



OPERATING SYSTEM (O.S.) LAB

SUBJECT CODE – BCSC 0803

SUBMITTED BY :

Name – Rohit Choudhary

Course – B.Tech CSE 2nd year

University Roll No. – 2215001485

SUBMITTED TO :

Mr. Ajitesh Kumar



Linux

ABOUT LINUX ENVIRONMENT

Linux is an open-source operating system kernel that serves as the foundation for various distributions (distros) of the Linux operating system. It was initially created by Linus Torvalds in 1991 and has since evolved into a powerful and versatile platform used in a wide range of computing devices, from servers and desktop computers to embedded systems and smartphones.

Here's a brief introduction to Linux covering its key features, components, and usage:

Key Features of Linux:

- 1. Open Source:** Linux is distributed under the GNU General Public License (GPL), which means its source code is freely available for anyone to view, modify, and distribute.
- 2. Multitasking and Multiuser:** Linux supports multitasking, allowing multiple processes to run concurrently. It also supports multiple users accessing the system simultaneously with secure user permissions.

3. **Stability and Reliability**: Linux is known for its stability and reliability, often running for extended periods without needing to reboot. This makes it a popular choice for servers and critical systems.
4. **Security**: Linux provides robust security features, including user authentication, file permissions, and access controls, which contribute to its reputation for security.
5. **Flexibility**: Linux offers a high degree of flexibility and customization. Users can choose from various desktop environments, package managers, and software repositories to tailor their Linux experience to their preferences.
6. **Scalability**: Linux is highly scalable, capable of running on a wide range of devices, from small embedded systems to large supercomputers.



ABOUT WINDOWS

Windows is a family of operating systems developed by Microsoft Corporation, which has been a dominant force in the personal computer (PC) market for decades.

Here's an introduction covering its key features, versions, and common use cases:

Key Features of Windows:

- 1. Graphical User Interface (GUI):** Windows is known for its user-friendly GUI, featuring a desktop environment with icons, windows, and menus, making it easy for users to navigate and interact with their computers.
- 2. Multitasking and Multiuser:** Like Linux, Windows supports multitasking, allowing multiple programs to run simultaneously. It also supports

multiple user accounts with customizable permissions.

3. **Compatibility**: Windows is compatible with a vast array of hardware devices, software applications, and peripherals, making it suitable for a wide range of use cases and environments.
4. **Security**: Windows includes built-in security features such as Windows Defender antivirus software, Windows Firewall, and User Account Control (UAC) to help protect against malware, viruses, and unauthorized access.
5. **Integration**: Windows seamlessly integrates with other Microsoft products and services, such as Office productivity suite, OneDrive cloud storage, and Azure cloud computing platform, providing a cohesive ecosystem for users and businesses.
6. **Updates and Support**: Microsoft regularly releases updates, patches, and new versions of Windows to improve performance, security, and compatibility. Users can receive support through Microsoft's online resources, community forums, and customer service channels.



Introduction To Cocalc..

CoCalc is a virtual online workspace for calculations, research, collaboration and authoring documents. Your web browser is all you need to escape the confined space of your desktop and move to the cloud. This guide explains the features of CoCalc in depth and shows how you can use them productively.

The main building blocks for working on CoCalc are [Projects](#). Create one or more projects in order to partition your work into separate *workspaces*. Each project consists of files, accessible only to you and your [collaborators](#).

You and your collaborators can *edit these files at the same time*, which means your changes are synchronized among all of you in real-time! Depending on the type of the file, it allows you to accomplish a specific task. They open up in their associated online editor and you start to work interactively in CoCalc's environment. For example, files ending in `*.sagews` are for working with [\[SageMath\]](#) in [Sage Worksheets](#), an `*.ipynb` file starts CoCalc's implementations of

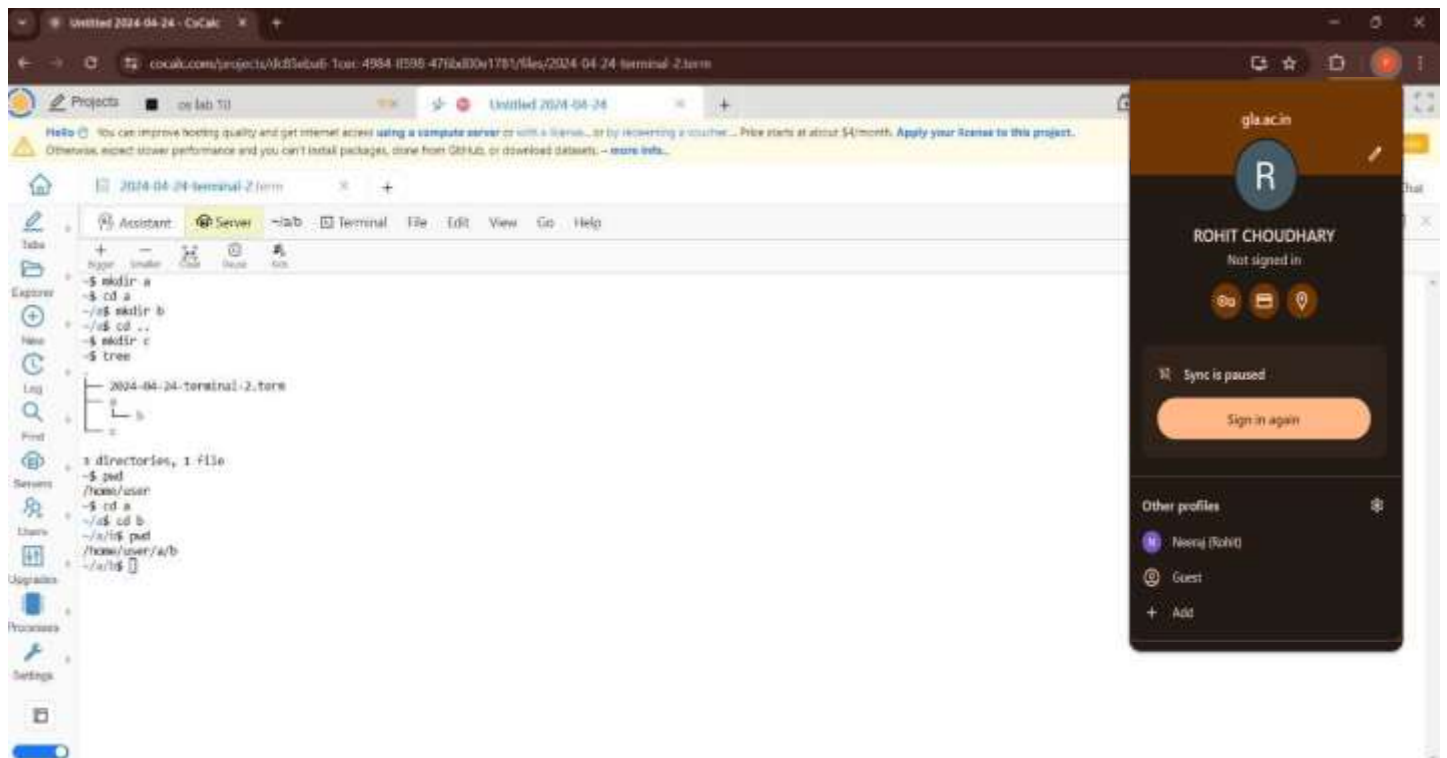
the [Jupyter Notebook](#), and a `*.tex` file opens an editor for [LaTeX documents](#)

