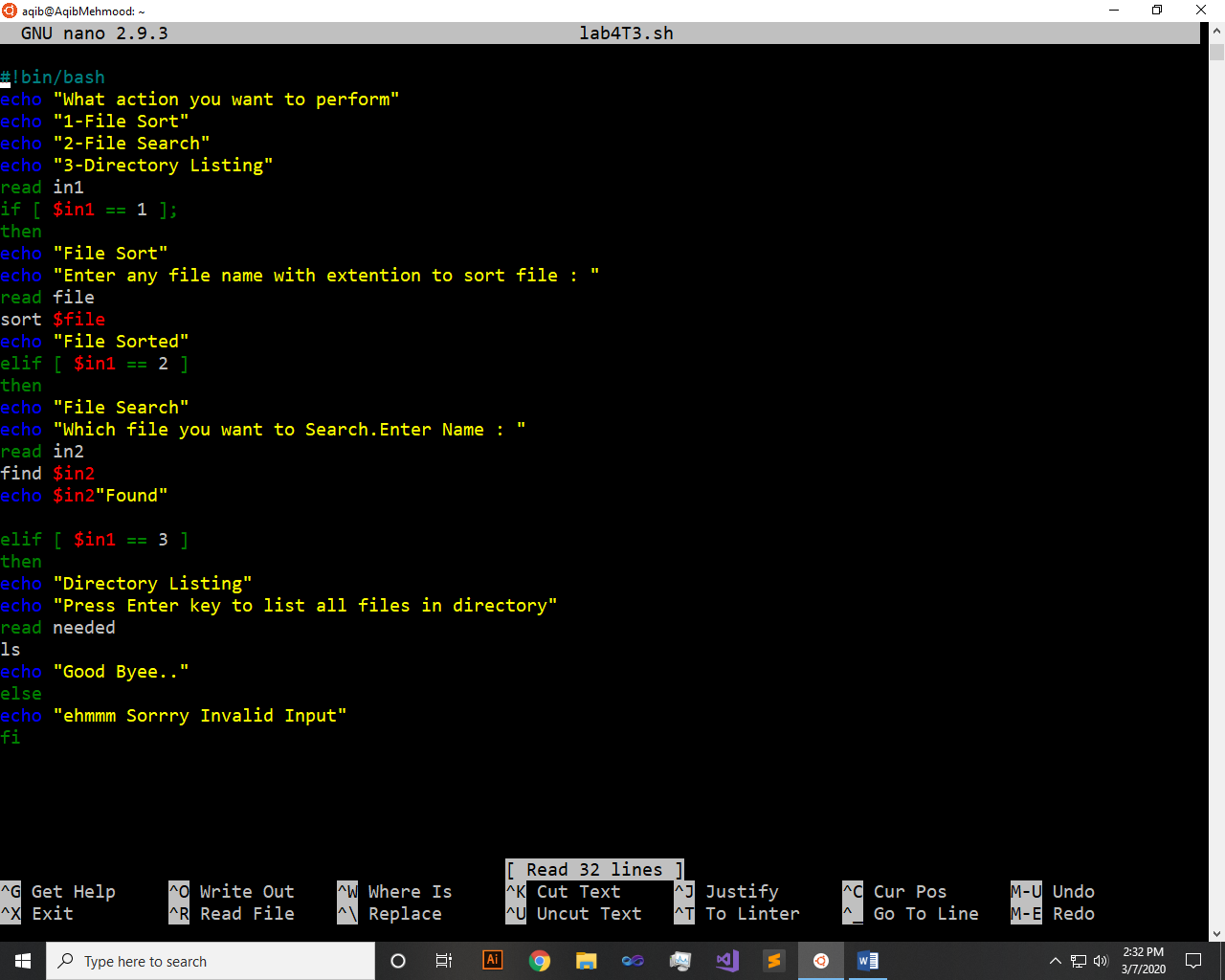
**Task 2**: Write a shell script that creates a “Files Location Log”. The paths of all files, having the same extension, should be stored in one log. The file extension should be taken as an input from the user, and the created logs should be named as “mylog\_extension.txt”, where “extension” is that taken as input from the user. The search process should be for all file in the system, starting from the root directory (/). All log files of different file extension should be stored inside a single directory by the name of “mylog” that would be present at your home directory.

