|  |  |  |
| --- | --- | --- |
| **++++Sr. No.** | **Contents** | **Page No.** |
| **1** | **Annexure I– Micro Project Proposal** |  |
| 1.Aims/Benefits of the Micro-Project |  |
| 2. Course Outcome Addressed |  |
| 3.Proposed Methodology |  |
| 4. Action Plan |  |
| 5. Resources Required |  |
| 6. Name of Team Members with Roll No.’s |  |
| **2** | **Annexure II – Micro Project Report** |  |
| 1.Rationale |  |
| 2.Aims/Benefits of the Micro-Project |  |
| 3.Course Outcome Achieved |  |
| 4. Literature Review |  |
| 5.Actual Methodology Followed |  |
| 5.1 Flow chart |  |
| 5.2 Source code |  |
| 6.Actual Resources Used |  |
| 7.Outputs of Micro-Projects |  |
| 8. Skill developed / Learning out of this Micro-Project |  |
| 9. Applications of this Micro-Project |  |

**Annexure I**

**Micro Project Proposal**

**Use of decision making statements**

**1. Aims/Benefits of the Micro-Project:**

Decision making statements allow you to decide the order of execution of specific statements in your program.

**2. Course Outcome Addressed:**

**Basic knowledge:** An ability to apply knowledge of basic mathematics, science and engineering to solve the engineering problems.

**Engineering tools:** Apply appropriate Computer Technology related techniques/ tools with an understanding of the limitations.

**Individual and team work:** Function effectively as a leader and team member in diverse/ multidisciplinary teams.

**3. Proposed Methodology:**

* In the case of decision control statements in C language (nested if and if-else), a group of available statements will get executed in case the conditions we have are true. The execution of the else part statements won't occur whenever these available conditions happen to be false
* Decision making statements allow you to decide the order of execution of specific statements in your program. You can set up a condition and tell the compiler to take a particular action if the condition is met. In case the condition is not met, you can instruct the compiler to execute a different block of code

**4. Action Plan**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr.**  **No.** | **Details of Activity** | **Planned**  **Start date** | **Planned**  **Finish date** | **Name of Responsible Team Members** |
| 1 | Search the topic | 13/02/2023  4:30pm-5:30pm | 14/02/2023  4:30pm-5:30pm | Patel Riyan |
| 2 | Search the information | 20/02/2023  4:30pm-5:30pm | 21/02/2023  4:30pm-5:30pm | Swami Malhar |
| 3 | Algorithm developing | 28/02/2023  4:30pm-5:30pm | 06/03/2023  4:30pm-5:30pm | Kore Vaibhav |
| 4 | Flowchart developing | 07/03/2023  4:30pm-5:30pm | 13/03/2023  4:30pm-5:30pm | Kore Vaibhav |
| 5 | Function making | 14/03/2023  4:30pm-5:30pm | 27/03/2023  4:30pm-5:30pm | Kadam Nikhil |
| 6 | Coding developing | 28/03/2023  4:30pm-5:30pm | 03/04/2023  4:30pm-5:30pm | Mashale Soham |
| 7 | Debugging | 10/04/2023  4:30pm-5:30pm | 11/04/2023  4:30pm-5:30pm | Mashale Soham |
| 8 | Finalizing Project with its report | 17/04/2023  4:30pm-5:30pm | 18/04/2023  4:30pm-5:30pm | Mashale Soham |

**5. Resources Required:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr.**  **No.** | **Name of resource / material** | **Specification** | **Quantity** | **Remarks** |
| 1 | Computer | WINDOWS 7,2GB RAM, 160GB HDD | 1 |  |
| 2 | Operating System | WINDOWS 7 | 1 |  |
| 3 | Compiler | Turbo C/GCC | 1 |  |
| 4 | Browser | Chrome | 1 |  |

**Names of Team Members with Roll No.’s:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr.**  **No.** | **Enrollment No.** | **Name of Team Member** | **Roll No.** |
| 1 | 2210950166 | Mashale Soham Kiran | 62 |
| 2 | 2210950173 | Patel Riyan Mahmmadrasul | 69 |
| 3 | 2210950172 | Swami Malhar Sanjivkumar | 68 |
| 4 | 2210950155 | Kadam Nikhil Dilip | 51 |
| 5 | 2210950157 | Kore Vaibhav Nagnath | 53 |

**Mr. Omkare R.S.**

**Name and Signature of the Teacher**

**Annexure – II**

**Micro-Project Report**

**Use of decision making statements**

**1. Rationale:**

Decision making statements allow you to decide the order of execution of specific statements in your program.

1. If
2. Else if
3. Nested if else
4. Else if ladder

**2. Aims/Benefits of the Micro-Project:**

Information gathering, evaluating the future consequences of present actions, determining alternative solutions

**3. Course Outcomes Achieved:**

**Basic knowledge:** An ability to apply knowledge of basic mathematics, science andengineering to solve engincering problems.

**Discipline knowledge:** Apply Basic Electronics engineering knowledge to solve broad based Computer Engineering related problems.

**Individual and team work:** Function effectively as a leader and team member indiverse / multidisciplinary teams.

**4. Literature Review**

1. **IF**

if statement supports two-way branching statement and multi-way branching statement.

* **Syntax of If Statement**

if(expression)

{

STMT

}

1. **Else if**

If the test expression is true, then the true-block statement(s), immediately following the if statements are executed first; otherwise, the false-block statement(s) are executed first.

* **Syntax of Else-If Statement**

if(test expression) {

true-block statement(s)

}

else {

false-block statement(s)

}

another-statement

1. **Nested If-else Statement:**

* When multiple decisions are involved, we can use more than one if-else statement in nested form. In the flowchart below we can see:
* If condition-1 is false the statement-3 will be executed, and condition-1 is true then the control is transferred to condition-2.
* If condition-2 is true, statement-1 will be executed; otherwise, statement-2 will be evaluated and then the control is transferred to another block of statement.

**4.Else-if Ladder:**

There is another way of setting up if statement together when multi-way decisions are involved. A multi-way decision is a series of ifs in which the statement linked with each else statement is an if statement.

* **Syntax of Else-If lADDER Statement**

if(test expression) {

true-block statement

}

else if(test expression){

block of statement

}

else if(test expression){

block of statement

}

else {

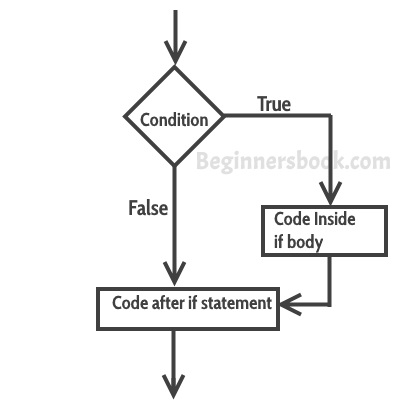
false-block statement

}

**5.Actual Methodology Followed**

* Describe the methodology used to gather information and data for the report.
* Specify the programming environment used (e.g., IDE, compiler).
* Explain the sample code used to demonstrate decision-making statements.
* Detail any specific scenarios or test cases considered.

**5.1 Flowchart**

****

****

**5.2 Source code**

**If**

**1.**

Int num = 5;

If (num > 0) {

Printf(“The number is positive.\n”);

}

**2.**

Int num = 6;

If (num % 2 == 0) {

Printf(“The number is even.\n”);

}

**3.**

Int num1 = 5, num2 = 5;

If (num1 == num2) {

Printf(“The numbers are equal.\n”);

}

**4.**

char c = 'a';

if (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u') {

printf("The character is a vowel.\n");

}

**5.**

int num = 10;

if (num > 5 && num < 15) {

printf("The number is between 5 and 15.\n");

}

**Else if**

int num = -5;

if (num > 0) {

printf(“The number is positive.\n”);

}

Else if (num < 0) {

Printf(“The number is negative.\n”);

}

Else {

Printf(“The number is zero.\n”);

}

**2.**

Int num = 0;

If (num == 0) {

Printf(“The number is zero.\n”);

}

Else if (num % 2 == 0) {

Printf(“The number is even.\n”);

}

Else {

Printf(“The number is odd.\n”);

}

**3.**

Int num = 15;

If (num % 3 == 0 && num % 5 == 0) {

Printf(“The number is a multiple of 3 and 5.\n”);

}

Else if (num % 3 == 0) {

Printf(“The number is a multiple of 3.\n”);

}

Else if (num % 5 == 0) {

Printf(“The number is a multiple of 5.\n”);

}

Else {

Printf(“The number is not a multiple of 3 or 5.\n”);

}

**4.**

Int num = -6;

If (num == 0) {

Printf(“The number is zero.\n”);

}

Else if (num > 0 && num % 2 == 0) {

Printf(“The number is positive and even.\n”);

}

Else if (num > 0 && num % 2 != 0) {

Printf(“The number is positive and odd.\n”);

}

Else if (num < 0 && num % 2 == 0) {

Printf(“The number is negative and even.\n”);

}

Else {

Printf(“The number is negative and odd.\n”);

}

**5.**

Int num = 24;

If (num % 4 == 0 && num % 6 == 0) {

Printf(“The number is a multiple of 4 and 6.\n”);

}

Else if (num % 4 == 0) {

Printf(“The number is a multiple of 4.\n”);

}

Else if (num % 6 == 0) {

Printf(“The number is a multiple of 6.\n”);

}

Else {

Printf(“The number is not a multiple of 4 or 6.\n”);

}

**Nested if else**

**1.**

Int num = 5;

If (num > 0) {

Printf(“The number is positive.\n”);

} else if (num < 0) {

Printf(“The number is negative.\n”);

} else {

Printf(“The number is zero.\n”);

}

**2.**

Int num = -6;

If (num > 0) {

If (num % 2 == 0) {

Printf(“The number is positive and even.\n”);

} else {

Printf(“The number is positive and odd.\n”);

}

} else if (num < 0) {

If (num % 2 == 0) {

Printf(“The number is negative and even.\n”);

} else {

Printf(“The number is negative and odd.\n”);

}

} else {

Printf(“The number is zero.\n”);

}

**3.**

Char c = ‘3’;

If (c >= ‘a’ && c <= ‘z’) {

If (c == ‘a’ || c == ‘e’ || c == ‘I’ || c == ‘o’ || c == ‘u’) {

Printf(“The character is a vowel.\n”);