Index

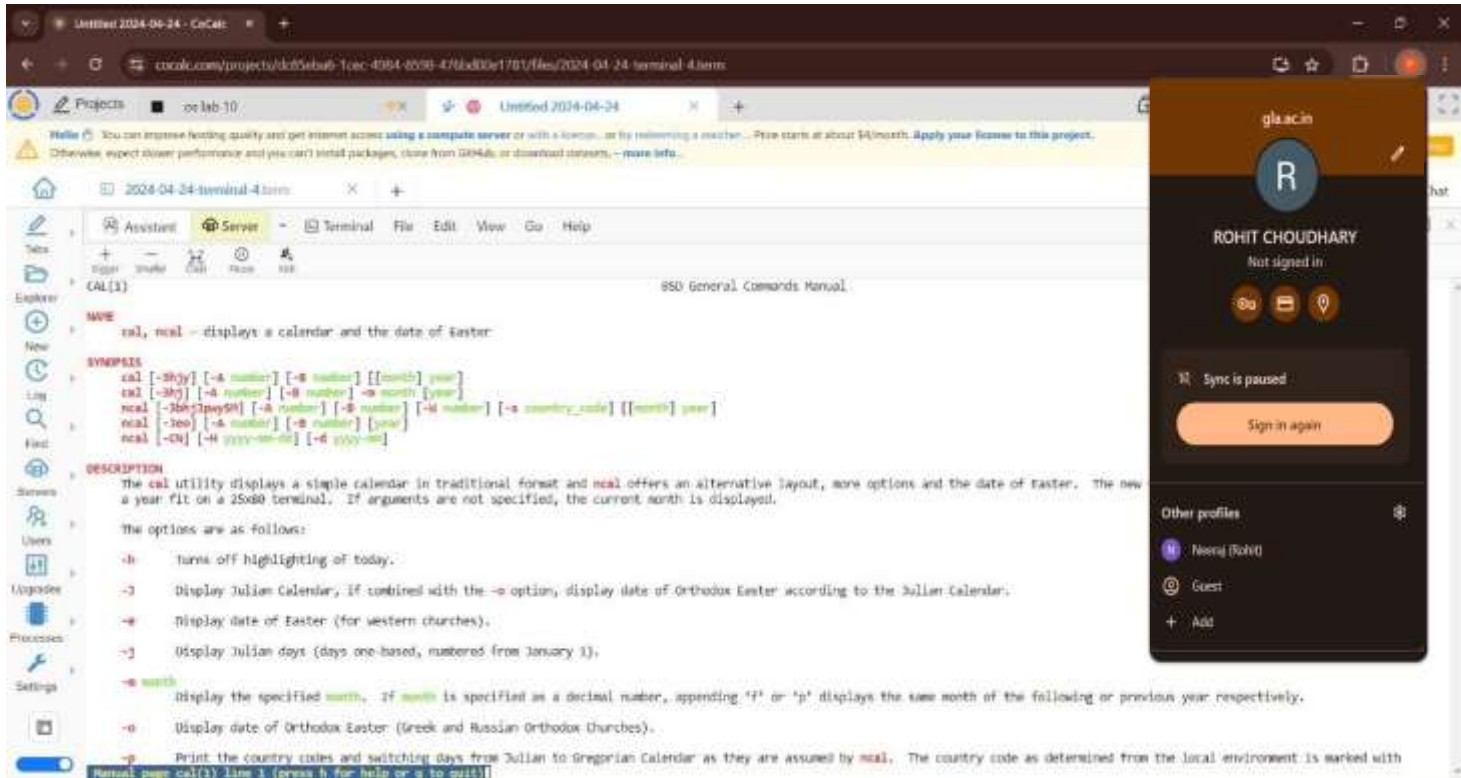
S. No.	Topic	Page No.	Date of Exp.	Date of Submission	Sign
1.	Directory commands	1	10/01/24	17/01/24	
2.	man command	2	17/01/24	24/01/24	
3.	Program to calculate sum	3	24/01/24	31/01/24	
4.	Program to calculate factoria using shell script	4	31/01/24	07/02/24	
5.	Program to calculate factoria using C language	5	07/02/24	14/02/24	
6.	Program to print triangle pattern	6	14/02/24	21/02/24	
7.	Program to find smallest and largest number	7	21/02/24	13/03/24	
8.	Print Fibonacci series	8	13/03/24	20/03/24	

9.	Program to check Palindrome number	9	20/03/24	27/03/24	
10.	Program to check even or odd	10	27/03/24	03/04/24	
11.	To print calendar of 2024	11	03/04/24	10/04/24	
12.	Program to check if a year is leap or not	12	10/04/24	24/04/24	

