|  |  |
| --- | --- |
| TASK NO | OBJECTIVE |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Task 1**: Answer the following questions.

Q1. In Linux operating system, describe the kernel.

**Answer:-**

The Linux kernel is the main component of a Linux operating system (OS) and is the core interface between a computer’s hardware and its processes. It communicates between the two, managing resources as efficiently as possible.

The kernel has four jobs:

1. **Memory management:** Keep track of how much memory is used to store what, and where
2. **Process management:** Determine which processes can use the CPU, when, and for how long.
3. **Device drivers:** Act as mediator/interpreter between the hardware and processes
4. **System calls and security:** Receive requests for service from the processes.

Q2. In Linux desktop environment, describe the benefits of virtual desktops.

**Answer:-**

Desktop virtualization is an increasingly important technology for many organizations. A virtual desktop or virtual desktop infrastructure (VDI) means that a user’s desktop environment is stored remotely on a server, rather than on a local PC or other client computing device. Desktop virtualization software separates the desktop operating systems, applications and data from the hardware client, storing this “virtual desktop” on a remote server. The remote server that runs and supports virtual desktops uses software called a hypervisor to create a “virtual machine” that simulates the user’s desktop environment and capabilities. In a virtual desktop environment, users access their personal desktop remotely, over the Internet, from any client device.

Desktop virtualization is a valuable technology platform and solves several business problems. With the many benefits desktop virtualization has to offer for an environment, there are four prominent categories that stand out:

1. **Cost Savings**
2. **Simplified Management**
3. **Enhanced Security**
4. **Increased Productivity**

Q3. While GUI based tools do exist in Linux, what is the purpose of using the command line interface, i.e. shell?

**Answer:-**

A Command Line Interface is a powerful way of user interacting with an operating system. One of the first CLIs was the MS-DOS. It was the operating system for the original personal digital computer (PC), which had been built in the 1980s.

Due to the following reasons command line interface are used:

* If the user knows the correct commands then this type of interface can be much faster than any other type of interface.
* This type of interface needs much less memory (Random Access Memory) in order to use compared to other types of user interfaces.
* This type of interface does not use as much CPU processing time as others
* A low resolution, cheaper monitor can be used with this type of interface.
* A CLI does not require Windows to run.

Q4. Use one of the options with the **ls** command, and describe its usage.

**Answer:-**

ls command is used to view the contents of a directory. By default, this command will display the contents of your current directory.

To display total information about Files/Directories(ls -l):



**Task 2:** By using the command line shell interface, practice the commands given in this lab. Write briefly about the usage of each command.

**ls command**:

It is used to list all the contents in the current working directory.

**Syntax:** $ ls – options <arguments>

**Example:-**



**date Command:**

This command is used to display the current date and time.

**Syntax:**

$date

$date +%ch

**Example:-**



**echo Command:**

Writes all its parameters to standard output, separated by spaces.

**Syntax:**

$echo –<options>

**Example:-**



**cd Command:**

Changes the directory.

**Syntax:**

$cd [parameter]

**Example:-**



**man Command:**

This command is used to display text-only manual pages.

**Syntax:**

$man –<options>

**Example:-**



**clear Command:**

This command is used to clear the terminal screen.

**Syntax:**

$clear

**Example:-**



**exit Command:**

Shell sessions can generally be terminated using this command.

**Syntax:**

$exit

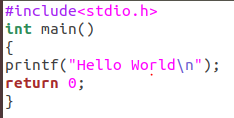
**Example:-**



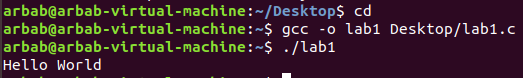
**Task 3**: By using gedit, open a text editor and write the C program given below. Save the written file as “hello.c”. In order to compile and execute the output file, do the following:

Write down the output of the program below (provide snapshot).

Code:-

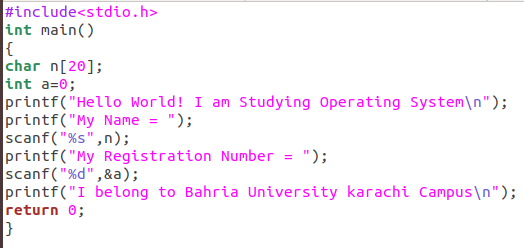


Output:-

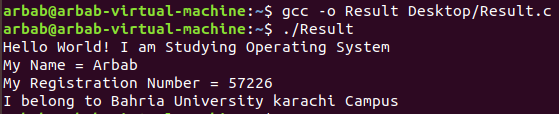


**Task 4**: Make changes within the above program to display a new output text as given below. Write down the developed program.