} else {

Printf(“The character is a consonant.\n”);

}

} else {

Printf(“The character is not an alphabet.\n”);

}

**4.**

int year = 2024;

if (year % 4 == 0) {

if (year % 100 == 0) {

if (year % 400 == 0) {

printf(“%d is a leap year.\n”, year);

} else {

printf(“%d is not a leap year.\n”, year);

}

} else {

printf(“%d is a leap year.\n”, year);

}

} else {

printf(“%d is not a leap year.\n”, year);

}

**5.**

int marks = 55;

if (marks >= 40) {

if (marks >= 60) {

printf(“Congratulations, you have passed with first division.\n”);

} else if (marks >= 50) {

printf(“Congratulations, you have passed with second division.\n”);

} else {

printf(“Congratulations, you have passed.\n”);

}

} else if (marks >= 20) {

printf(“You have failed, but you are eligible to appear for re-examination.\n”);

} else {

printf(“You have failed.\n”);

}

**If else ladder**

**1.**

#include <stdio.h>

Int main() {

Int num;

Printf(“Enter a number: “);

Scanf(“%d”, &num);

If (num == 0) {

Printf(“The number is zero.\n”);

}

Else if (num > 0) {

Printf(“The number is positive.\n”);

}

Else {

Printf(“The number is negative.\n”);

}

Return 0;

}

**2.**

#include <stdio.h>

Int main() {

Int age;

Printf(“Enter your age: “);

Scanf(“%d”, &age);

If (age < 18) {

Printf(“You are a minor.\n”);

}

Else if (age >= 18 && age < 60) {

Printf(“You are an adult.\n”);

}

Else {

Printf(“You are a senior citizen.\n”);

}

Return 0;

}

**3.**

#include <stdio.h>

Int main() {

Int marks;

Printf(“Enter your marks: “);

Scanf(“%d”, &marks);

If (marks >= 90) {

Printf(“You got an A+ grade.\n”);

}

Else if (marks >= 80 && marks < 90) {

Printf(“You got an A grade.\n”);

}

Else if (marks >= 70 && marks < 80) {

Printf(“You got a B grade.\n”);

}

Else if (marks >= 60 && marks < 70) {

Printf(“You got a C grade.\n”);

}

Else {

Printf(“You failed the exam.\n”);

}

Return 0;

}

**4.**

#include <stdio.h>

Int main() {

Int num;

Printf(“Enter a number: “);

Scanf(“%d”, &num);

If (num % 2 == 0) {

Printf(“The number is even.\n”);

}

Else if (num % 2 != 0) {

Printf(“The number is odd.\n”);

}

Return 0;

}

**5.**

#include <stdio.h>

Int main() {

Int year;

Printf(“Enter a year: “);

Scanf(“%d”, &year);

If (year % 4 == 0) {

If (year % 100 == 0) {

If (year % 400 == 0) {

Printf(“%d is a leap year.\n”, year);

}

Else {

Printf(“%d is not a leap year.\n”, year);

}

}

Else {

Printf(“%d is a leap year.\n”, year);

}

}

Else {

Printf(“%d is not a leap year.\n”, year);

}

Return 0;

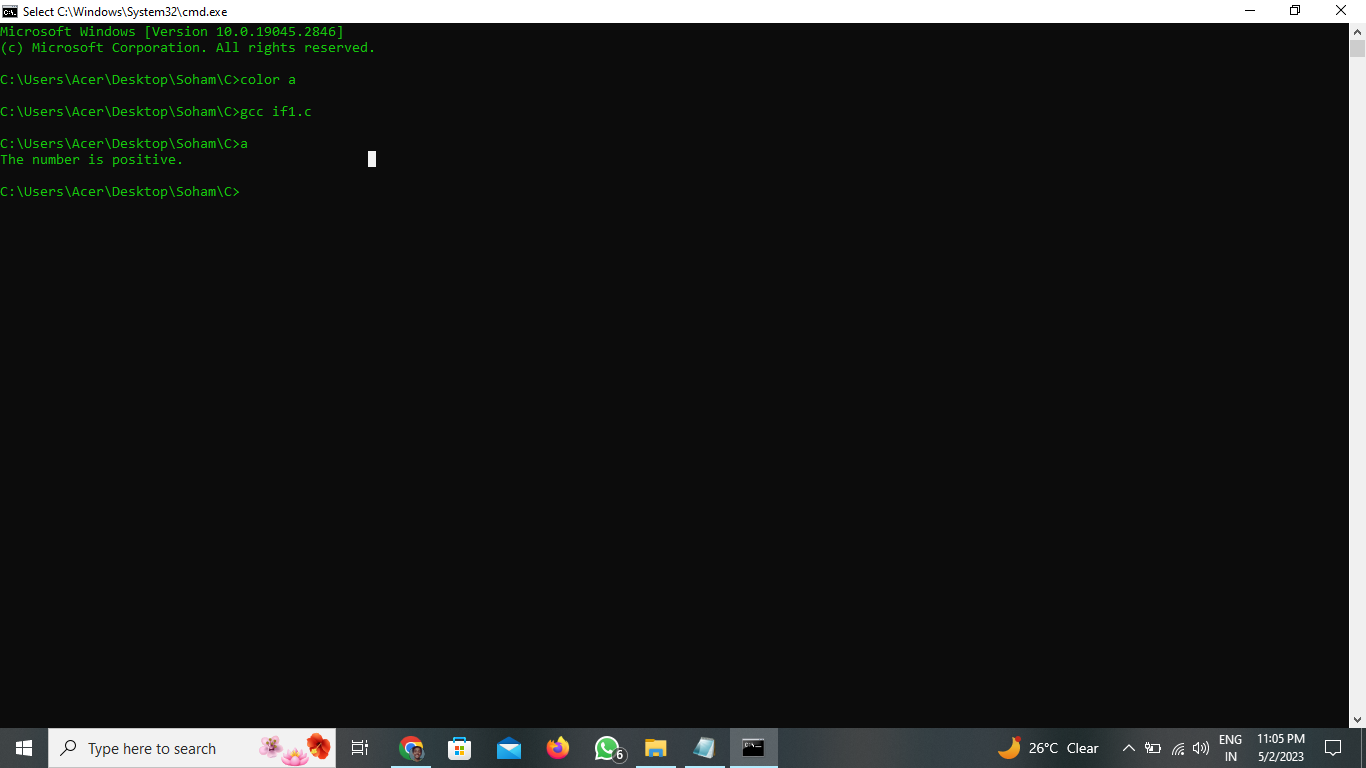
}

**6. Actual resources used.**

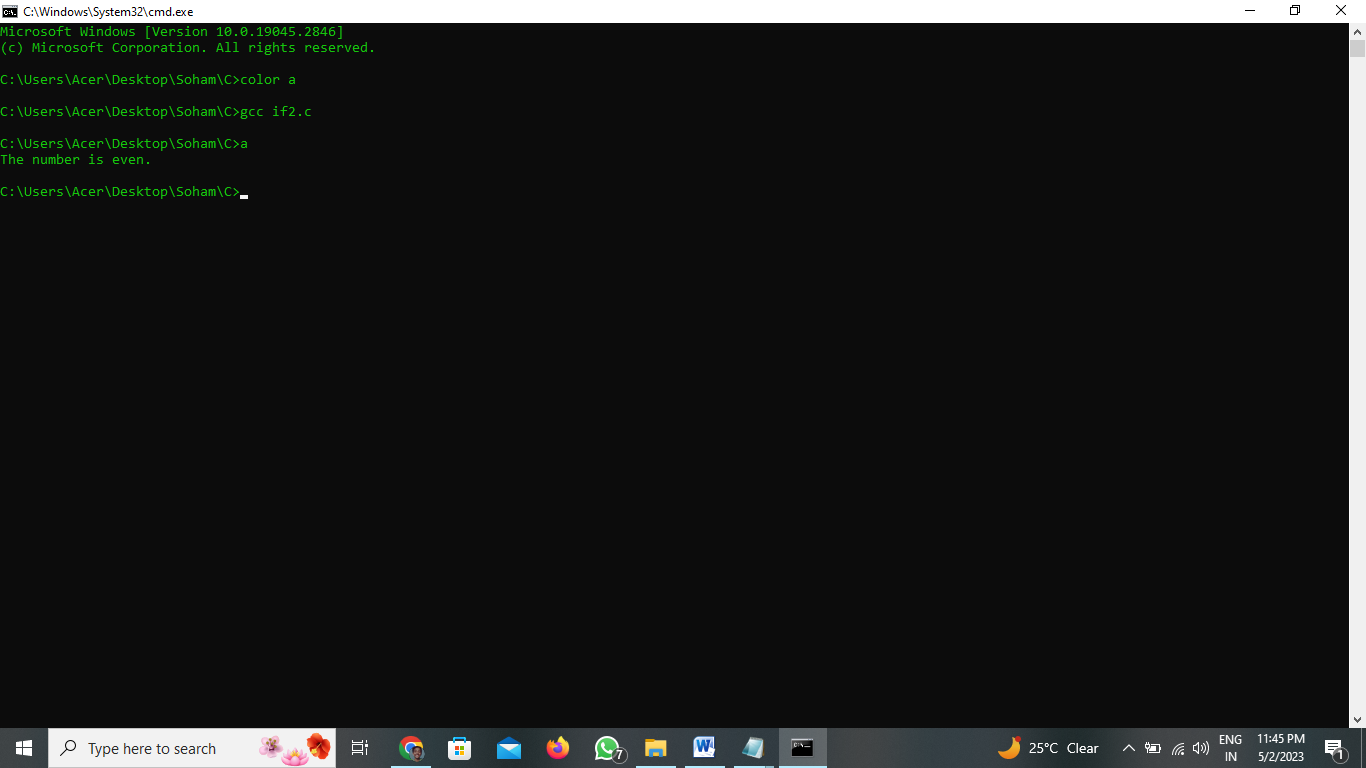
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr. No.** | **Name of resource / material** | **Specification** | **Quantity** | **Remarks** |
| 1 | Computer | WINDOWS 7, 2GB RAM, 160GB HDD | 1 |  |
| 2 | Operating System | WINDOWS 7 | 1 |  |
| 3 | Compiler | Turbo C/GCC | 1 |  |
| 4 | Browser | Chrome | 1 |  |