Directory commands



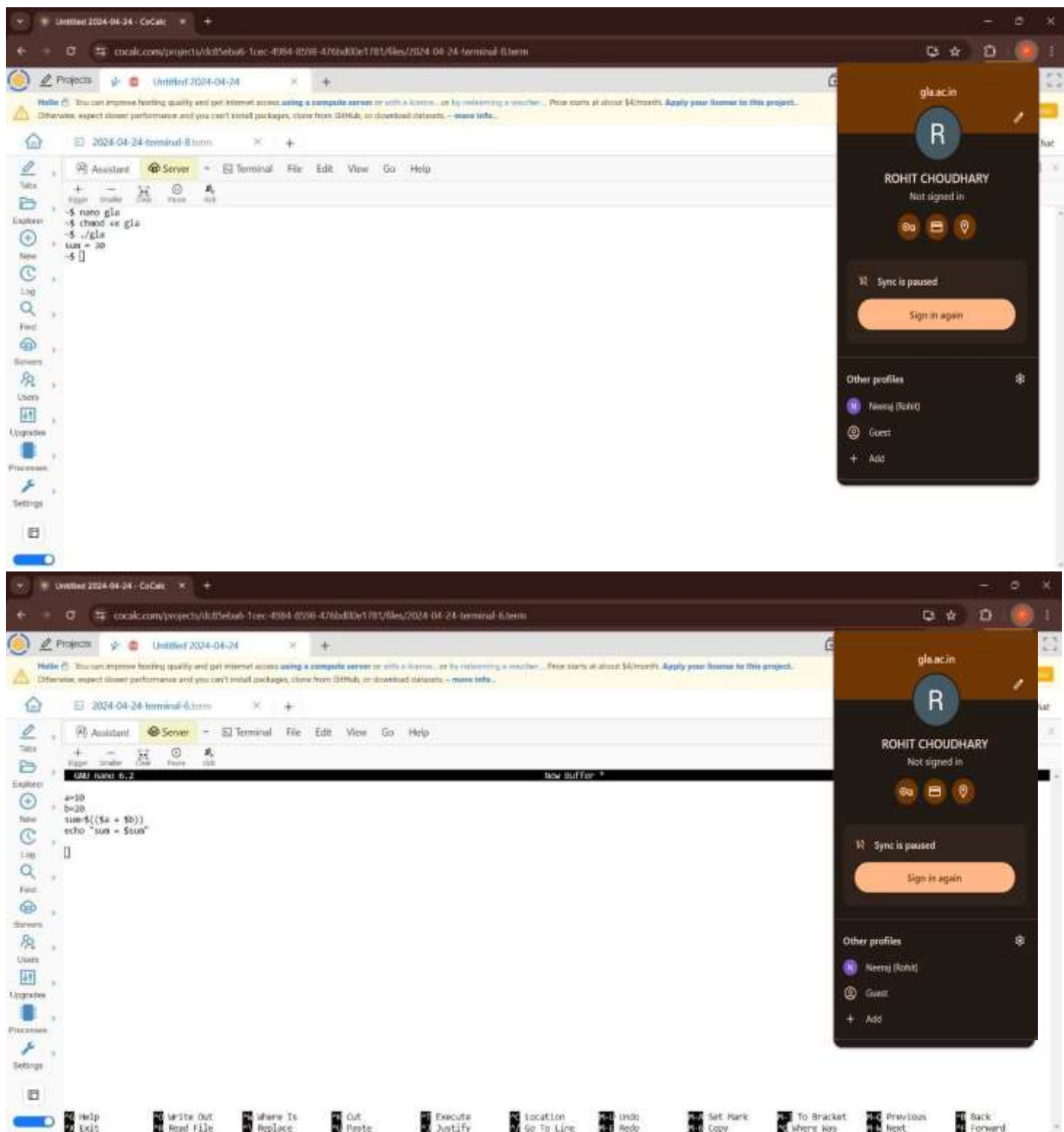
man command



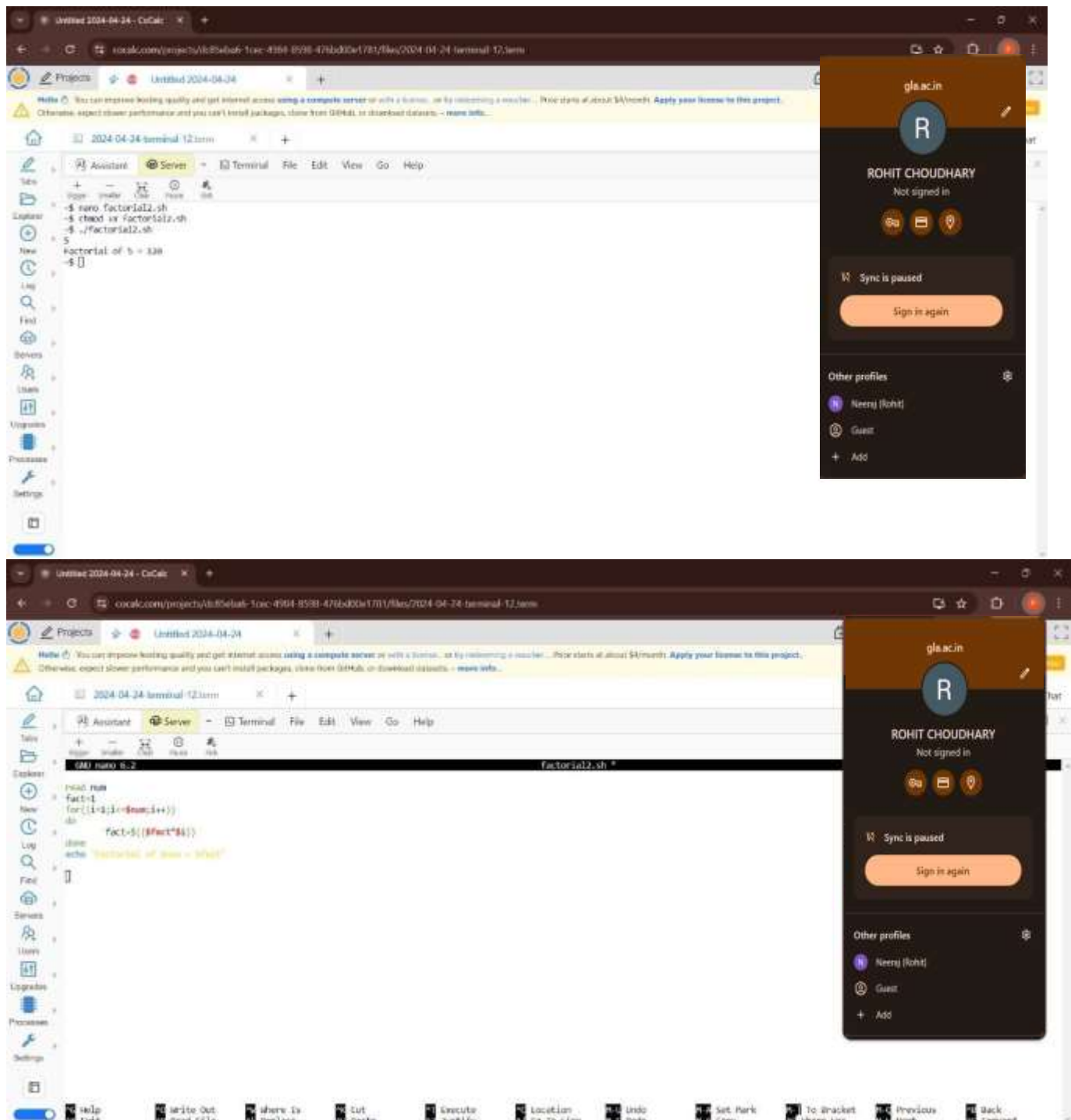
The screenshot shows a web-based terminal interface with a sidebar on the right. The terminal window displays the output of the `man cal` command. The sidebar contains a user profile for Rohit Choudhary, who is not signed in, and a list of other profiles including Neeraj (Rohit), Guest, and an Add button.

```
BSD General Commands Manual.  
  
NAME  
cal, ncal - displays a calendar and the date of Easter  
  
SYNOPSIS  
cal [-shjy] [-d number] [-B number] [[month] year]  
cal [-shj] [-d number] [-B number] -o month [year]  
ncal [-bhe] [-p year] [-d number] [-B number] [-s country_code] [[month] year]  
ncal [-bhe] [-d number] [-B number] [year]  
ncal [-CN] [-H yyyy-mm-dd] [-d yyyy-mm]  
  
DESCRIPTION  
The cal utility displays a simple calendar in traditional format and ncal offers an alternative layout, more options and the date of Easter. The new  
a year fit on a 25x80 terminal. If arguments are not specified, the current month is displayed.  
  
The options are as follows:  
-h Turns off highlighting of today.  
-j Display Julian Calendar; if combined with the -o option, display date of Orthodox Easter according to the Julian Calendar.  
-e Display date of Easter (for western churches).  
-J Display Julian days (days one-based, numbered from January 1).  
-m month Display the specified month. If month is specified as a decimal number, appending 'f' or 'p' displays the same month of the following or previous year respectively.  
-o Display date of Orthodox Easter (Greek and Russian Orthodox Churches).  
-p Print the country codes and switching days from Julian to Gregorian Calendar as they are assumed by ncal. The country code as determined from the local environment is marked with  
Manual page cal(1) line 1 (press h for help or q to quit)
```