Code:-



Output:-



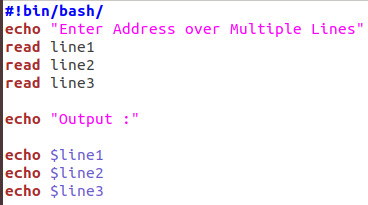
**Lab 2**

**Task 1**: Practice all the Linux commands discussed in this lab while taking assistance using the **man** command. Write the complete syntax used for utilizing the **cp**, **mv** and **rm** commands in Linux shell.

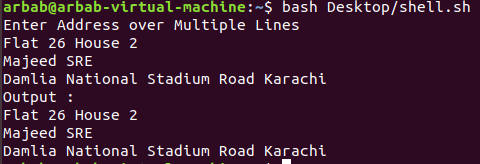
|  |  |
| --- | --- |
| cp | Copy file or directory |
| mv | Move file or directory |
| rm | Remove file or directory |

**Task 2**: Write a shell script to display your address over multiple lines.

Code:-



Output:-



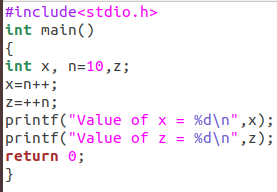
**Task 3**: Write a shell script that would traverse among any three directories that are placed under the /home directory. While moving from one directory to another, the script should display the name of the current working directory and list the content within that directory, including the hidden files.

**Output:-**

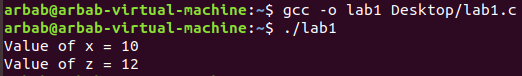


**Task 4**: Write the C programs provided in this lab and generate their outputs over Linux environment (provide snapshot).

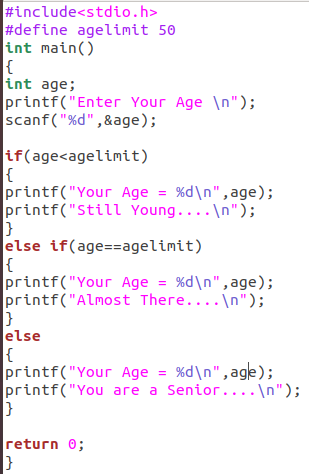
**Code:-**



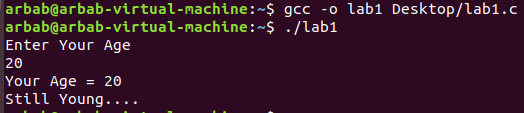
**Output:-**



**Code:-**

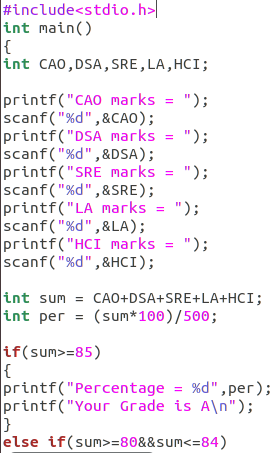
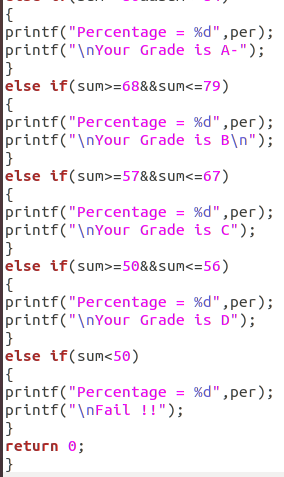


**Output:-**

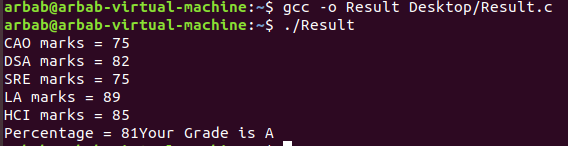


**Task 5:** Write a C program on the Linux environment that takes your marks as an input and display your grades accordingly to that followed at Bahria University. Limit your program to a maximum of five subjects. Use the suitable logical operator(s), i.e. and (&&), or (||), not (!), if required.