**7. Outputs of Microprojects.**

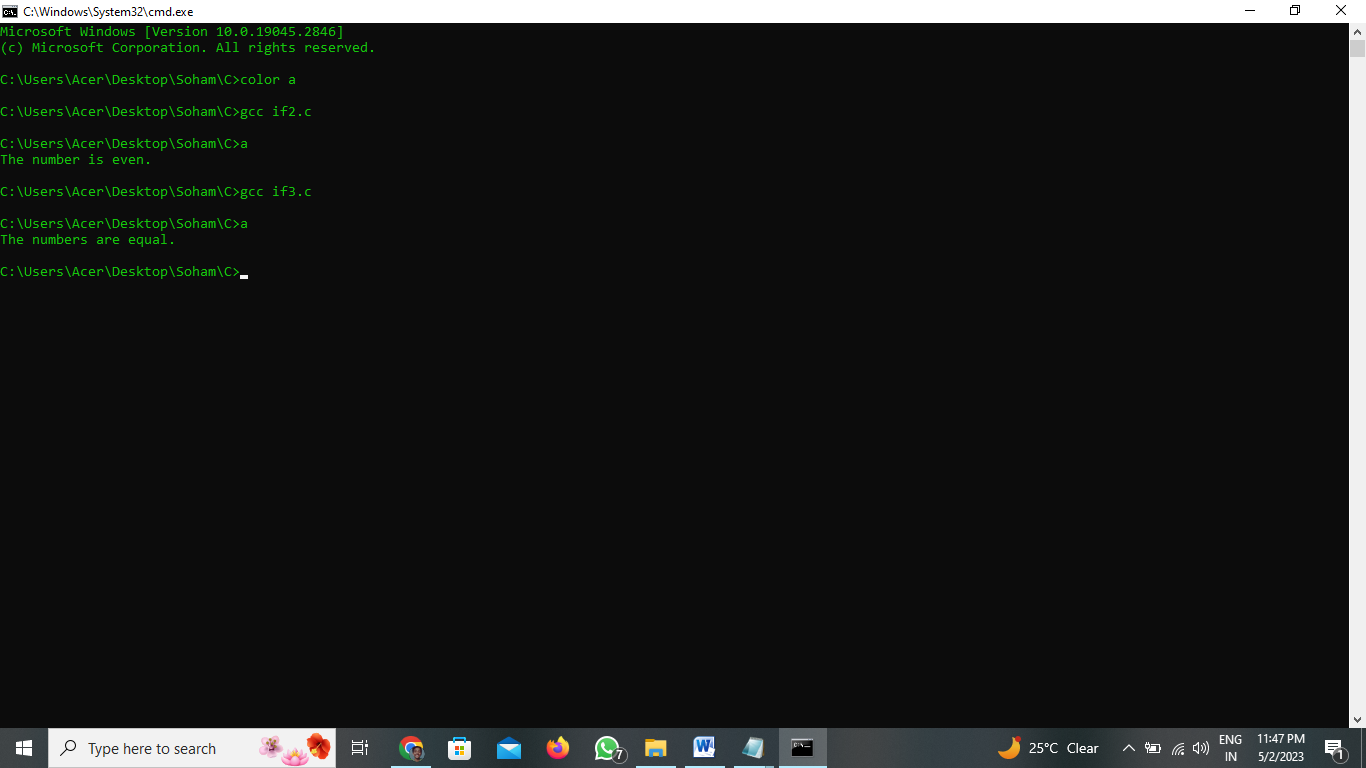
**If**

1

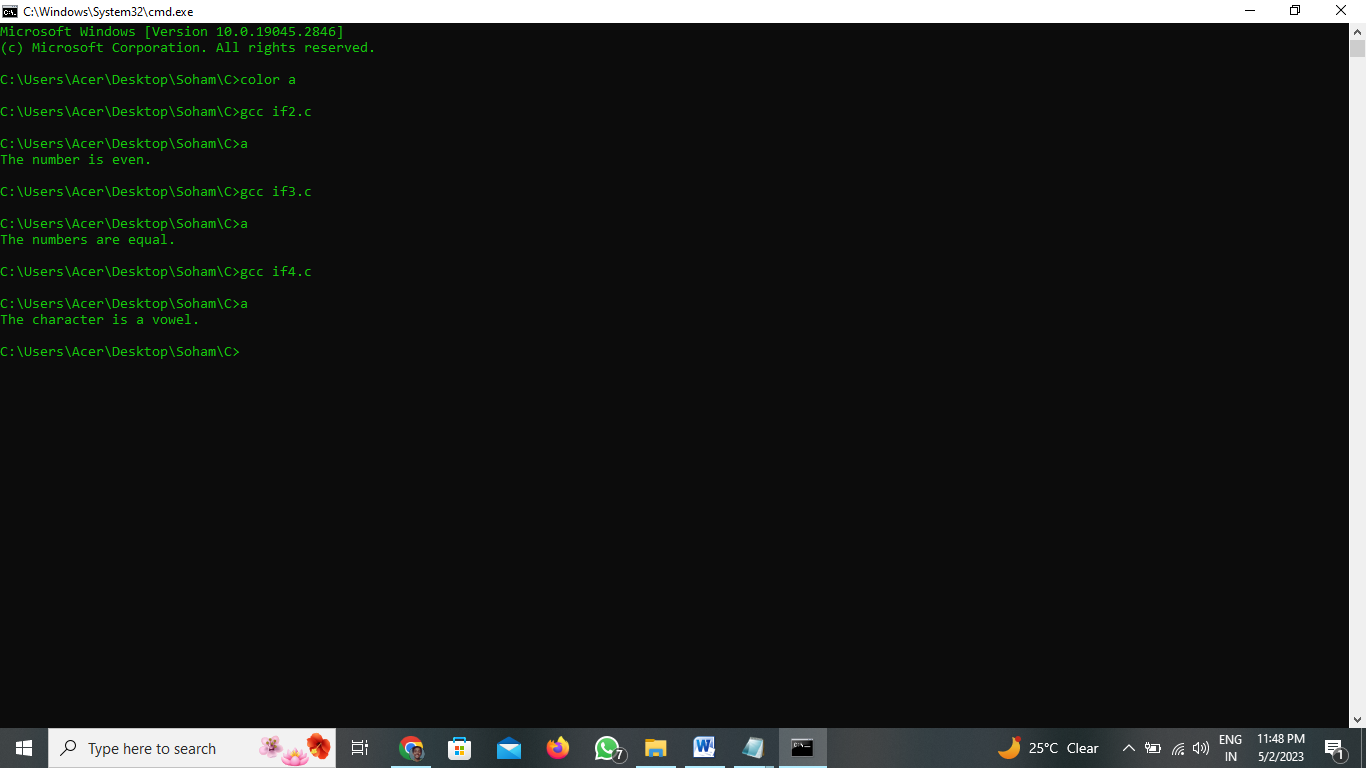
2



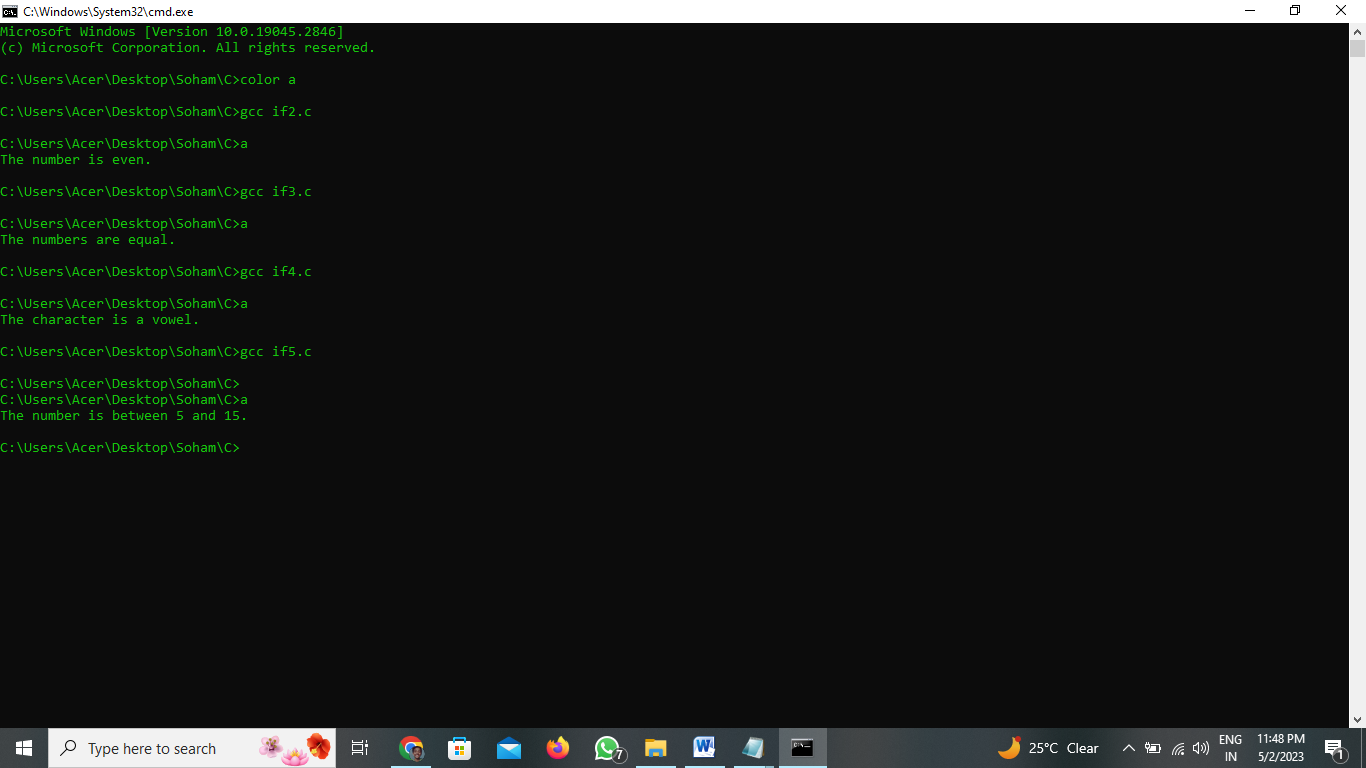
3



4

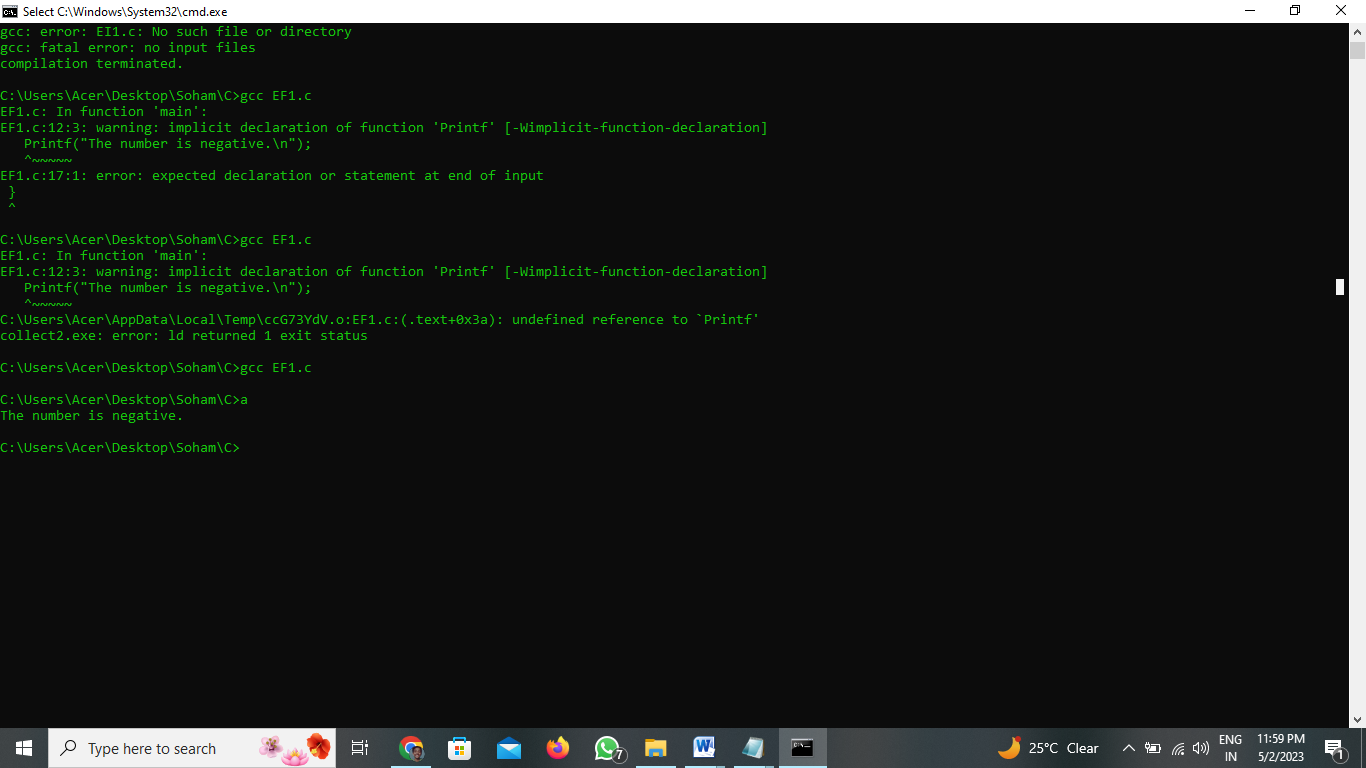


5