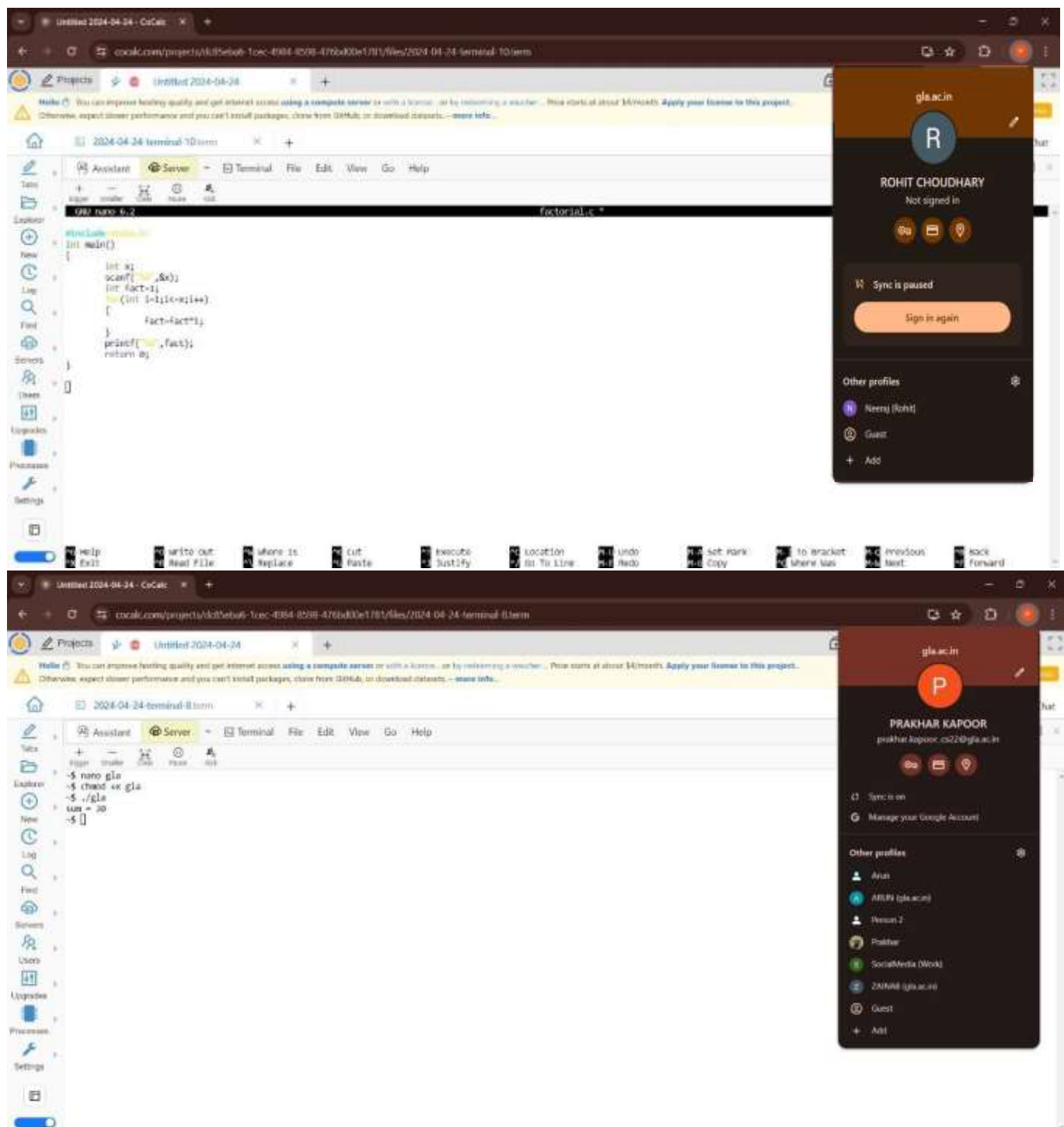
Program to calculate sum



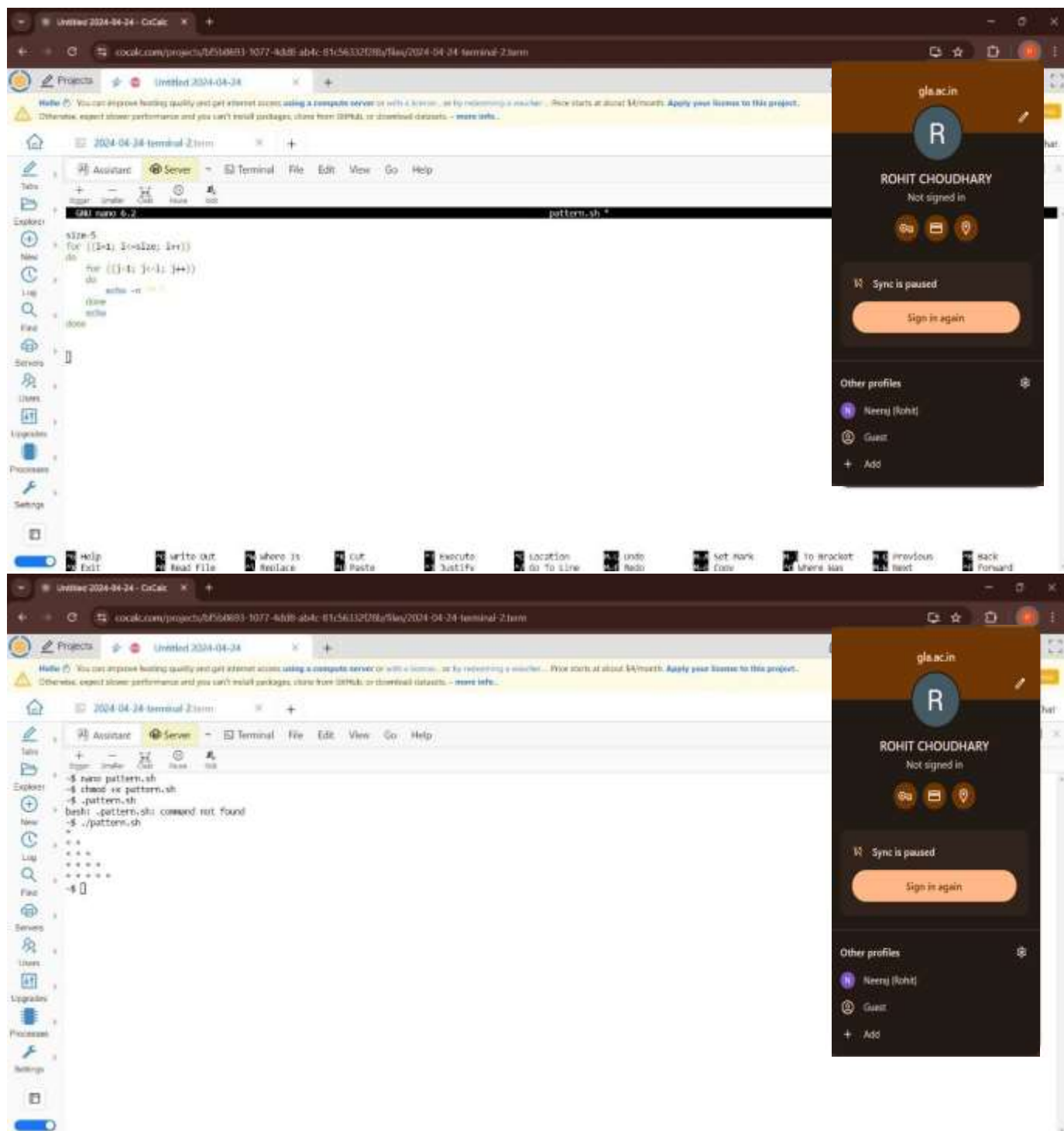
Program to calculate factorial using shell script