**Code:-**

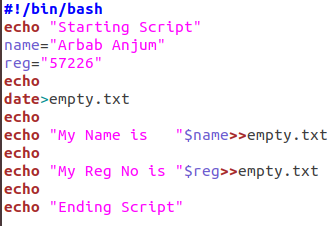
**Output:-**



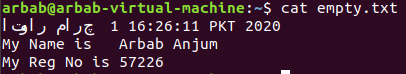
**Lab 3**

**Task 1**: Create an empty file with a .txt extension. Write a shell script that would write the current date, student’s name and registration number into that file, while using variables for all three entries.

**Code:-**



**Output:-**

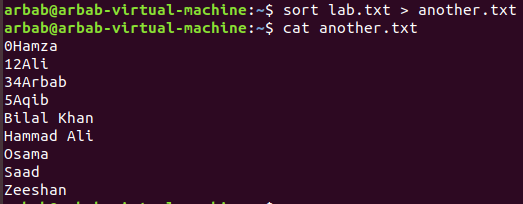


**Task 2**: Create a .txt file and input ten lines of entry while mixing it with both alphanumeric characters. Sort the contents of the created file in an ascending order and write the sorted output into another file.

**Code:-**



**Output:-**

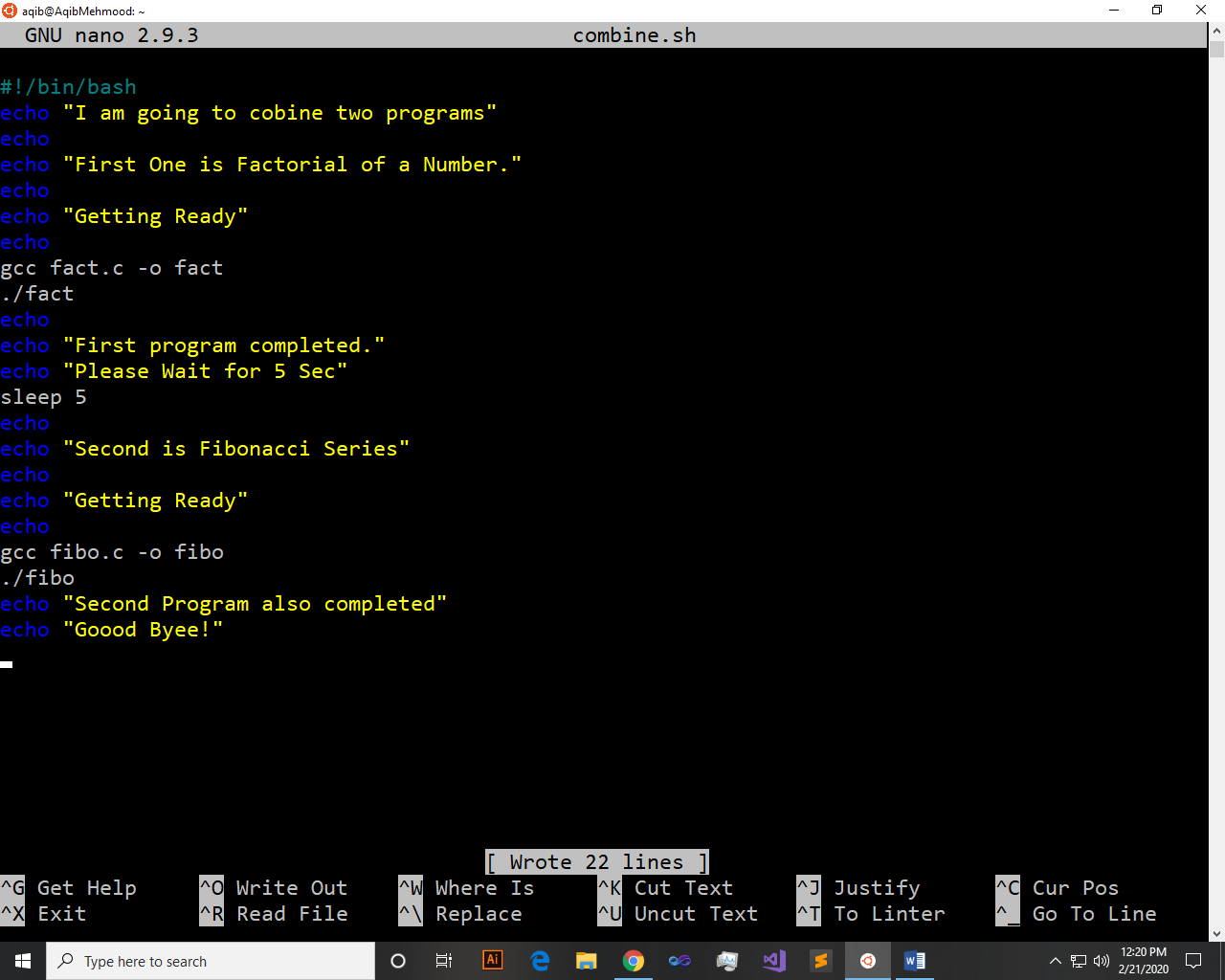


