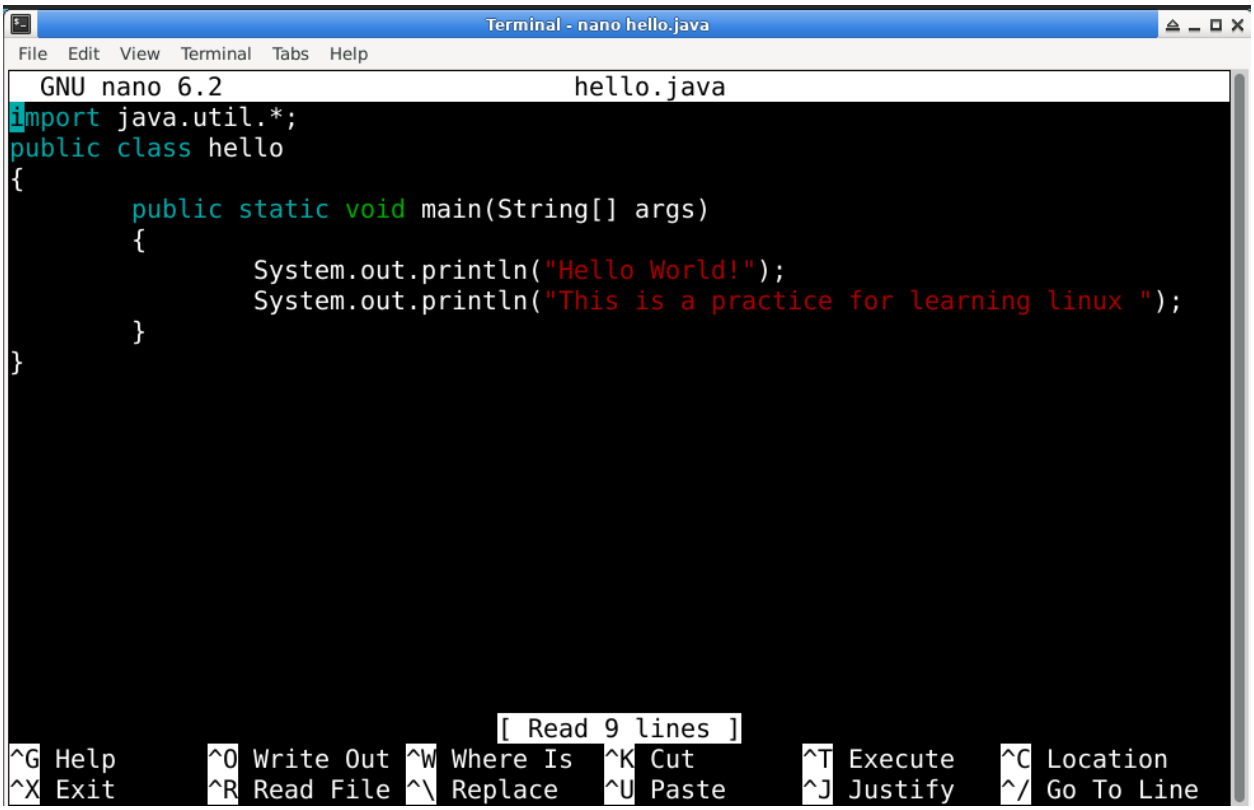
DAY 2 :
SHIVA RAMALINGAM V ECE

PROGRAMMING IN JAVA USING LINUX TERMINAL:



```
Terminal - labex@66e2a2ad4479d21feb6d5bf5:~/project
File  Edit  View  Terminal  Tabs  Help
labex:project/ $ nano hello.java
labex:project/ $ nano hello.java
labex:project/ $ cat hello.java
import java.util.*;
public class hello
{
        public static void main(String[] args)
        {
                System.out.println("Hello World!");
                System.out.println("This is a practice for learning linux ");
        }
}
labex:project/ $ javac hello.java
labex:project/ $ java hello
Hello World!
This is a practice for learning linux
```

PROGRAM IN C USING LINUX TERMINAL:

```
Terminal - nano cprogramming.c
File  Edit  View  Terminal  Tabs  Help
  GNU nano 6.2                    cprogramming.c *
#include<stdio.h>

int main()
{
        int n;
        printf("Enter a number:");
        scanf("%d",&n);
        if(n%2==0)
        {
                printf("Even");
        }
        else
        {
                printf("Odd");
        }
        return 0;
}
```

```
Terminal - labex@66e2a2ad4479d21feb6d5bf5:~/project
File  Edit  View  Terminal  Tabs  Help
labex:project/ $ nano hello.java
labex:project/ $ nano hello.java
labex:project/ $ cat hello.java
import java.util.*;
public class hello
{
        public static void main(String[] args)
        {
                System.out.println("Hello World!");
                System.out.println("This is a practice for learning linux ");
        }
}
labex:project/ $ javac hello.java
labex:project/ $ java hello
Hello World!
This is a practice for learning linux
labex:project/ $ nano cprogramming.c
labex:project/ $ gcc cprogramming.c -o print
labex:project/ $ ./print
Enter a number:5
Odd
labex:project/ $
```

SHELL PROGRAM :

```
root@a0f429dbfafe:/# nano shellp.sh
root@a0f429dbfafe:/# ./shellp.sh
bash: ./shellp.sh: Permission denied
root@a0f429dbfafe:/# chmod +x shellp.sh
root@a0f429dbfafe:/# ./shellp.sh
./shellp.sh: line 1: Hello: command not found
root@a0f429dbfafe:/# nano shellp.sh
root@a0f429dbfafe:/# ./shellp.sh
Hello All , Shell Programming !.
root@a0f429dbfafe:/#
```

LINUX COMMANDS :

1. pwd – Displays the current working directory.
2. ls – Lists files and directories in the current directory.
3. cd [directory] – Changes the current directory to the specified one.
4. mkdir [directory] – Creates a new directory.
5. rm [file] – Removes a file.
6. rm -r [directory] – Recursively removes a directory and its contents.
7. cp [source] [destination] – Copies files or directories.
8. mv [source] [destination] – Moves or renames files or directories.
9. cat [file] – Displays the contents of a file.
10. nano [file] – Opens a file in the nano text editor.
11. touch [file] – Creates a new, empty file.
12. chmod +x [file] – Grants execute permission to a file.
13. chown [user] [file] – Changes the owner of a file.
14. ps – Displays currently running processes.

GIT COMMANDS :

1. git init – Initializes a new Git repository.
2. git clone [url] – Clones a repository from a remote URL.
3. git status – Displays the status of the working directory and staging area.
4. git add [file] – Adds a file to the staging area.
5. git add . – Stages all changes (new, modified, and deleted files).
6. git commit -m "[message]" – Commits staged changes with a message.
7. git push [remote] [branch] – Pushes commits to a remote repository.
8. git pull [remote] [branch] – Fetches and merges changes from the remote branch.
9. git fetch [remote] – Downloads changes from the remote repository without merging.
10. git merge [branch] – Merges a branch into the current branch.
11. git checkout [branch] – Switches to the specified branch.
12. git branch – Lists all branches in the repository.
13. git branch [branch-name] – Creates a new branch.
14. git branch -d [branch-name] – Deletes a branch.
15. git log – Shows a log of all commits.