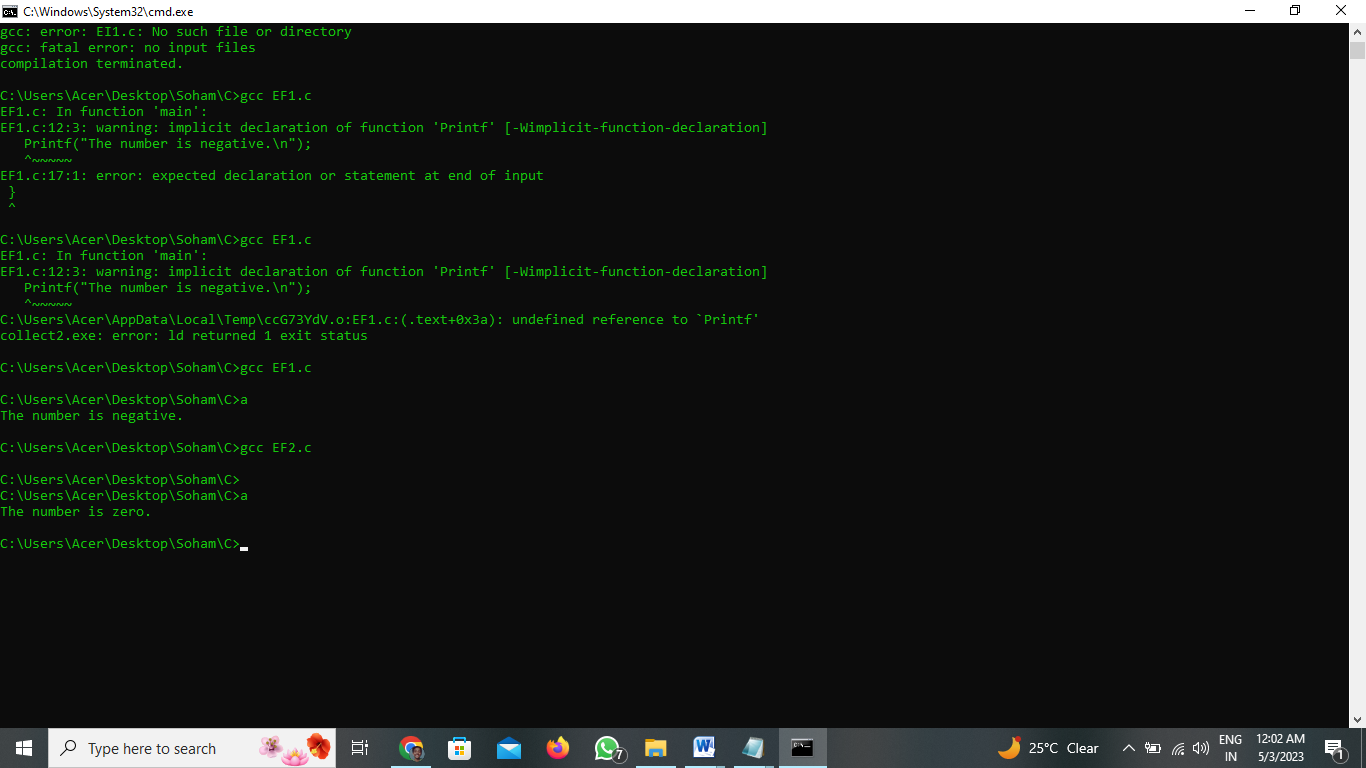


**Else if**

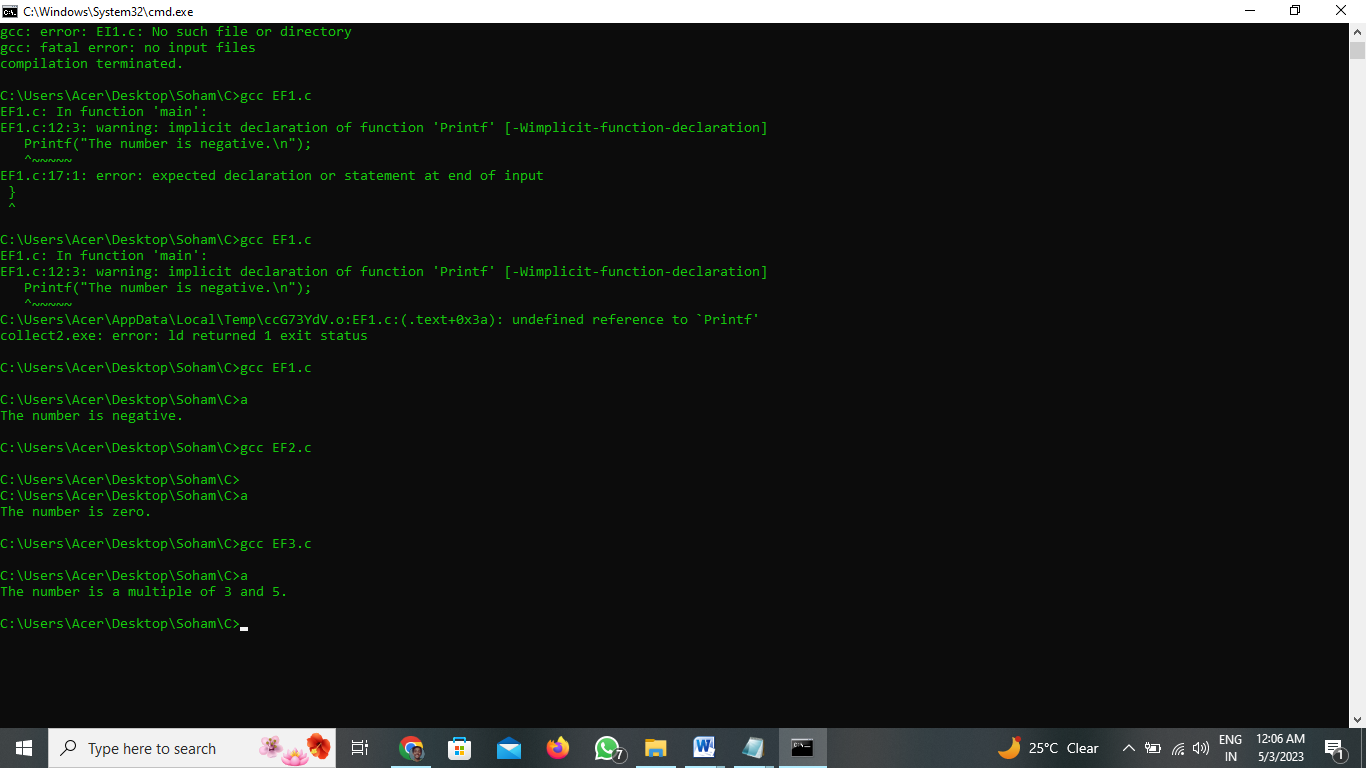
1



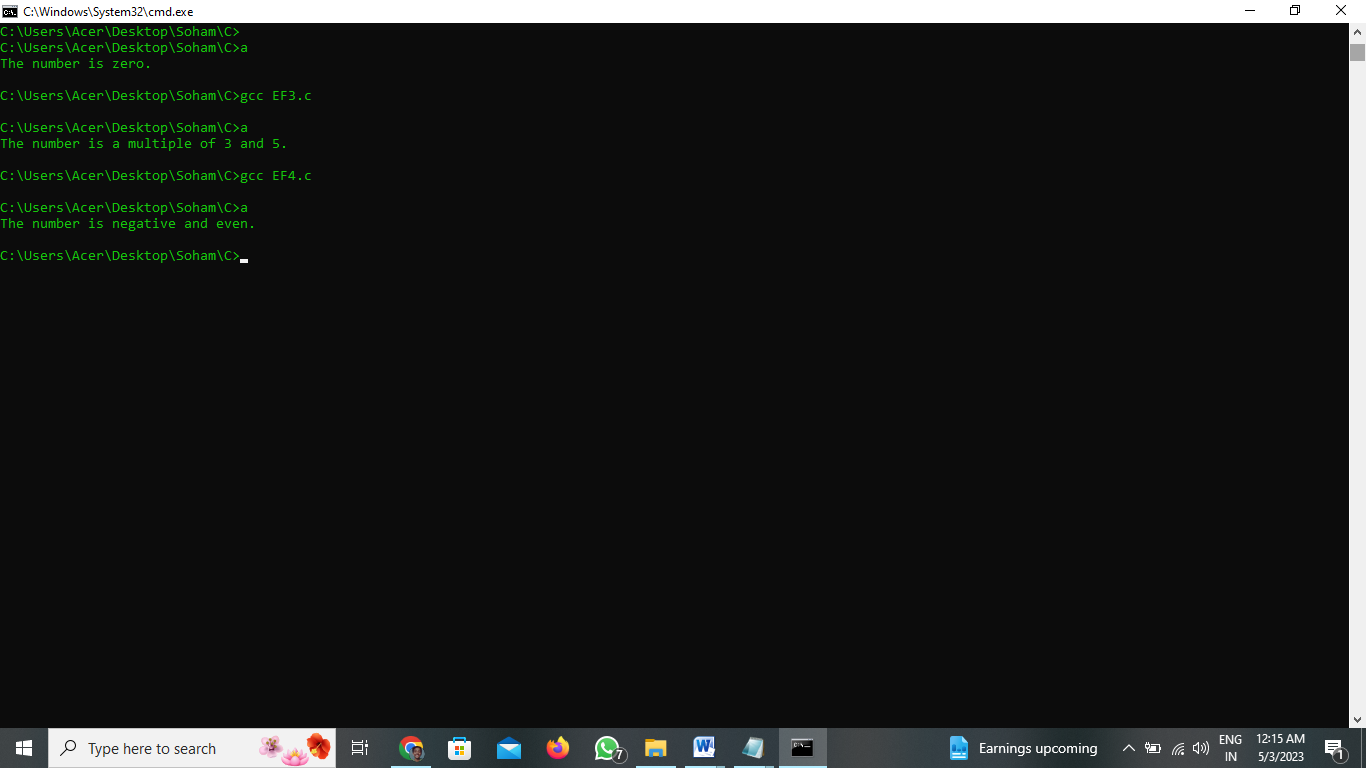
2



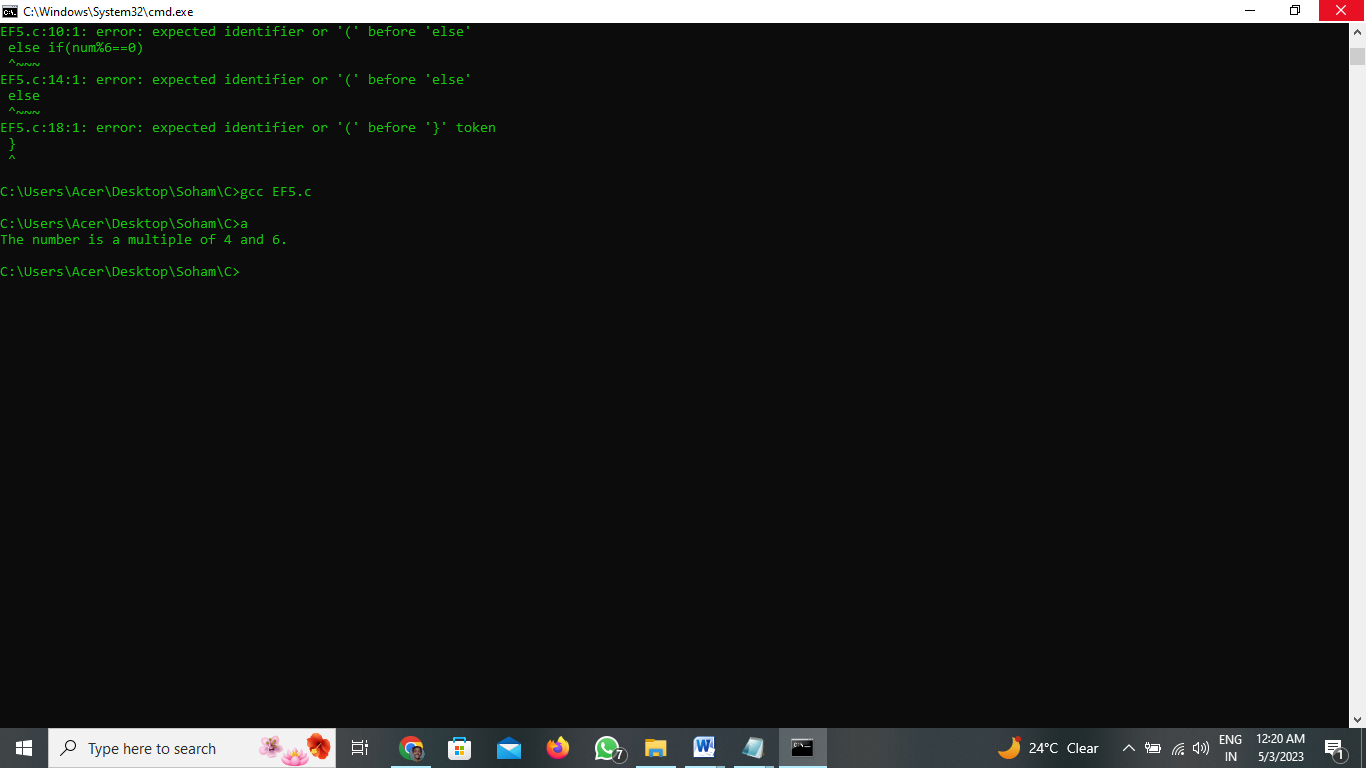
3



4

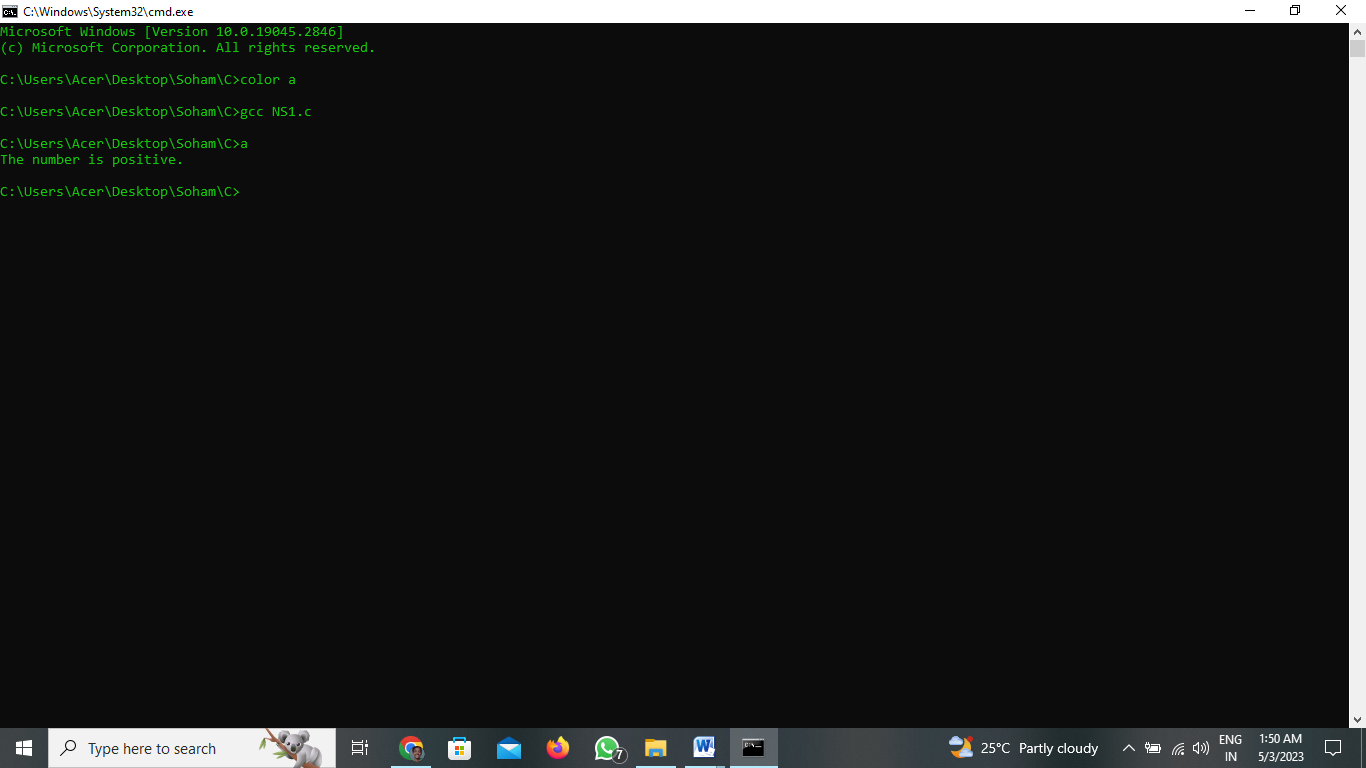