**Code:-**

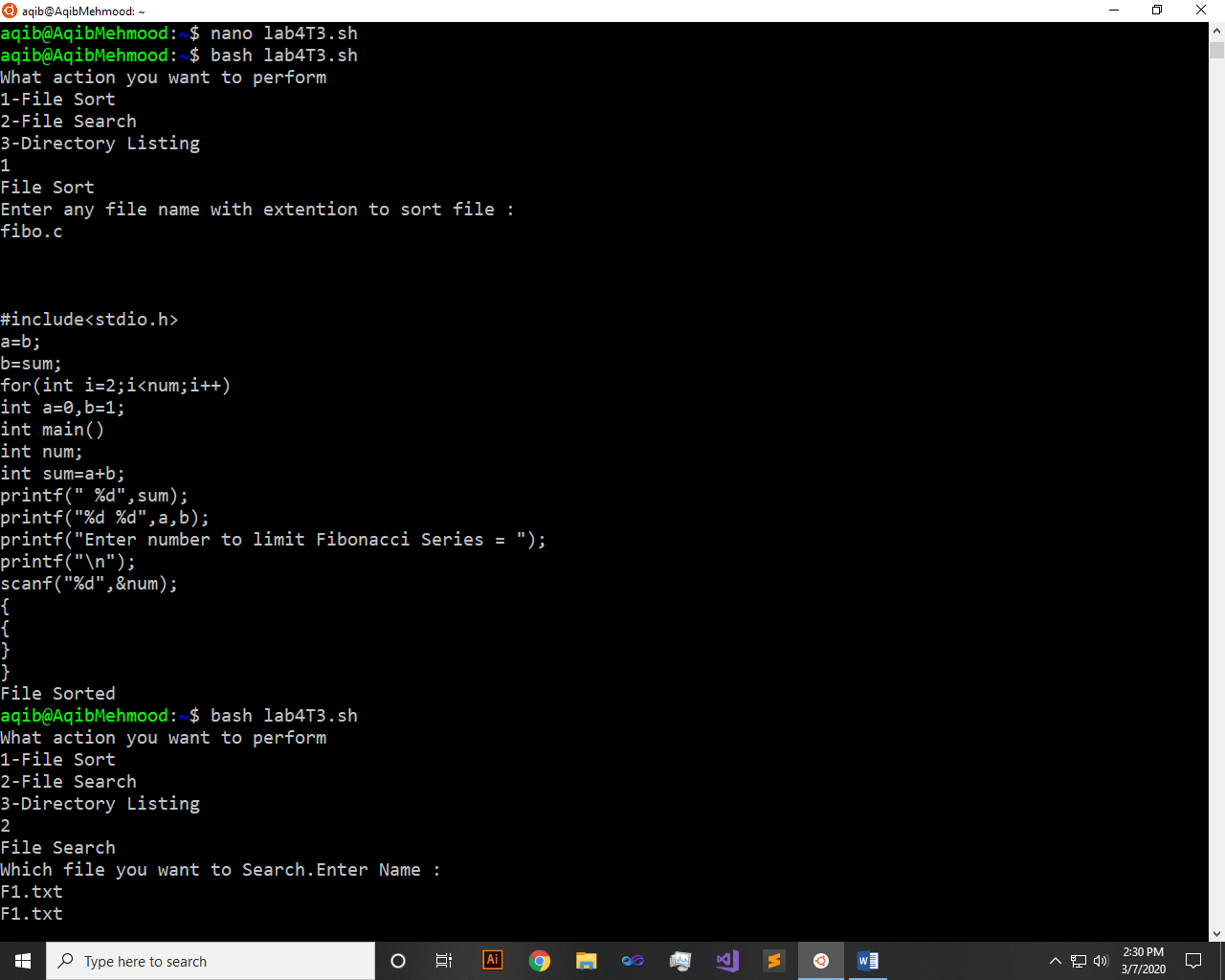
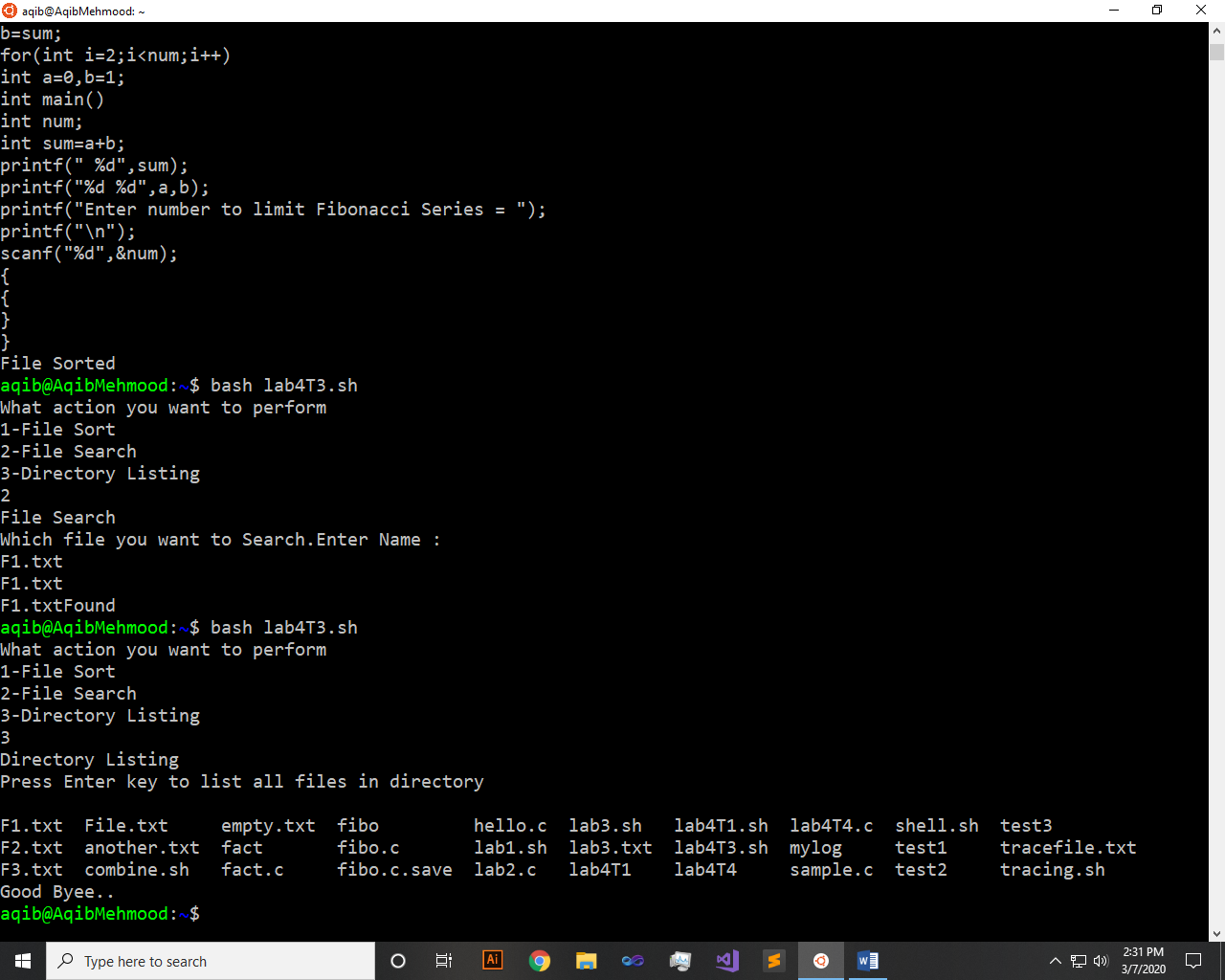
 **Output:-** 