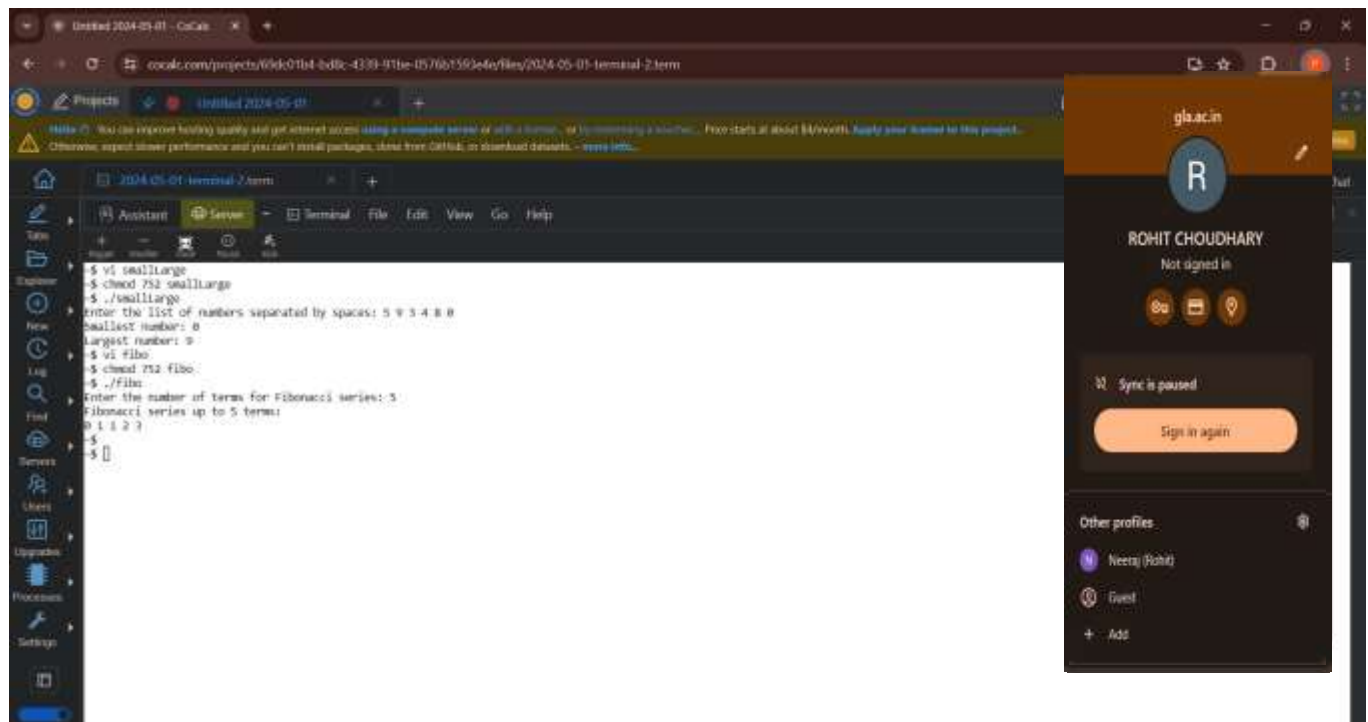
Program to calculate factorial using C language



Program to print triangle pattern

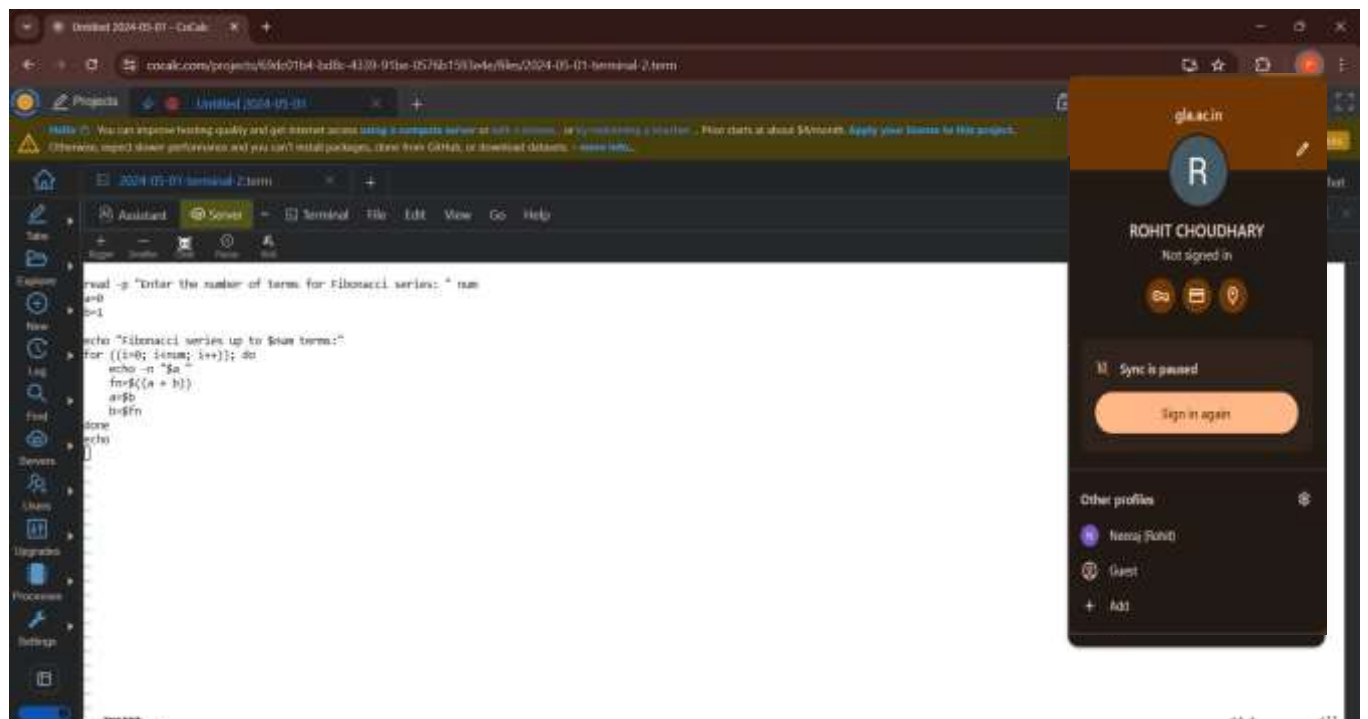


Print Fibonacci series



The screenshot shows a VS Code terminal window with the following commands and output:

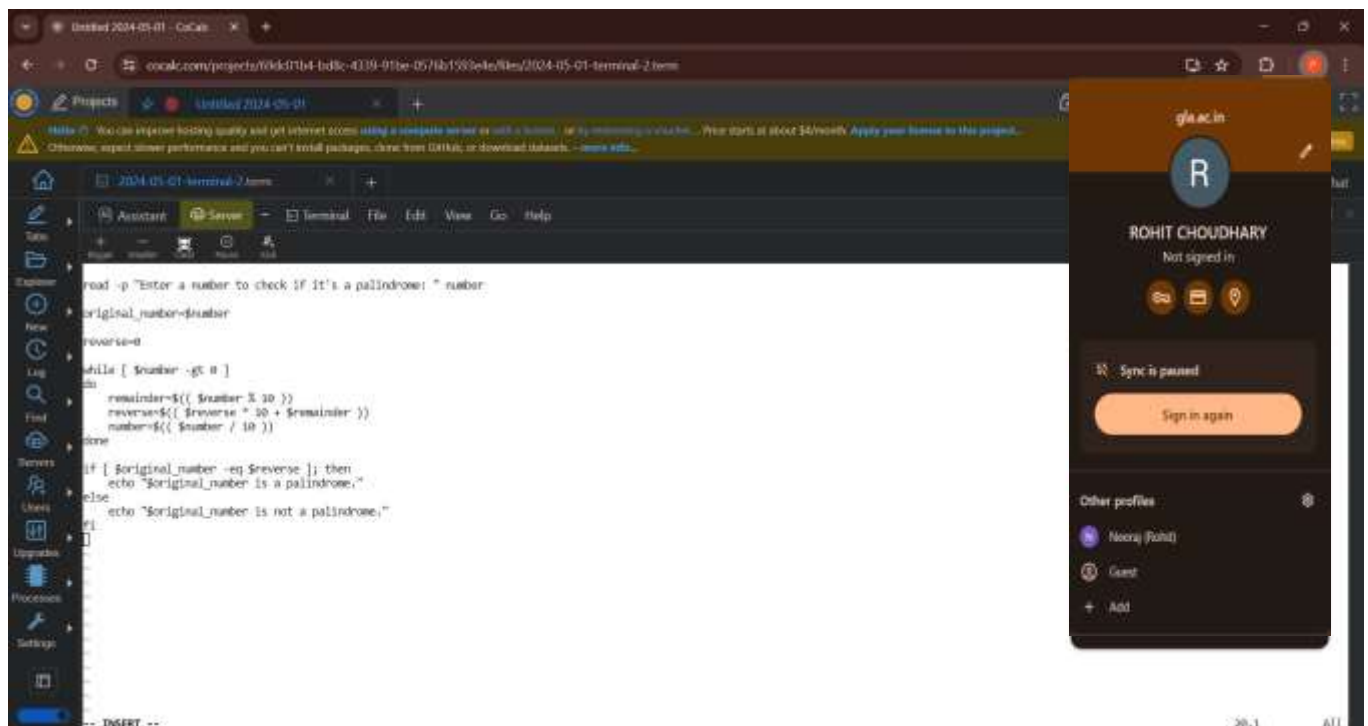
```
$ vi smalllarge
$ chmod 755 smalllarge
$ ./smalllarge
Enter the list of numbers separated by spaces: 5 9 3 4 8 0
Smallest number: 0
Largest number: 9
$ vi fibo
$ chmod 755 fibo
$ ./fibo
Enter the number of terms for Fibonacci series: 5
Fibonacci series up to 5 terms:
0 1 1 2 3
$
```



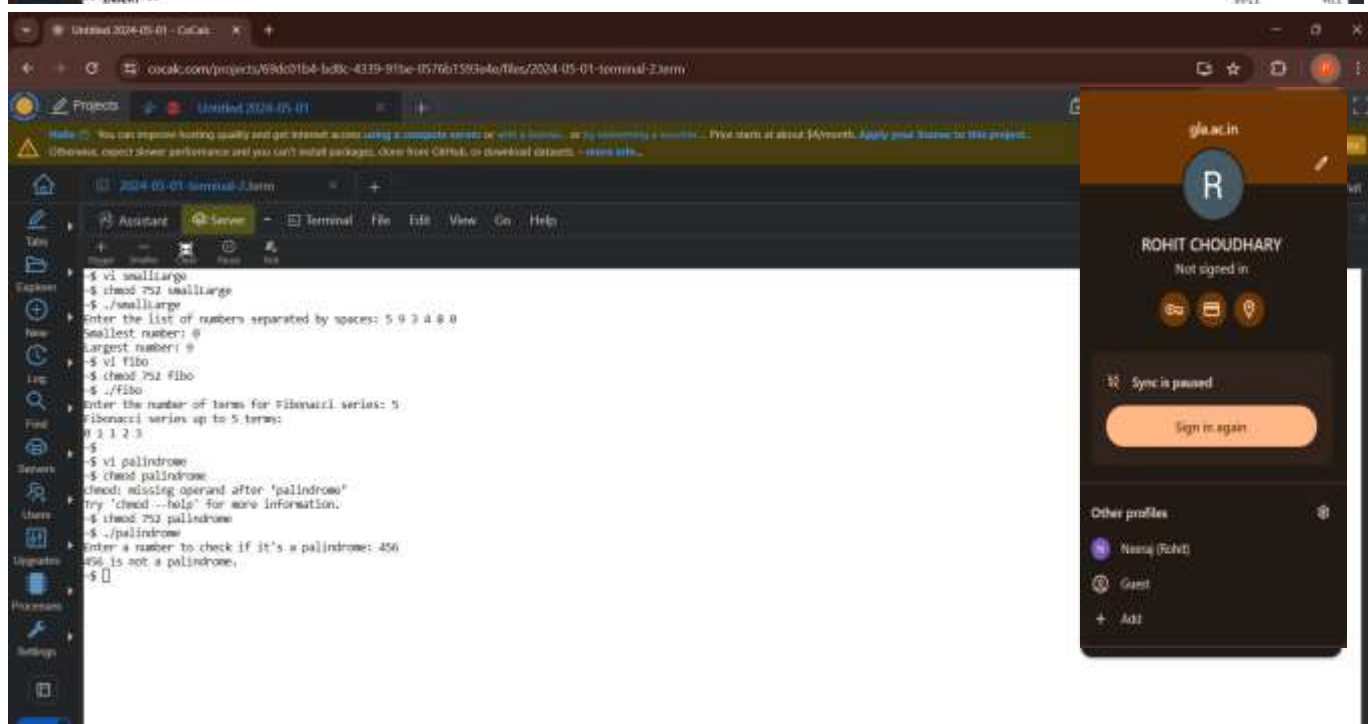
The screenshot shows a VS Code terminal window with the following commands and output:

```
read -p "Enter the number of terms for Fibonacci series: " num
n=0
echo "Fibonacci series up to $num terms:"
for ((i=0; i<num; i++)); do
  echo -n "$a "
  fn=$((a + b))
  a=$b
  b=$fn
done
echo
```