5

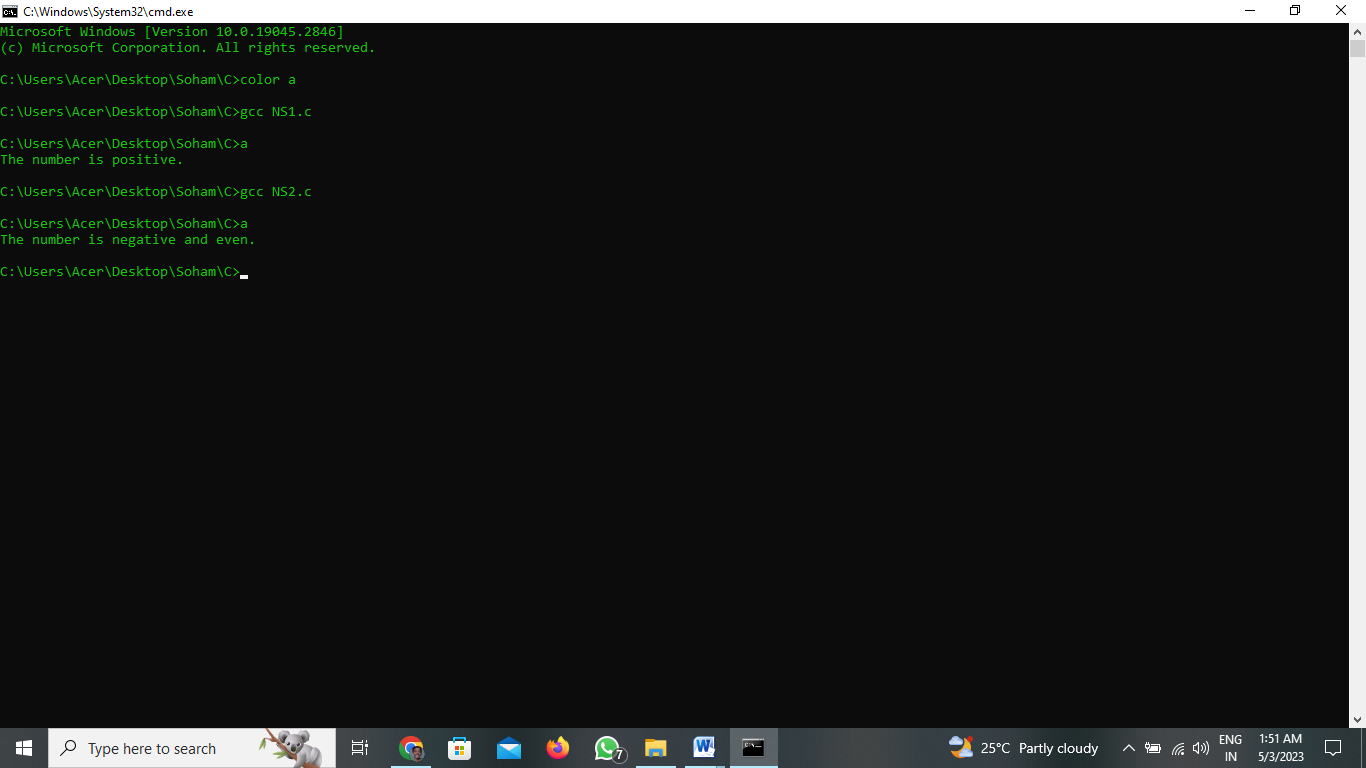


**Nested if else**

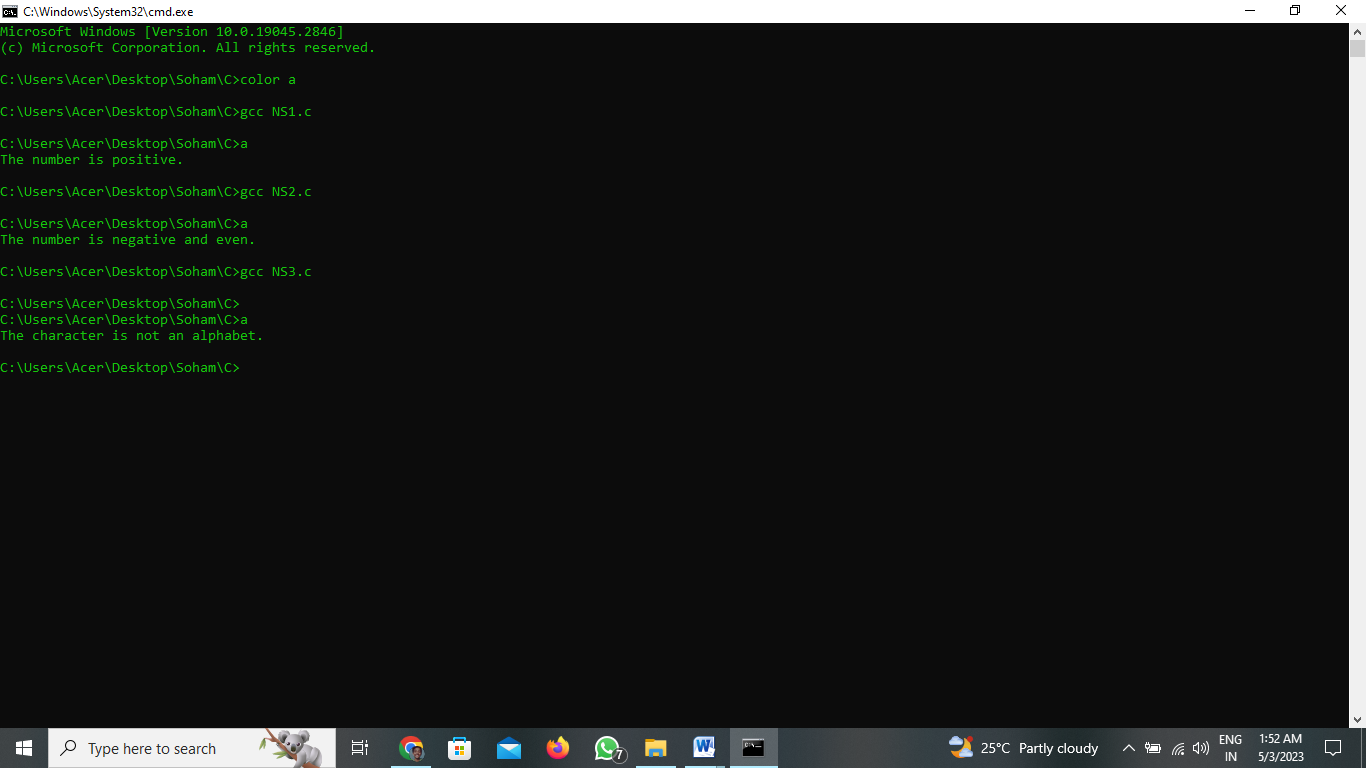
1



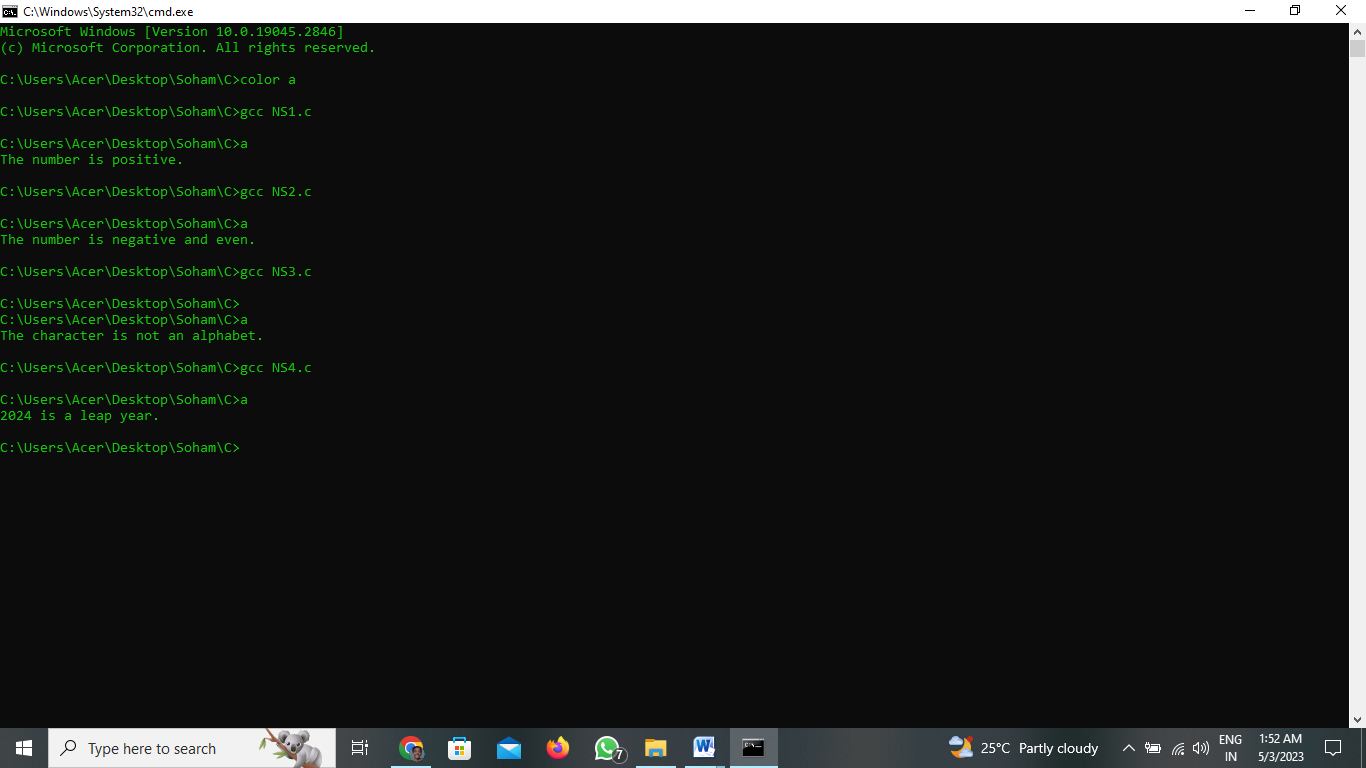
2



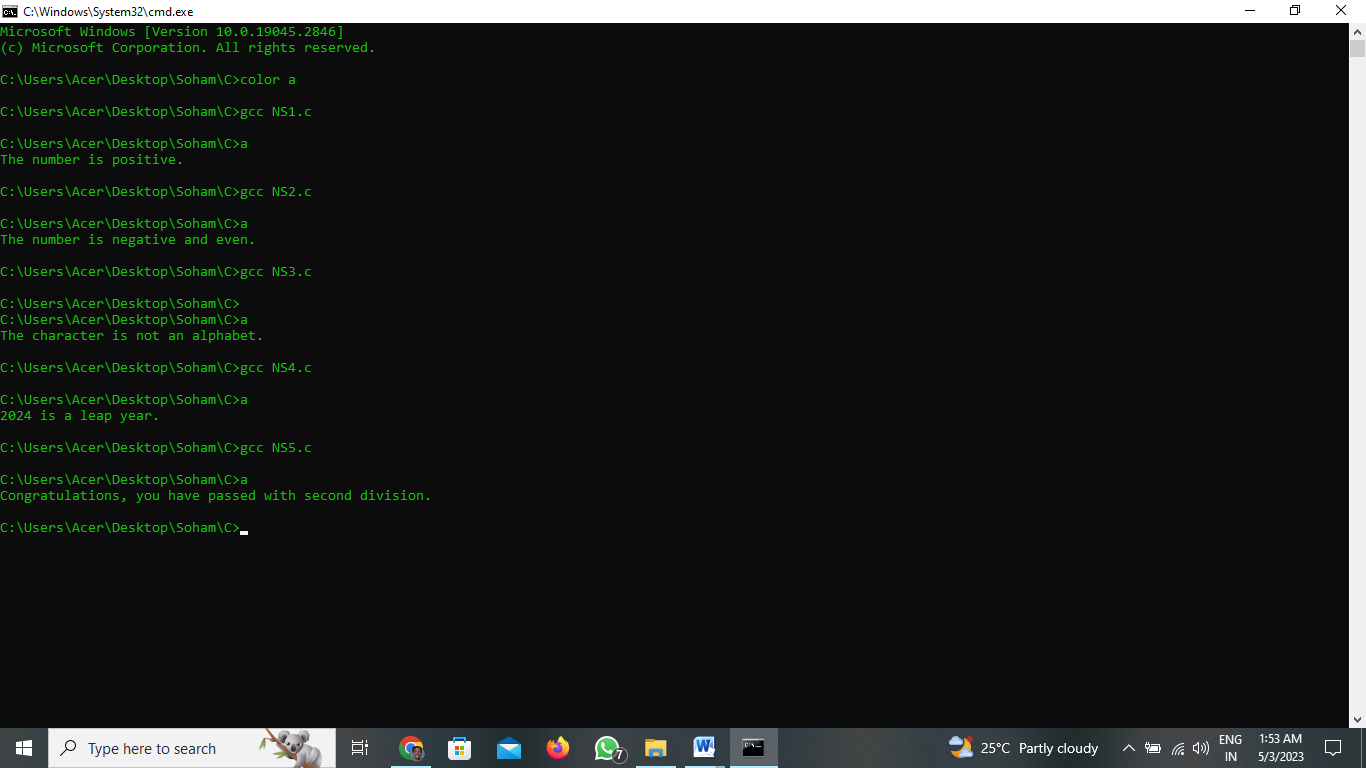
3



4

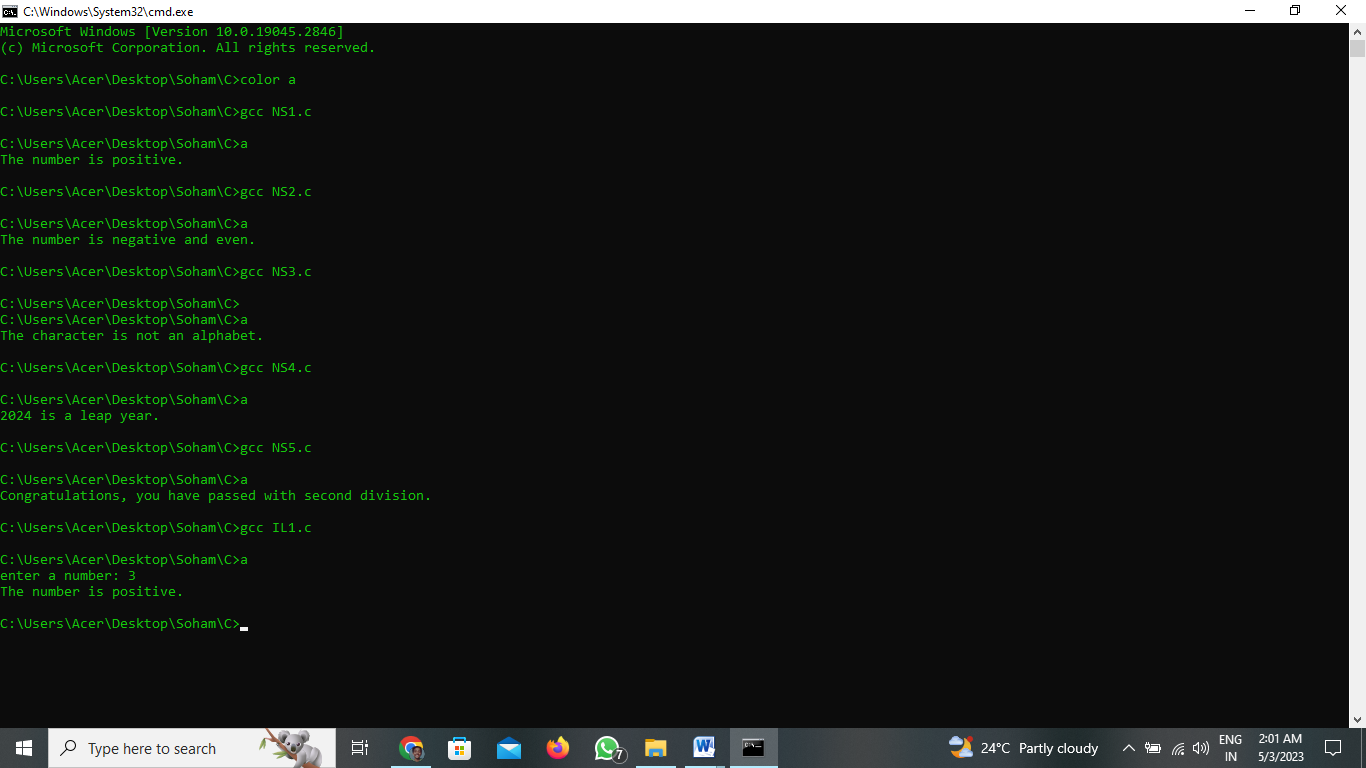