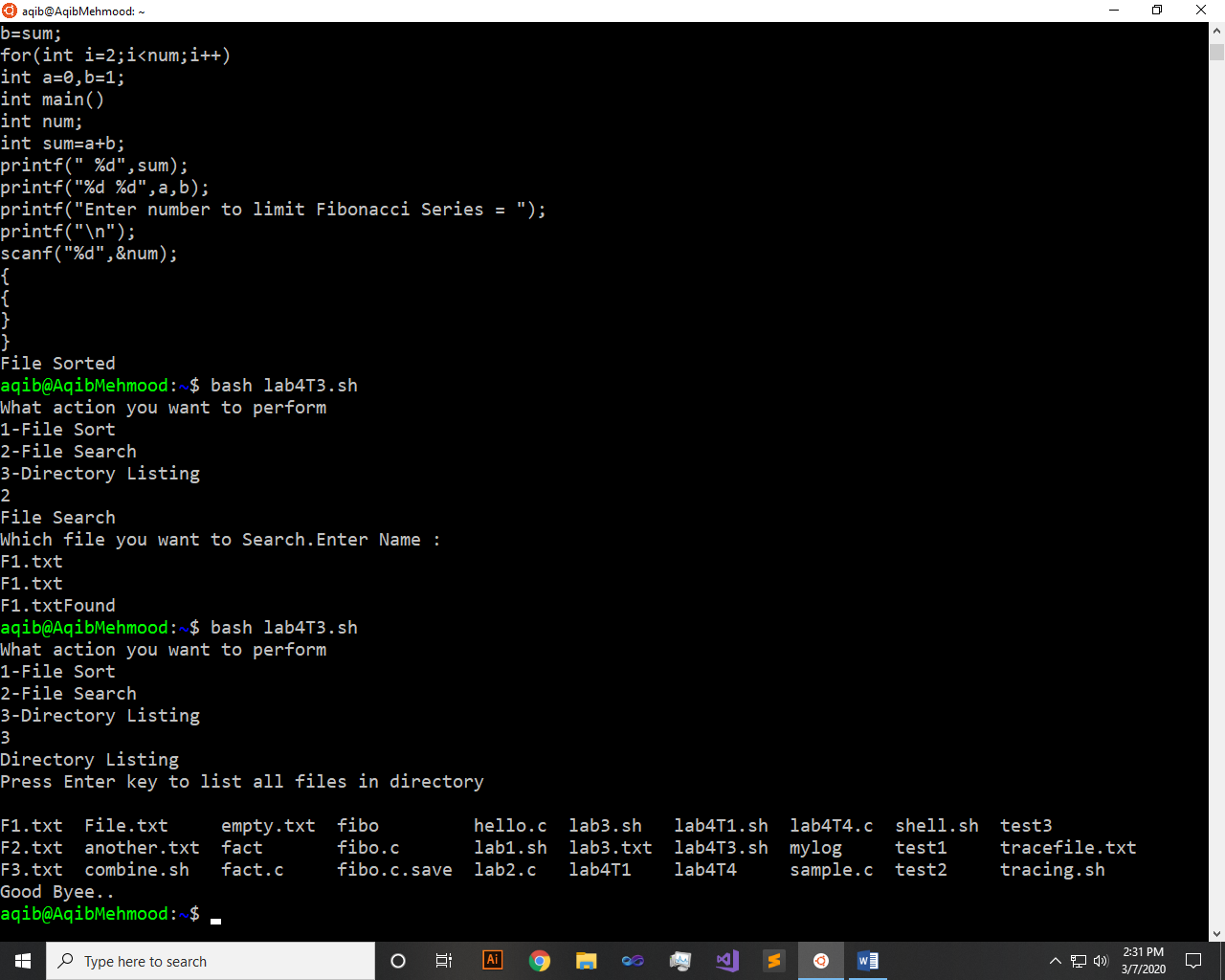
**Task 3**: Write a shell script that either performs a file sort, file search or directory listing operation based on the user’s selection of the operation he/she would like to execute.

**Code:-**



**Output:-**



**Task 4**: Write a C program that takes values of two matrices of size and as input from the user. Multiply the above two matrixes and store the resulting matrix in a 2D array. Display the contents of the first and second matrices and also the resulting matrix. Achieve alignment in the displayed content as much possible.

**Code:-**

#include <stdio.h>

int main(void) {

int a[2][2];

int b[2][2];

int c[2][2];

printf("Enter four values for 1st Matrix\n");

for(int i=1;i<=2;i++)

{

for(int j=1;j<=2;j++)

{

scanf("%d",&a[i][j]);

}

}

printf("Enter four values for 2nd Matrix\n");

for(int i=1;i<=2;i++)

{

for(int j=1;j<=2;j++)

{

scanf("%d",&b[i][j]);

}

}

printf("Matrix 1 \n");

for(int i=1;i<=2;i++)

{

for(int j=1;j<=2;j++)

{

printf("%d\t",a[i][j]);

}

printf("\n");

}

printf("Matrix 2 \n");

for(int i=1;i<=2;i++)

{

for(int j=1;j<=2;j++)

{

printf("%d\t",b[i][j]);

}

printf("\n");

}

//printf("Matrix 1 \* Matrix 2\n");

for(int i=1;i<=2;i++)

{

for(int j=1;j<=2;j++)

{

c[i][j]=0;

for(int k=1;k<=2;k++)

{

c[i][j] += a[i][k] \* b[k][j];

}

}

}

printf("Result of Multiplication:\n");

for(int i=1;i<=2;i++)

{

for(int j=1;j<=2;j++)

{

printf("%d\t",c[i][j]);

}

printf("\n");

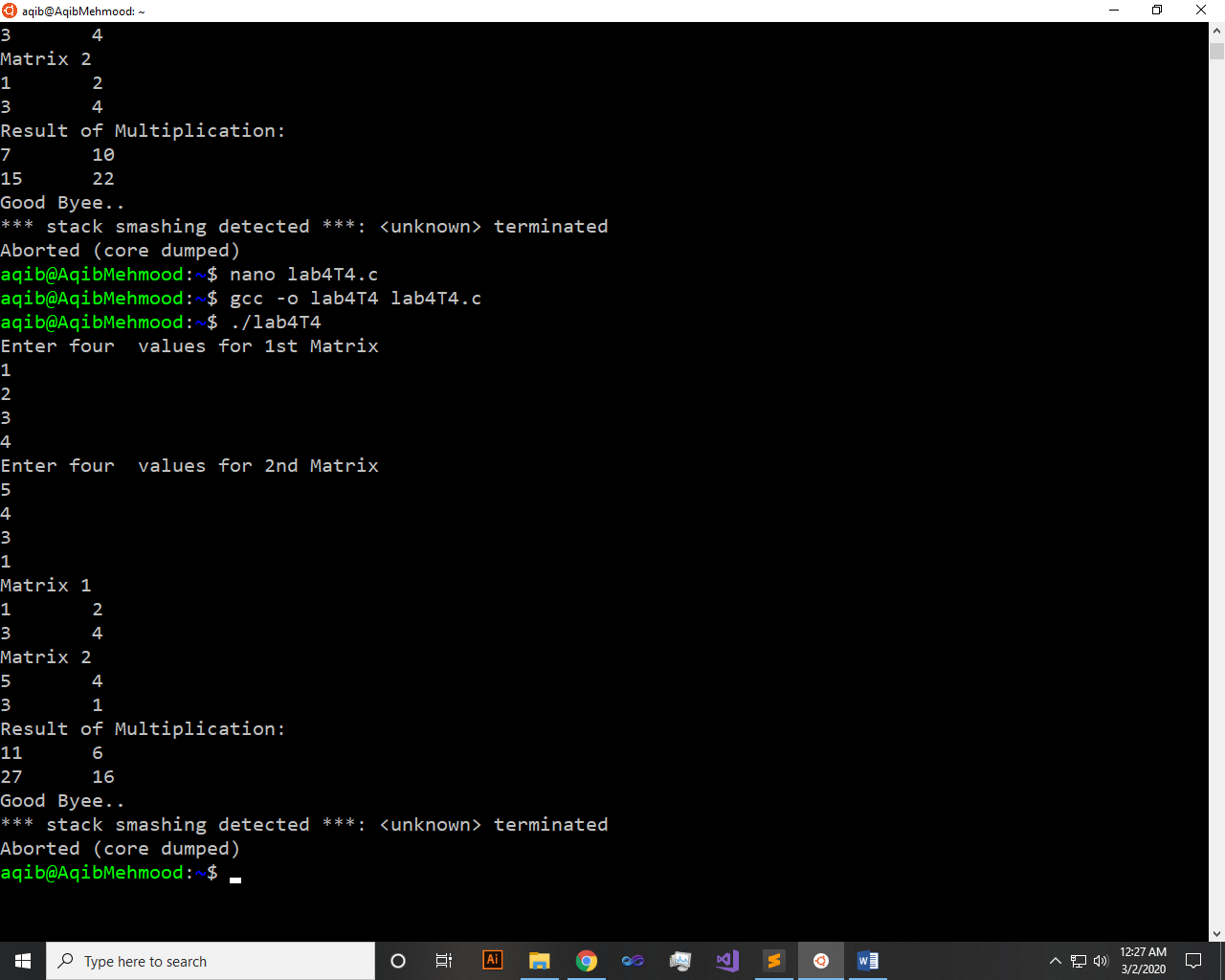
}

printf("Good Bye..\n");

return 0;

}

**Output:-**



~~~~~~\*\*/**THE END**/\*\*~~~~~~