Program to check Palindrome number

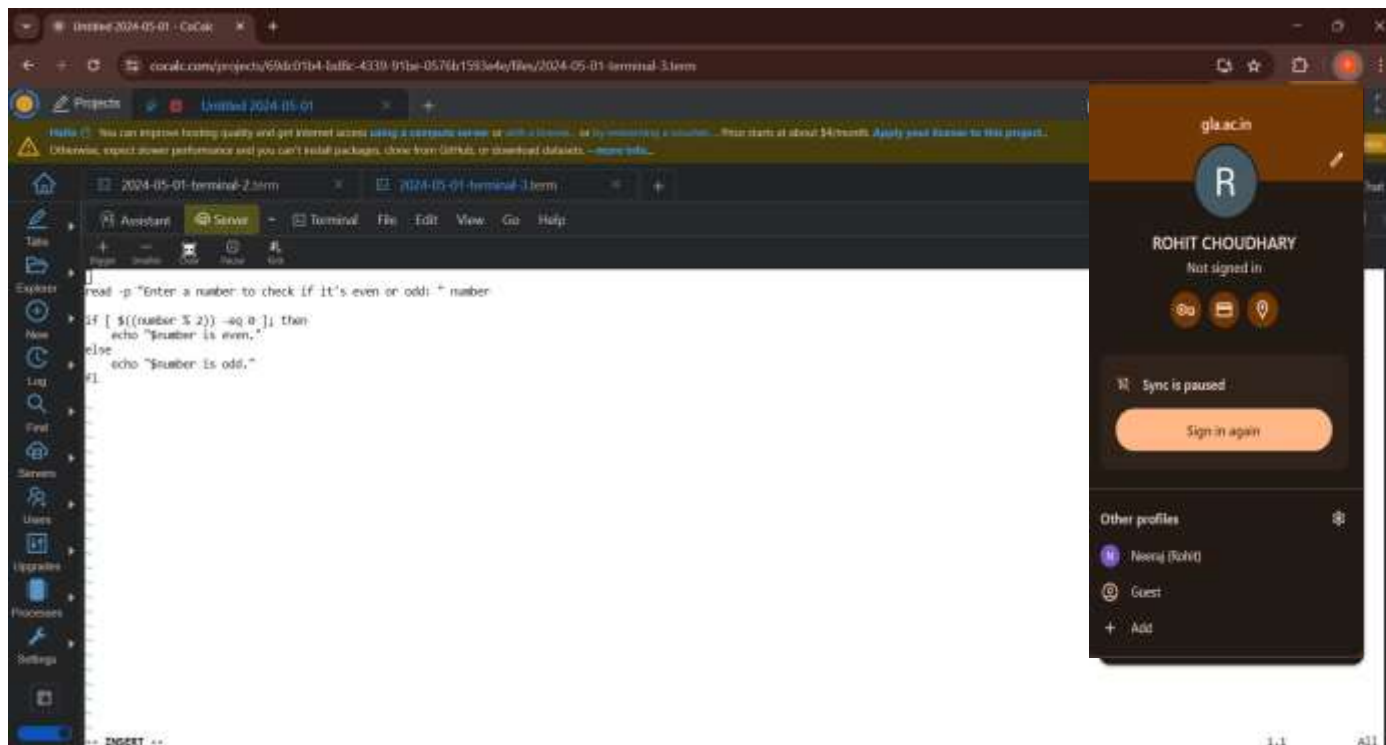


```
read -p "Enter a number to check if it's a palindrome: " number
original_number=$number
reverse=0
while [ $number -gt 0 ]
do
    remainder=$(( $number % 10 ))
    reverse=$(( $reverse * 10 + $remainder ))
    number=$(( $number / 10 ))
done
if [ $original_number -eq $reverse ]; then
    echo "$original_number is a palindrome."
else
    echo "$original_number is not a palindrome."
fi
```



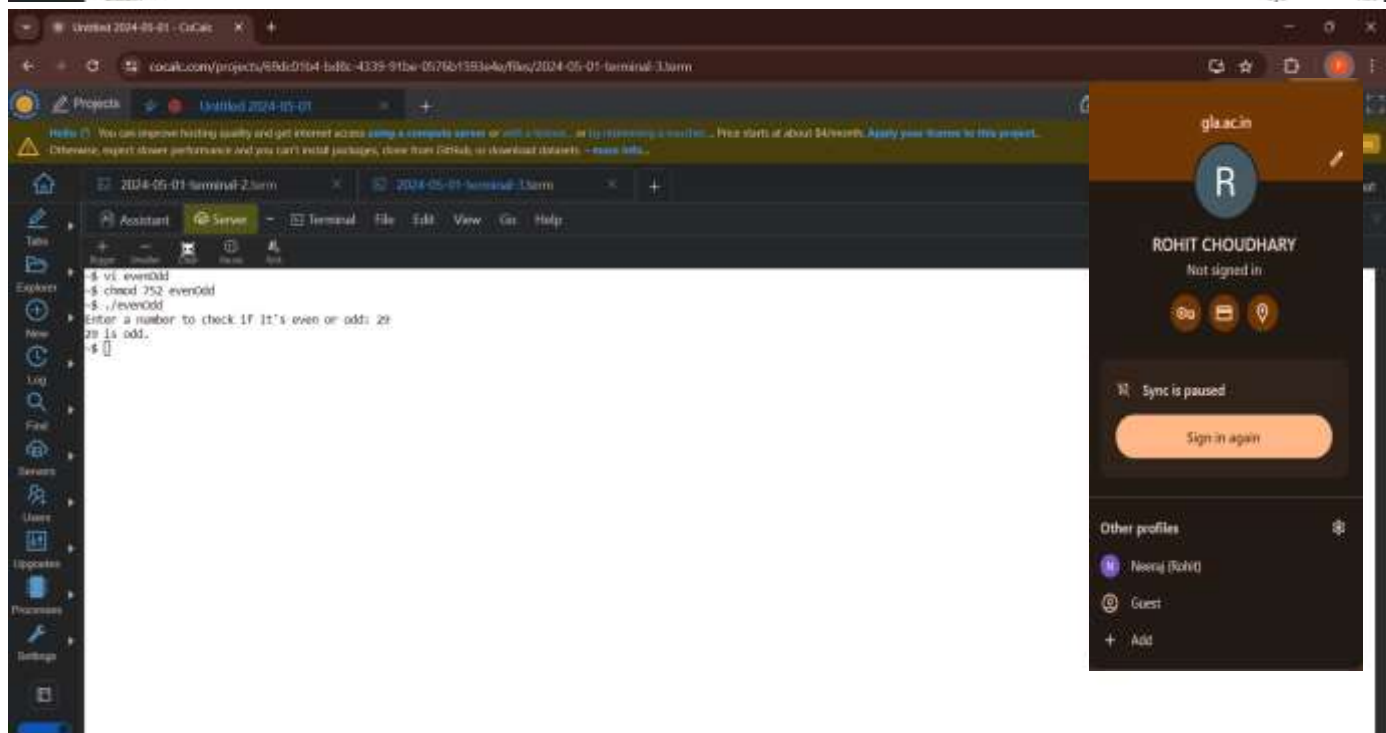
```
$ vi smallalgo
$ chmod 755 smallalgo
$ ./smallalgo
Enter the list of numbers separated by spaces: 5 9 3 4 0
Smallest number: 0
Largest number: 9
$ vi fibo
$ chmod 755 fibo
$ ./fibo
Enter the number of terms for Fibonacci series: 5
Fibonacci series up to 5 terms:
0 1 1 2 3
$
$ vi palindrome
$ chmod 755 palindrome
$ ./palindrome
Enter a number to check if it's a palindrome: 456
456 is not a palindrome.
$
```

Program to check even or odd



The screenshot shows a code editor with a dark theme. The top bar indicates the project is 'Untitled 2024-05-01 - CoCac'. The left sidebar shows a file explorer with a '2024-05-01-terminal-3.terminal' file selected. The main editor area contains a shell script:

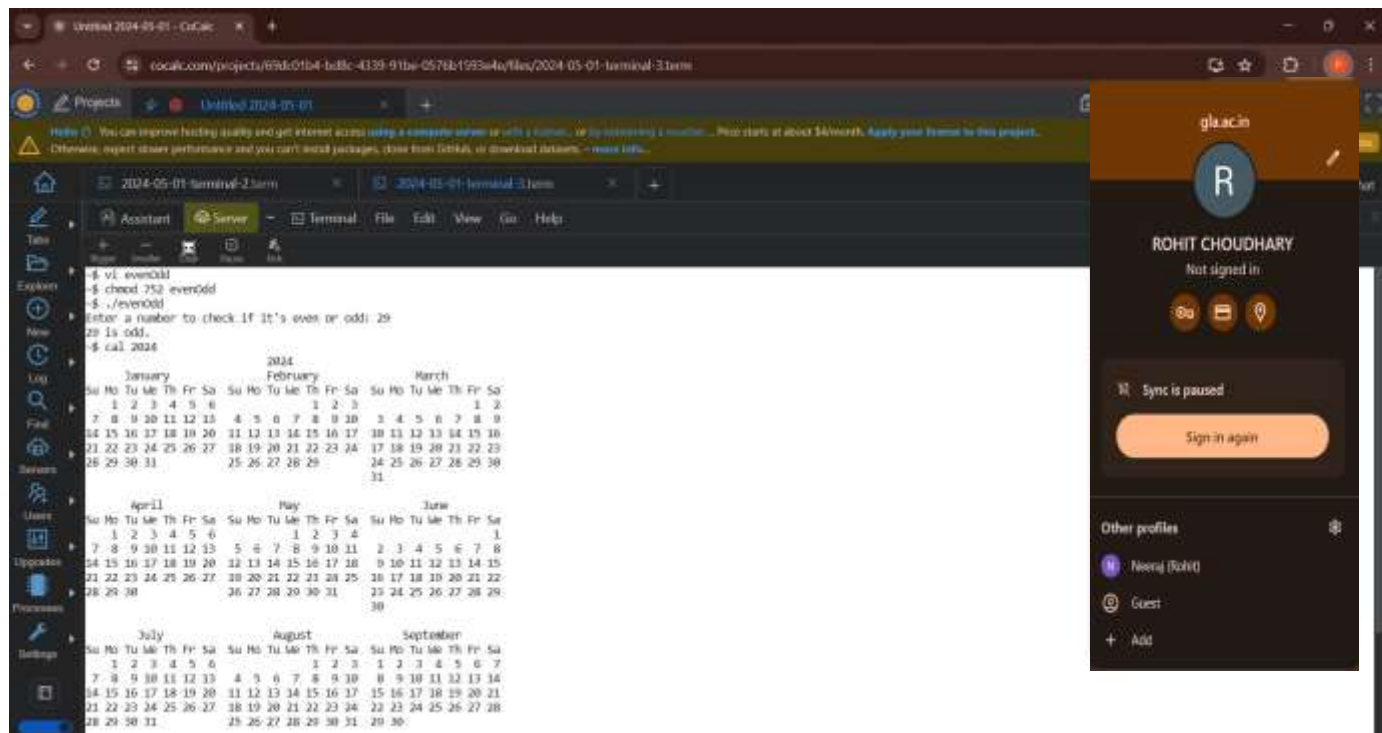
```
read -p "Enter a number to check if it's even or odd: " number
if [ $(($number % 2)) -eq 0 ]; then
    echo "$number is even."
else
    echo "$number is odd."
fi
```



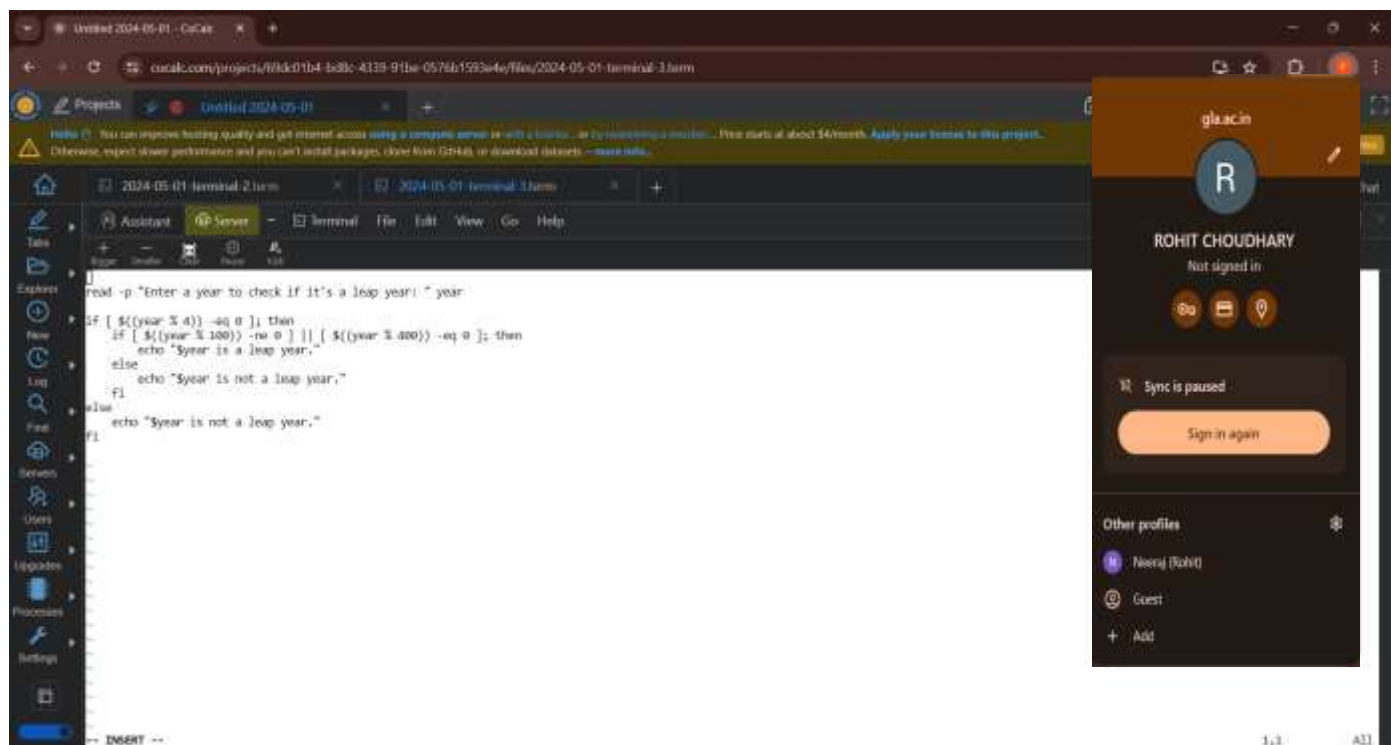
The screenshot shows the same code editor as above, but now the script has been executed. The terminal output is visible:

```
$ vi evenOdd
$ chmod 755 evenOdd
$ ./evenOdd
Enter a number to check if it's even or odd: 29
29 is odd.
$
```

To print calendar of 2024



Program to check if a year is leap or not



The image shows a screenshot of a VS Code editor window. The main editor area displays a shell script for checking if a year is a leap year. The script uses a `read` command to prompt the user for a year, followed by a series of `if` statements to check divisibility by 4, 100, and 400. The output of the script is displayed in the terminal. On the right side of the editor, there is a sidebar showing a user profile for 'ROHIT CHOUDHARY' with a 'Sign in again' button. The sidebar also lists other profiles: 'Neesaj (Rohit)', 'Guest', and an 'Add' button.

```
read -p "Enter a year to check if it's a leap year: " year
if [ ${year%0} -eq 0 ]; then
    if [ ${year%00} -eq 0 ]; then
        echo "Year is a leap year."
    else
        echo "Year is not a leap year."
    fi
else
    echo "Year is not a leap year."
fi
```