5

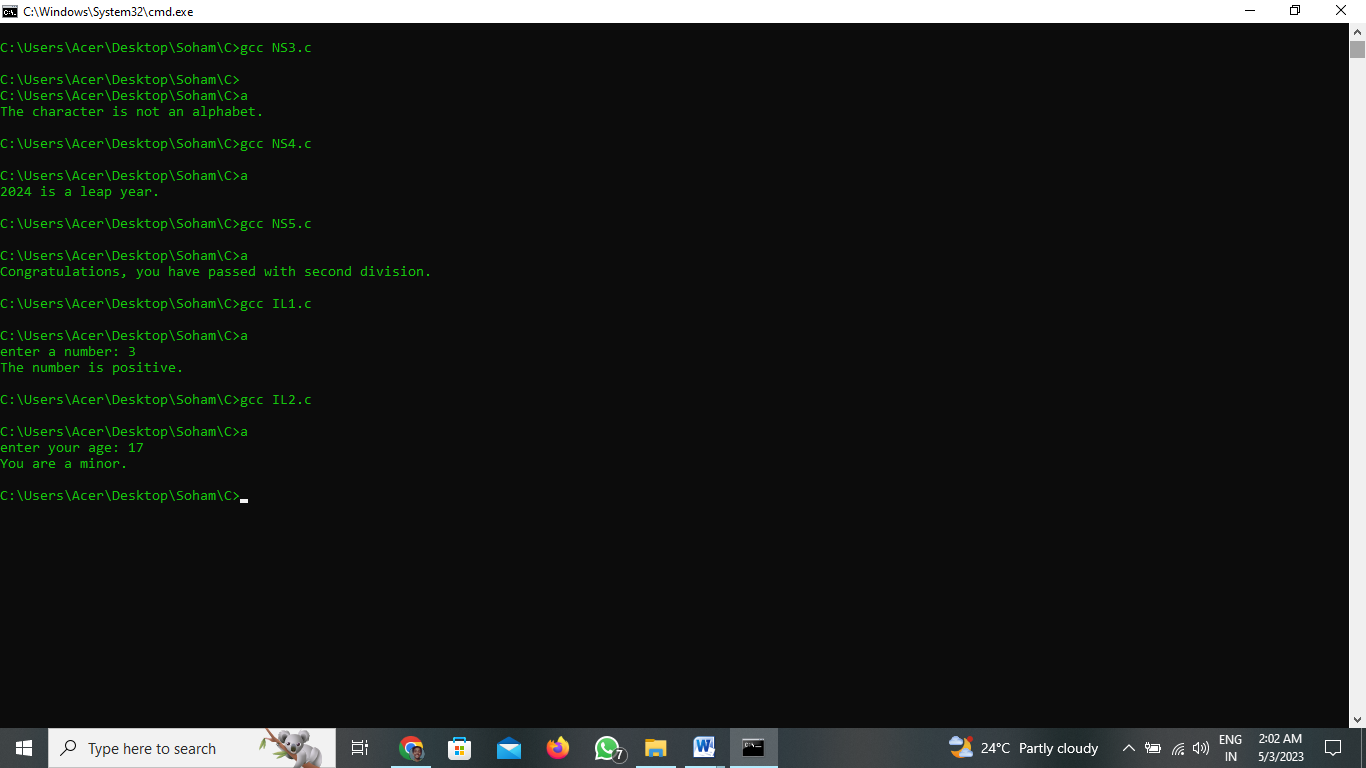


**If else ladder**

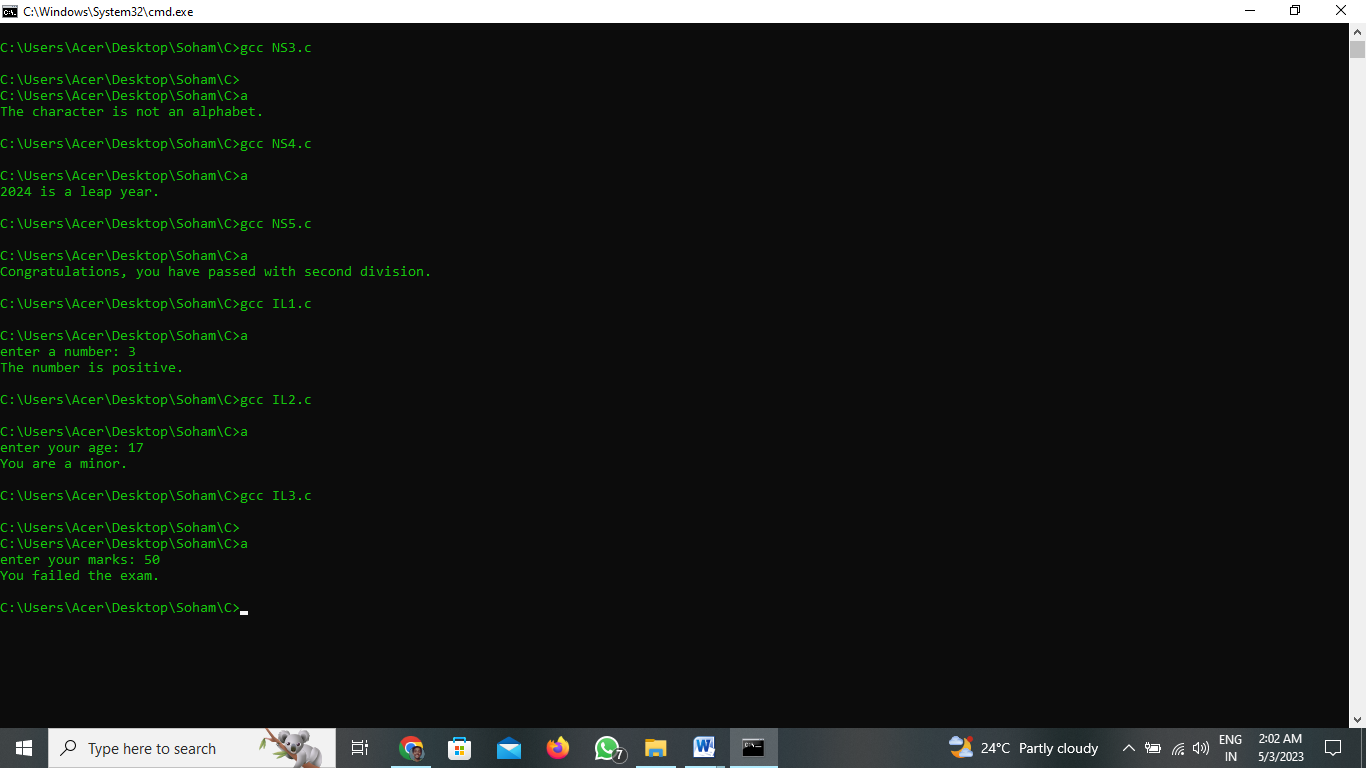
1



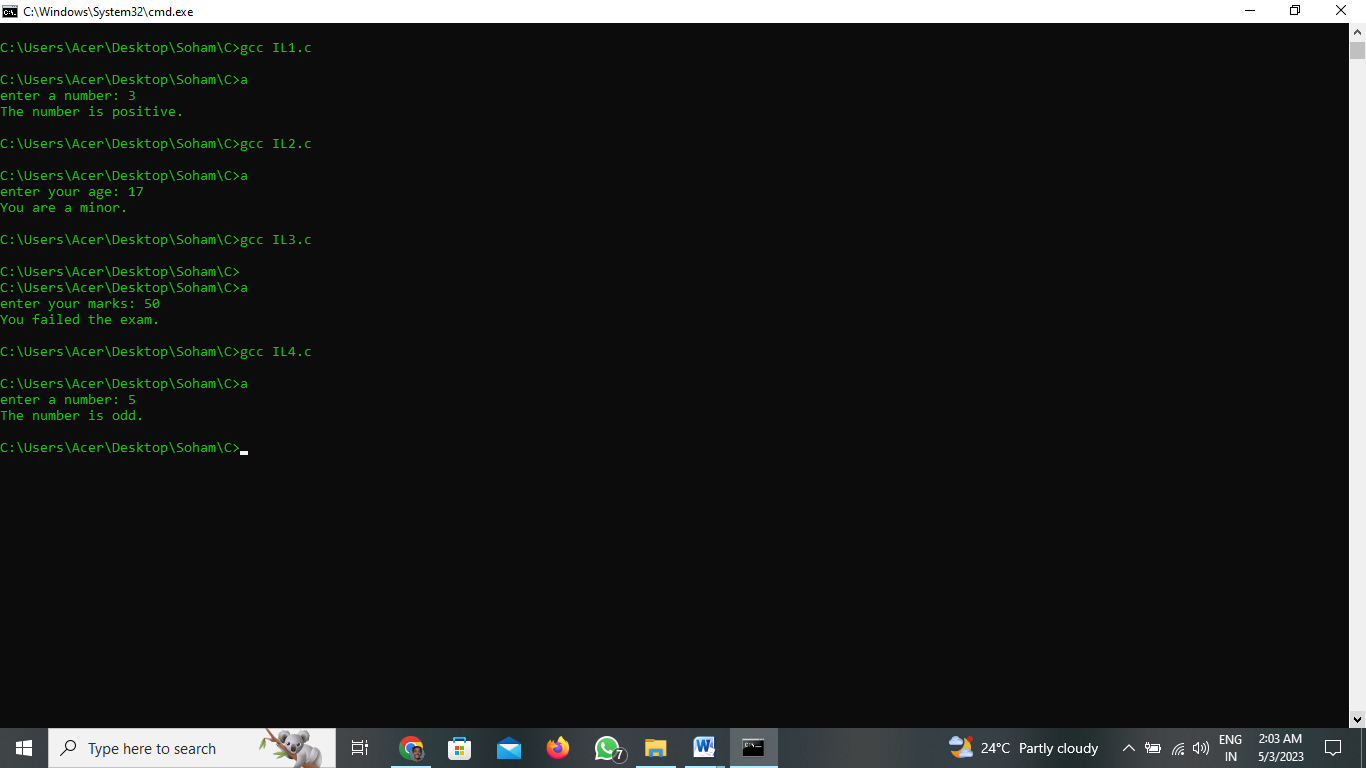
2



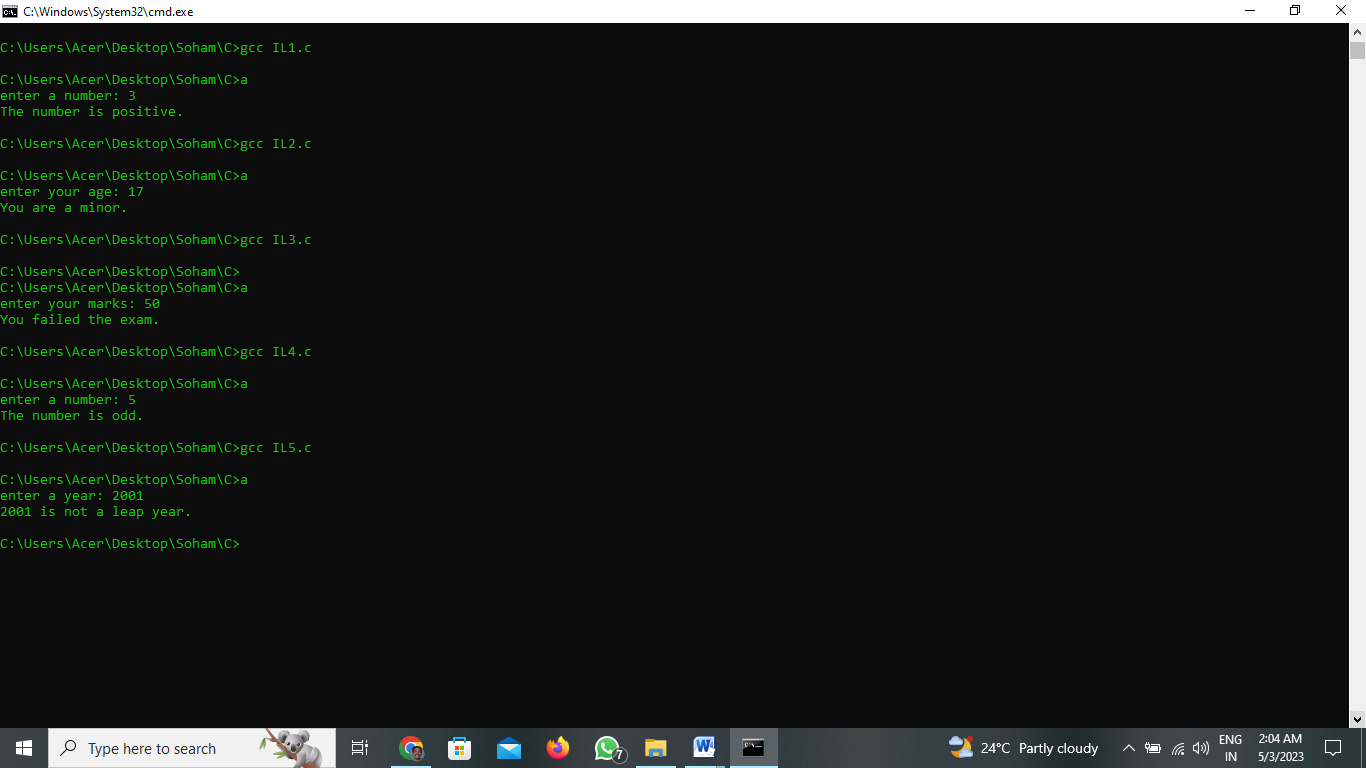
3



4



5



**8. Skill Developed / Learning out of this project.**

* Using if statement we can control the flow of statement(s) in the program.
* There are four types of if Statement in c: simple if, if-else, nested if-else, and else-if ladder.
* In C, if statement supports two-way branching statement and multi-way branching statement.
* We can ignore the ‘else’ part of the program statement and we can simply show the result of the ‘if’ condition/expression in our program.