**Task 3**: Write two different C language programs that would generate the following for a given number, e.g.:

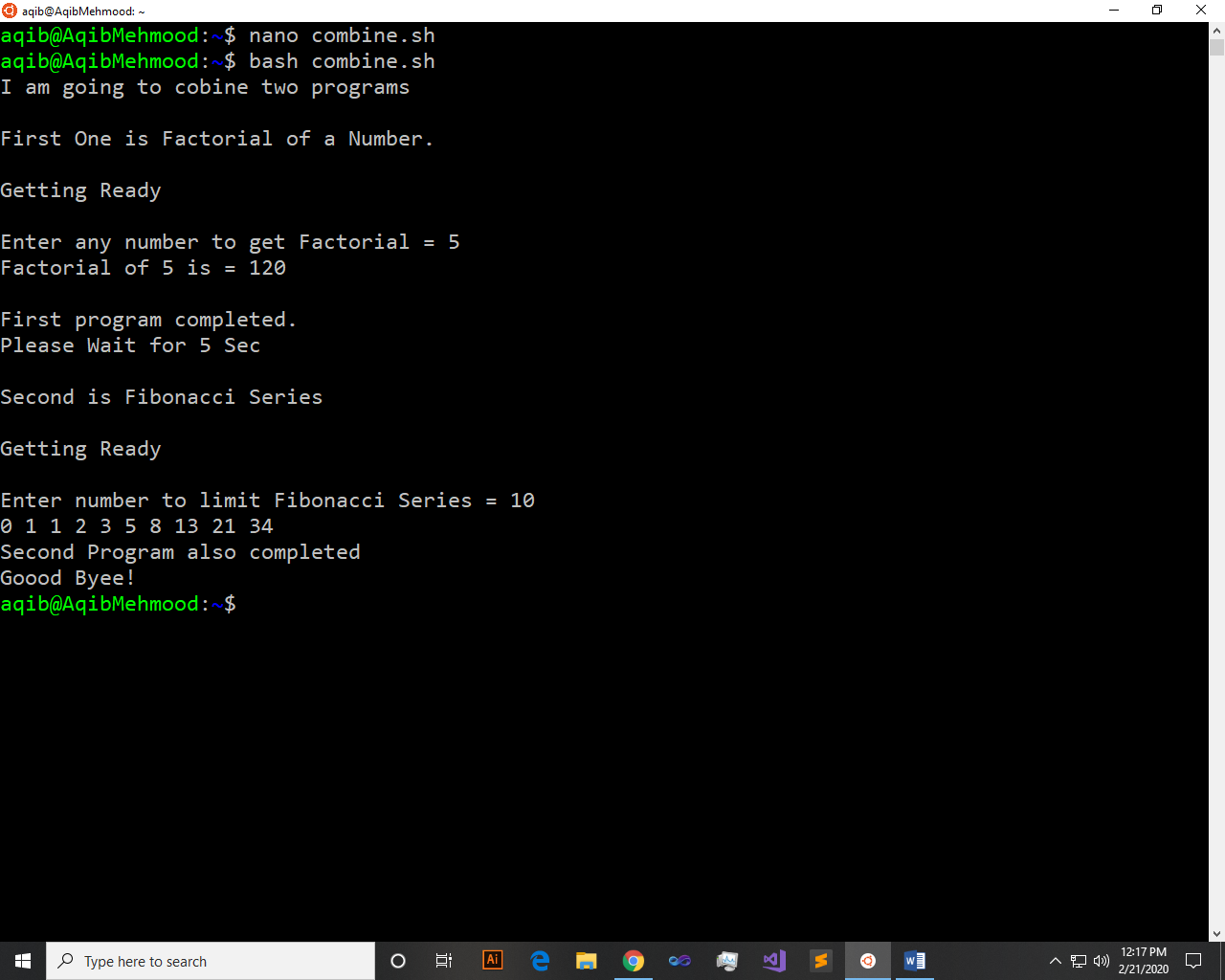
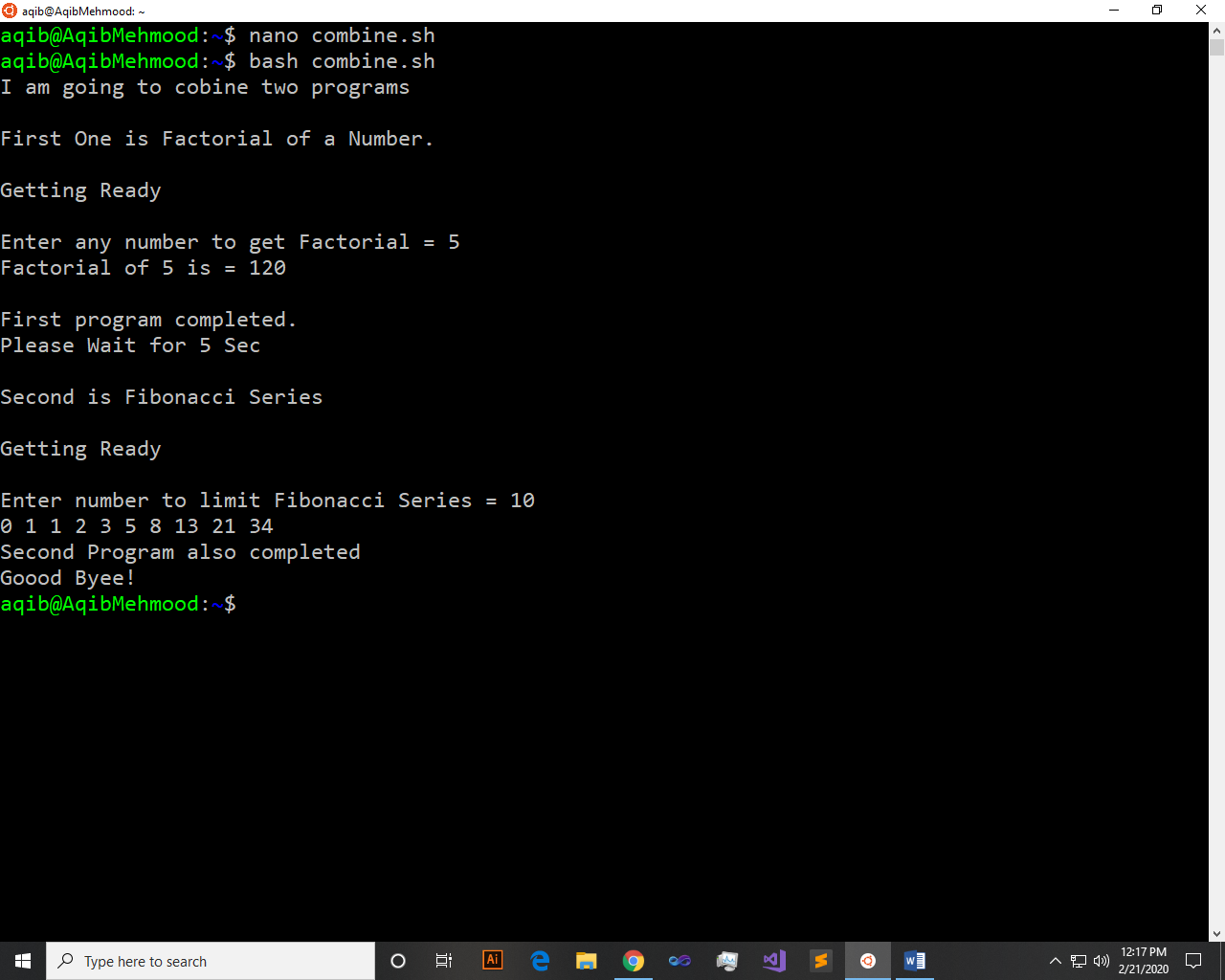
1. Calculate the factorial of.
2. Calculate the Fibonacci series (0, 1, 1, 2, 3, 5,8…) up to .

Compile and run both programs using a single shell script, while having a running gap of 5 seconds between the first and second program. The generated output should properly display on which program is currently running (tip use ***echo*** command).

**Code:-**



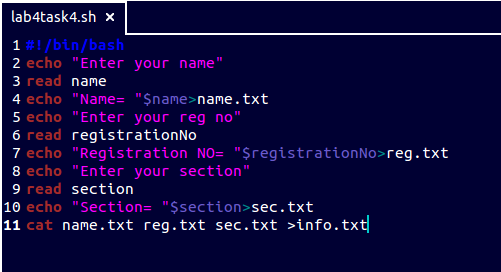
**Output:-**



Lab 4

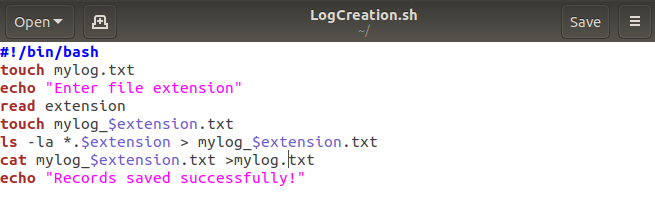
**Task1: Write a single shell script that creates four different files, while taking the names of all created files as input from the user. As the files contents, insert your name in the first file, registration number in the second and section details in the third. These should be followed by merging the contents of all three files into the fourth one.**

**Code:**

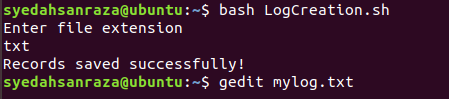


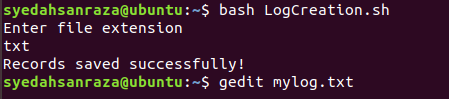
**Task2: Write a shell script that creates a “Files Location Log”. The paths of all files, having the same extension, should be stored in one log. The file extension should be taken as an input from the user, and the created logs should be named as “mylog\_extension.txt”, where “extension” is that taken 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.**

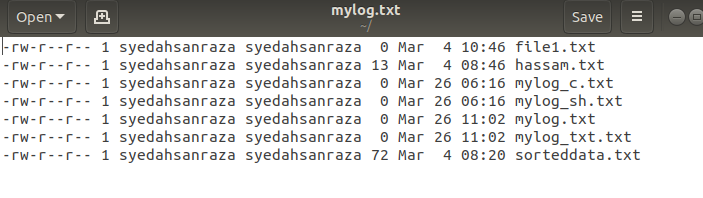
**Code:**



**Output:**

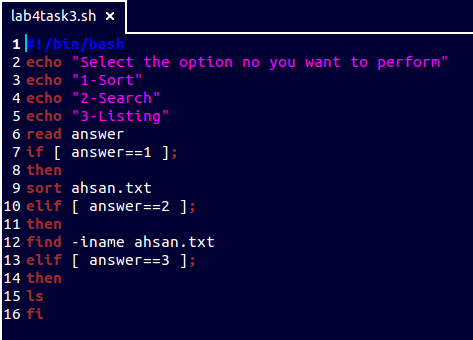




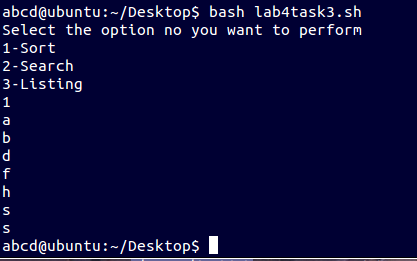


**Task3: Write a shell script that either performs a file sort, file search or directory listing operation based on the user’s selection of the operation he/she would like to execute.**

**Code:**



**Code**:



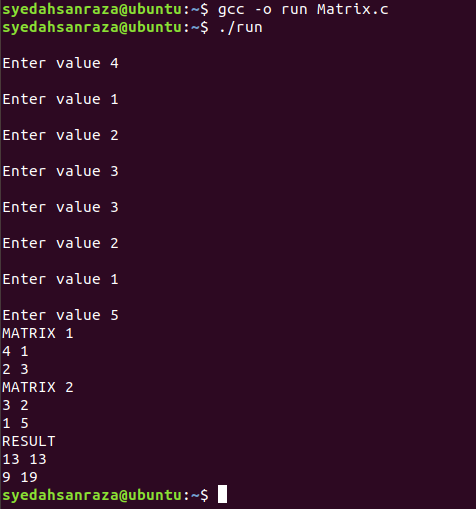
**Task4: Write a C program that takes values of two matrices of size (𝑚×1) and (1×𝑛) as input from the user. Multiply the above two matrixes and store the resulting (𝑚×𝑛) matrix in a 2D array. Display the contents of the first and second matrices and also the resulting matrix. Achieve alignment in the displayed content as much possible.**

**Code:**





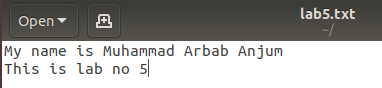
**Output:**



Lab 5

**Task 1**: While utilizing the Linux commands studied so far, provide an example *(other than the one shown in this Lab)* of a combination of several Linux commands in which pipes are used more than once. Also provide a snapshot of the generated output.

CODE:

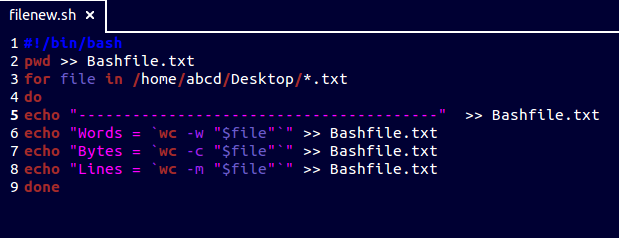


OUTPUT:

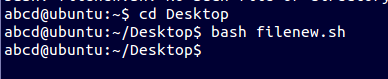


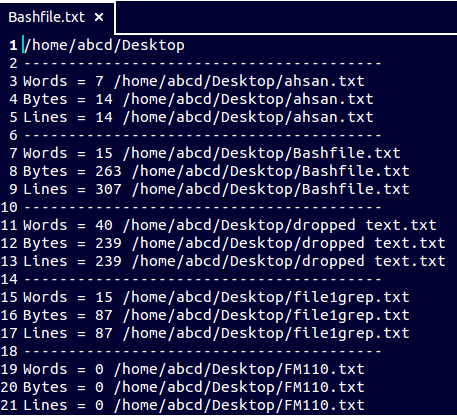
**Task 2**: Write a shell script that records the full path of all the files present within a directory into a record.txt file. Along with full path name, the script should also record the number of words, characters and lines within each file.

CODE:



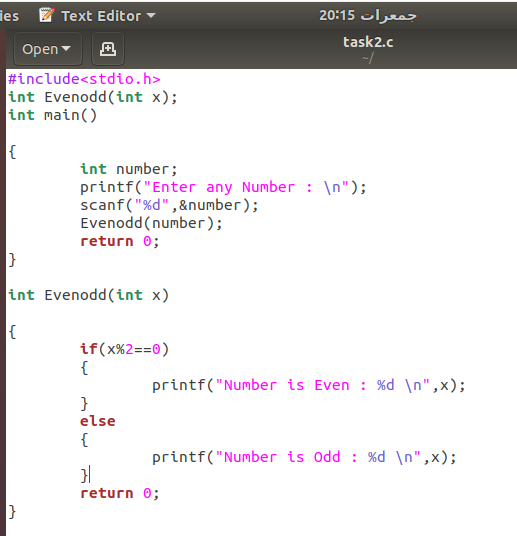
**OUTPUT:**





**Task 3**: Write a C program that asks the user to provide an integer input in the ***main()*** function. The program would call a function ***even\_odd()*** from the ***main()*** function, where the function ***even\_odd()*** accepts an integer as an argument, determine and display if the passed integer is either even or odd.

CODE:



OUTPUT:

