

www.sjcit.ac.in

|| Jai Sri Gurudev ||
Sri Adichunchanagiri Shikshana Trust®

SJC INSTITUTE OF TECHNOLOGY

An Autonomous Institution under VTU from 2024-25

AICTE Approved, Accredited by NBA [CSE, ISE, ECE, ME, CV, AE] and NAAC with A+ Grade, QS I-Gauge Gold rated

P.B. No. 20, B.B. Road, Chikkaballapur - 562 101, Karnataka.



Estd. 1986

Department Of
Computer Science and Engineering
SEMESTER I/II
PSP LAB MANUAL



www.sjcit.ac.in

|| Jai Sri Gurudev ||
Sri Adichunchanagiri Shikshana Trust®

SJC INSTITUTE OF TECHNOLOGY

An Autonomous Institution under VTU from 2024-25

AICTE Approved, Accredited by NBA [CSE, ISE, ECE, ME, CV, AE] and NAAC with A+ Grade, QS I-Gauge Gold rated

P.B. No. 20, B.B. Road, Chikkaballapur - 562 101, Karnataka.



Estd. 1986

SJC INSTITUTE OF TECHNOLOGY

VISION

Preparing Competent Engineering and Management Professionals to Serve the Society.

MISSION

- **Providing Students with Sound Knowledge in Fundamentals of their branch of Study.**
- **Promoting Excellence in Teaching, Training, Research and Consult**
- **Exposing Students to Emerging Frontiers in various domains enabling Continuous Learning.**
- **Developing Entrepreneurial acumen to venture into Innovative areas.**
- **Imparting Value based Professional Education with a sense of Social Responsibility.**

Department of C S E

Vision Building up-skilled Computer Professionals, enriched with interactive design skills as to serve the dynamic needs of the Industry and Society.

Mission

- **Develop innovative, Proficient and ethically strong Computer Design Engineers with design skills to meet Universal Challenges.**
- **Nurture competence as well as applied Research activities to solve concrete problems.**
- **Aspire for constant up-gradation of engineering skills to cater the needs of the corporate and Society.**
- **Imbibing the spirit of teamwork, core skills, professionalism and confidence to the Leadership role.**
- **Inculcate research culture to design and develop smart computing solutions for humanity and nation.**

INSTRUCTIONS

- 1. Arrive on time with all necessary materials like Pen, Observation.**
- 2. Log in with your assigned credentials and log out before leaving.**
- 3. Save your work frequently and maintain backups in a folder named with your USN.**
- 4. Follow the given tasks and faculty instructions during the session.**
- 5. Write clean, structured, and well-commented code.**
- 6. Seek help from faculty or lab assistants when facing difficulties.**
- 7. Maintain silence and avoid distracting others.**
- 8. Negligence of one candidate will result in penalty for the whole batch.**
- 9. Use the systems only for academic purposes and avoid unauthorized installations.**
- 10. Write original code and avoid copying from peers or online sources.**
- 11. Log out, clean your workspace, and leave the equipment in good condition.**

INDEX

Sl. No	Contents	Page No
1	Syllabus	i
2	CO – PO Mapping	ii
3	Lab Assessment Rubrics for CIE	iii
Program No.	Title of the Program	Page No.
1	C-Program to evaluate complex expressions $((a+b)>(c-d))*d/a$.	4
2	C-Program to calculate the total sales given the unit price, quantity, discount and tax rate.	6
3	Develop a C Program to find the roots of quadratic equation for non-zero co-efficients using if-else ladder construct.	8
4	Develop a C Program to implement to check type of triangle (equilateral triangles, isosceles triangles, and scalene triangles) using switch construct. Display appropriate messages for invalid case.	12
5	Develop a C Program to find the GCD & LCM of two integers using Euclid's algorithm.	15
6	Develop a C Program to generate the Prime numbers between the ranges m & n using functions. Also, print the number of prime numbers generated.	18
7	Develop a recursive C function to find the factorial of a number, n!, defined by $fact(n)=1$, if $n=0$. Otherwise $fact(n)=n*fact(n-1)$. Using this function, develop a C program to compute the Binomial coefficient nCr . Perform input validation as well.	21
8	Develop a C program to read n characters and sort the characters using Bubble sort technique and reverse the sorted array. Print the input array, sorted array and the reverse array with suitable messages.	24
9	Develop a C program to read two matrices A (m x n) and B (p x q) and compute the product of the two matrices. Print both the input matrices and resultant matrix with suitable headings and output should be in matrix format only. Program must check the compatibility of orders of the matrices for multiplication. Report appropriate message in case of incompatibility.	26
10	Develop a C program to accept a matrix of order m x n. Implement the following functions: a. Find the sum of each row b. Find the sum of each column c. Search an element in a matrix	29
11	Develop a C program to Check whether a String is Palindrome or not.	33
12	Develop a C program to find the sum of all the elements of an integer array using pointer.	36
13	Program to read marks of three students in 3 subjects. Calculate the total marks scored, student wise and subject wise using structure.	39
14	Write a C program to copy the contents of one file to another using <code>fscanf()</code> and <code>fprintf()</code> functions.	41
15	Viva Voice Questions	43

Semester: I/II			
C Programming Lab			
Course Code	: 1BCPL107/1BCPL207	Credits	: 01
Hours/Week(L:T:P)	: 0:0:2	CIE	: 50 Marks
Total Hours	: 14 Hrs	SEE	: 50 Marks
Course Type	: PCCL	SEE Duration	: 02 Hours

Prerequisites:	
Courses	Basic Computer Literacy, Programming Fundamentals
Knowledge on Fundamental Concepts	Problem-Solving Skills, Mathematics
Knowledge on Connected Tools (If any)	Visual Studio, Code Blocks, GCC, Clang

Course Learning Objectives: This course will enable the students to	
CLO1	Understand the fundamental syntax, semantics, and constructs of the programming language.
CLO2	Apply decision-making, looping, and control structures to solve basic computational problems.
CLO3	Implement modular programming concepts using functions, recursion, and user-defined data types (structures).
CLO4	Utilize arrays, strings, pointers, and file handling to develop efficient and reusable code.
CLO5	Analyze and solve real-world mathematical and logical problems by designing and developing structured programs.

Sl.NO	PRACTICAL COMPONENT OF IPCC
1.	C-Program to evaluate complex expressions $((a+b)>(c-d))*d/a$.
2.	C-Program to calculate the total sales given the unit price, quantity, discount and tax rate
3.	Develop a C Program to find the roots of quadratic equation for non-zero co-efficient using if-else ladder construct.
4.	Develop a C Program to implement to check type of triangle (equilateral triangles, isosceles triangles, and scalene triangles) using switch construct. Display appropriate messages for invalid case.
5.	Develop a C Program to find the GCD & LCM of two integers using Euclid's algorithm.
6.	Develop a C Program to generate the Prime numbers between the ranges m & n using functions. Also, print the number of prime numbers generated.
7.	Develop a recursive C function to find the factorial of a number, n!, defined by $fact(n)=1$, if $n=0$. Otherwise $fact(n)=n*fact(n-1)$. Using this function, develop a C program to compute the Binomial coefficient nCr . Perform input validation as well.
8.	Develop a C program to read n characters and sort the characters using Bubble sort technique and reverse the sorted array. Print the input array, sorted array and the reverse array with suitable messages.
9.	Develop a C program to read two matrices A (m x n) and B (p x q) and compute the product of the two matrices. Print both the input matrices and resultant matrix with suitable headings and output should be in matrix format only. Program must check the compatibility of orders of the matrices for multiplication. Report appropriate message in case of incompatibility.
10.	Develop a C program to accept a matrix of order m x n. Implement the following functions: a. Find the sum of each row b. Find the sum of each column

	c. Search an element in a matrix
11.	Develop a C program to Check whether a String is Palindrome or not.
12.	Develop a C program to find the sum of all the elements of an integer array using pointer.
13.	C-Program to read marks of three students in 3 subjects. Calculate the total marks scored, student wise and subject wise using structure.
14.	Write a C program to copy the contents of one file to another using fscanf() and fprintf() functions.

Course Outcomes: After completing the course, the students will be able to	
CO1	Develop and evaluate expressions, use operators, and perform computations.
CO2	Design programs using conditional and looping constructs.
CO3	Implement modular programs using functions and recursion.
CO4	Demonstrate the use of arrays, strings, pointers, and sorting techniques.
CO5	Apply operations and file handling techniques to solve data manipulation problems.

Text Book	
1	Computer Fundamentals and Programming in C, Reema Thareja, 3rd Edition, ISBN: 9789354977893, Oxford University Press, 2023.

Reference Books	
1	The C Programming Language, Kernighan B.W and Dennis M. Ritchie, 2015, 2nd Edition, Prentice Hall, ISBN (13): 9780131103627.
2	Behrouz A. Forouzan and Richard F. Gilberg, Computer Science: A Structured Approach Using C, Cengage Learning, 3rd edition, 2013, ISBN :-9788131517888
3	Let Us C: Authentic Guide to C Programming Language, Yashavant Kanetkar 17th Edition, 2020, BPB PUBN , ISBN- 9789389845686

Web links and Video Lectures (e – Resources)	
1	https://tinyurl.com/4xmrexre
2	https://www.geeksforgeeks.org/category/programming-language/c/?ref=article_category
3	https://nptel.ac.in/courses/106/105/106105171/ MOOC courses can be adopted for more clarity in understanding the topics and verities of problem solving methods
4	https://nptel.ac.in/courses/106/105/106105171/ MOOC courses can be adopted for more clarity in understanding the topics and verities of problem solving methods
5	https://nptel.ac.in/courses/106/105/106105171/ MOOC courses can be adopted for more clarity in understanding the topics and verities of problem solving methods

CO – PO MAPPING

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	3	2	1	1	2			1	1	2	
CO2	3	3	2	2	2			1	1	2	
CO3	3	3	2	2	2			1	1	2	
CO4	3	3	2	2	2			1	1	2	
CO5	3	3	3	2	2			1	1	2	

Lab Assessment Rubrics for CIE

Components		#	Weightage	Max. Marks	Min. Requirement
(i)	Class Work (A)	Continuous	50%	25	10
(ii)	Lab Test (B)	1	30%	15	06
(iii)	Open Ended Experiment/ Project Based Learning (PBL)/ Model demonstration (C)	1	20%	10	04
Final CIE Marks = (A) + (B) + (C) = 50					
Passing Requirement: 40% of Final CIE marks = 20					

a) Continuous Evaluation Rubrics : 25 Marks

Parameters		Marks
A	Observation Write-up and Attendance	05
B	Conduction of Experiment and Output	10
C	Viva Voice	05
D	Record Write up	05
Total		25

b) Lab Test : 15 Marks

Parameters		Marks
A	Write Up	03
B	Conduction of Experiment and Output	08
C	Viva Voice	04
Total		15

c) Open Ended Experiment / Project Based Learning: 10 Marks

4. Introduction to Problem Solving Through Programming

Problem Solving Techniques: The process of working through details of a problem to reach a solution.

There are three approaches to problem solving:

- Algorithm
- Flowchart
- Pseudo Code

Algorithm: The algorithm is a step-by-step procedure to be followed in solving a problem. It provides a scheme to solve a particular problem in finite number of unambiguous steps. It helps in implementing the solution of a problem using any of the programming languages.

In order to qualify as an algorithm, a sequence of instructions must possess the following characteristics:

Input: It may accept a zero or more inputs.

Output: It should produce at least one output (result).

Definiteness: Each instruction must be clear, well defined and precise. There should not be any ambiguity.

Finiteness: It should be a sequence of finite instructions. That is, it should end after a fixed time. It should not enter into an infinite loop.

Effectiveness: This means that operations must be simple and are carried out in a finite time at one or more levels of complexity. It should be effective whenever traced manually for the results.

Key features of an algorithm: Any algorithm has a finite number of steps and some steps may involve decision making, repetition. Broadly speaking, an algorithm exhibits three key features that can be given as:

Sequence: Sequence means that each step of the algorithm is executed in the specified order.

Decision: Decision statements are used when the outcome of the process depends on some condition.

Repetition: Repetition which involves executing one or more steps for a number of times can be implemented using constructs like the while, do-while and for loops. These loops executed one or more steps until some condition is true.

Example: To compute the Area of Rectangle

Algorithm: Area_of_rectangle [This algorithm takes length and breadth, the sides of the rectangle as input and computes the area of rectangle using the formula $area=length * breadth$. Finally, it prints the area of rectangle]

STEPS:

Step 1:[Initialize], Start

Step 2: [Input the sides of Rectangle], Read length, breadth

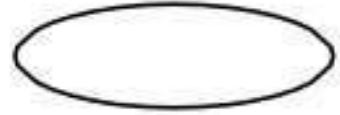


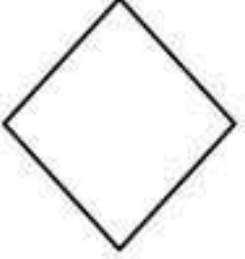
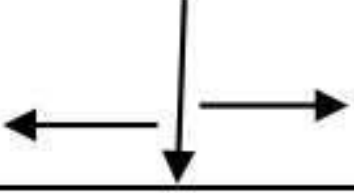
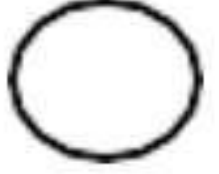


Step 3:[Compute the area of rectangle], $Area=length*breadth$

Step 4:[Display the Area], Print Area

Step 5: [Finished], Stop

Flowcharts: A flowchart is a graphical or symbolic representation of an algorithm. They are basically used to design and develop complex programs to help the users to visualize the logic of the program so that they can gain a better understanding of the program and find flaws, bottlenecks, and other less-obvious features within it. Basically, a flowchart depicts the “flow” of a program.

The following table shows the symbols used in flow chart along with its descriptions.

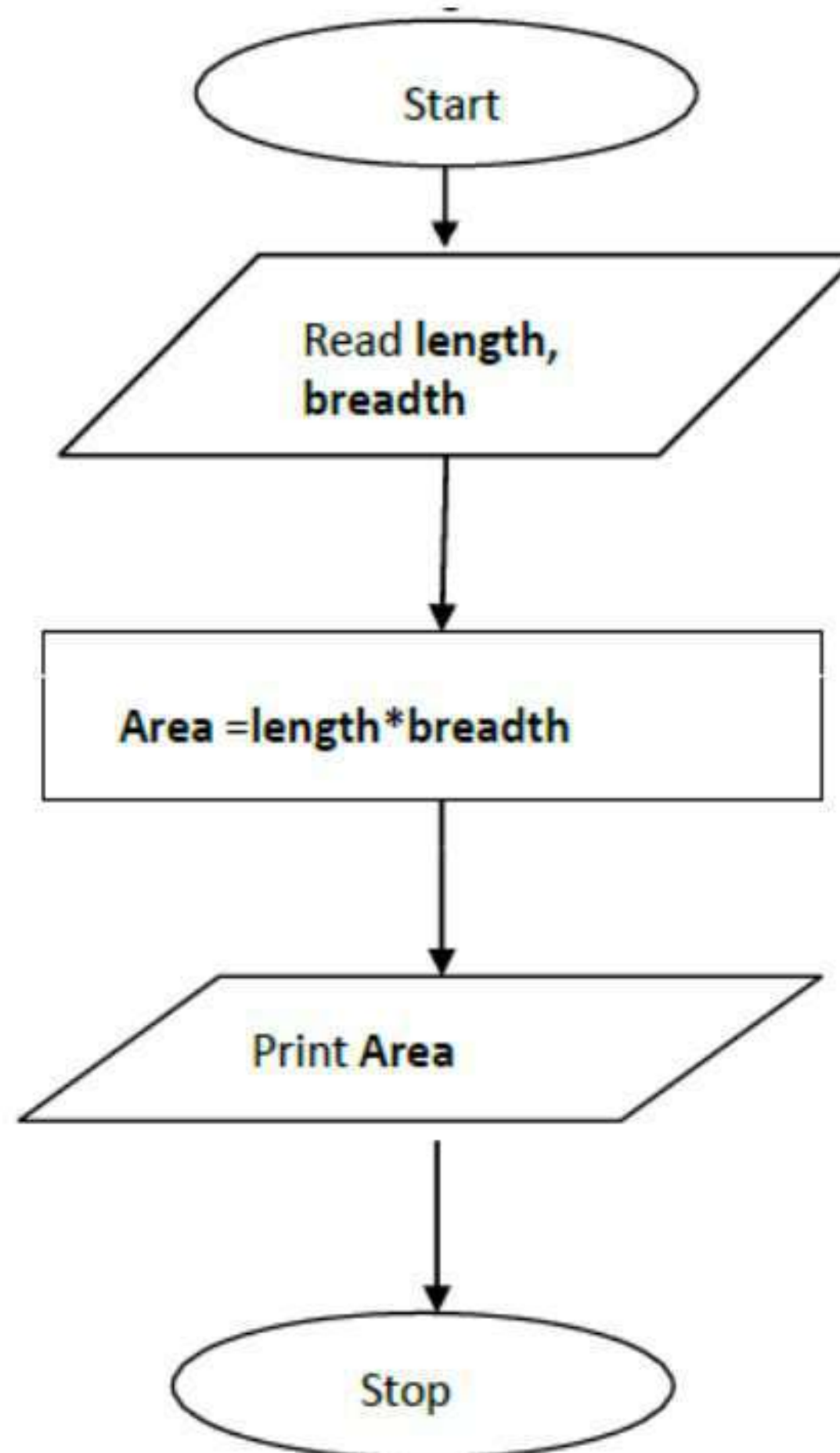
Symbol	Name	Description
	Oval	Represents the terminal point
	Rectangle	Represents the process steps defined in algorithm
	Parallelogram	Indicate the reading Operation used for input/output or data or information from/to any device
	Diamond	Indicates the decisions(questions) and consequently the Branch points or the paths to be followed based on the result of the question
	Arrows	Shows the flow chart direction and connects the various flow chart symbols
	Small circle	Shows the continuation from one point in the process flow To another
	Hexagon	Represents Looping structures
	Predefined Process	Indicates Subroutines

Advantages of Flowcharts:

1. A flowchart is a diagrammatic representation that illustrates the sequence of steps that must be performed to solve a problem.
2. They are usually drawn in the early stages of formulating computer solutions to facilitate communication between programmers and businesspeople.

3. Flowcharts help programmers to understand the logic of complicated and lengthy problems and they help to analyse the problem in a more effective manner.
4. Flow chart can be used to debug programs that have error(s).

Example: To compute the area of rectangle



Program – 1**C- Program to evaluate complex expressions $((a+b)>(c-d))*d/a$.****Program:**

```
#include <stdio.h>

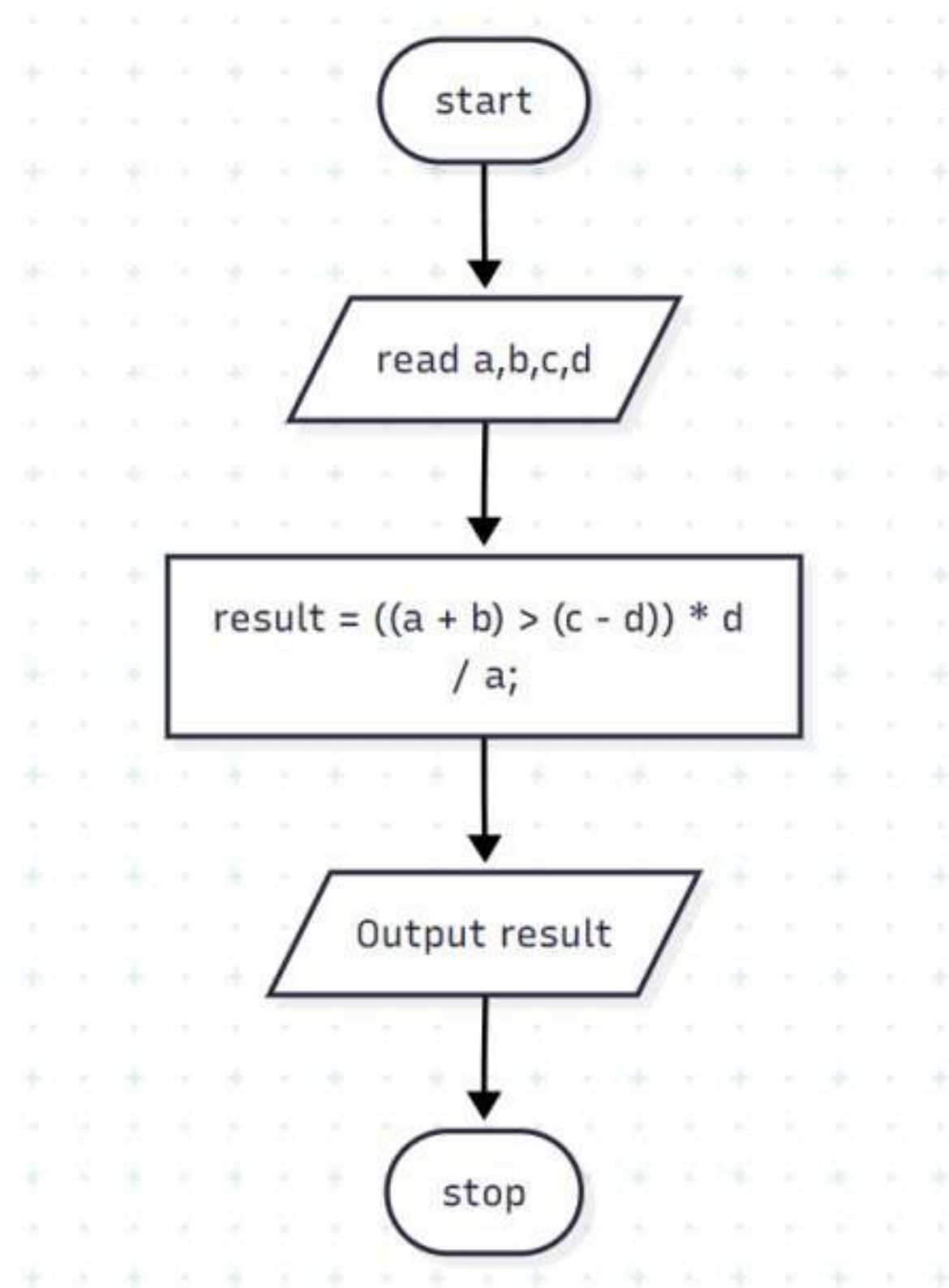
int main()
{
    float a, b, c, d, result;

    printf("Enter values for a, b, c, and d (a should not be zero):\n");
    scanf("%f %f %f %f", &a, &b, &c, &d);

    result = ((a + b) > (c - d)) * d / a;

    printf("The result of the expression is: %.2f\n", result);

    return 0;
}
```

Flow Chart :

Test Cases:

Test No.	Input Parameters	Expected Output	Obtained Output	Remarks	Student Remarks
1	a=1,b=2,c=3 and d=4	4	4	PASS	
2	a=7,b=19,c=20 and d=33	4.71	4.71	PASS	
Test for the following cases and record your observation					
3					
4					

Rubrics for Continuous Evaluation of Experiments:

Continuous Evaluation for 25 Marks		Marks Allotted	Marks Obtained
A	Observation Write-up and Attendance	05	
B	Conduction of Experiment and Output	10	
C	Viva Voice	05	
D	Record Write up	05	
Total		25	

Signature of Lab Faculty In-charge/Co-Faculty

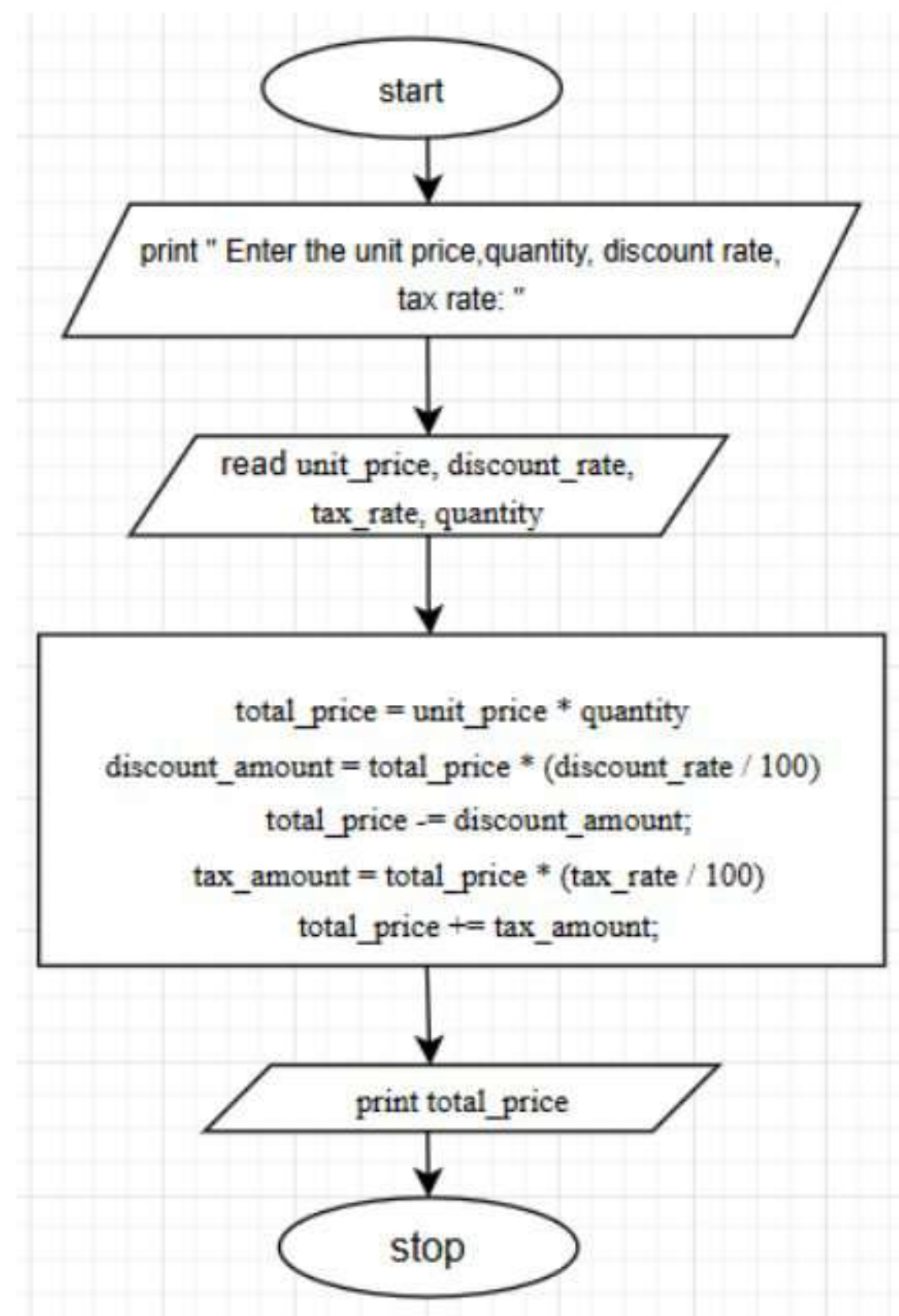
Program -2

C- Program to calculate the total sales given the unit price, quantity, discount and tax rate.

Program:

```
#include <stdio.h>
int main()
{
    float unit_price, discount_rate, tax_rate, total_price, discount_amount,
    tax_amount, quantity;
    printf("Enter the unit price, quantity, discount rate, tax rate: ");
    scanf("%f%f%f%f", &unit_price, &quantity, &discount_rate, &tax_rate);
    total_price = unit_price * quantity;
    discount_amount = total_price * (discount_rate / 100);
    total_price -= discount_amount;
    tax_amount = total_price * (tax_rate / 100);
    total_price += tax_amount;
    printf("The total sales amount is: %.2f\n", total_price);
    return 0;
}
```

Flow chart:



Test Cases:

Test No.	Input Parameters	Expected Output	Obtained Output	Remarks	Student Remarks
1	unit_price=30, quantity=10, discount_rate=05, tax_rate=18,	336.30	336.30	PASS	
2	unit_price=120, quantity=05, discount_rate=3, tax_rate=5,	611.10	611.10	PASS	
Test for the following cases and record your observation					
3					
4					

Rubrics for Continuous Evaluation of Experiments:

Continuous Evaluation for 25 Marks		Marks Allotted	Marks Obtained
A	Observation Write-up and Attendance	05	
B	Conduction of Experiment and Output	10	
C	Viva Voice	05	
D	Record Write up	05	
Total		25	

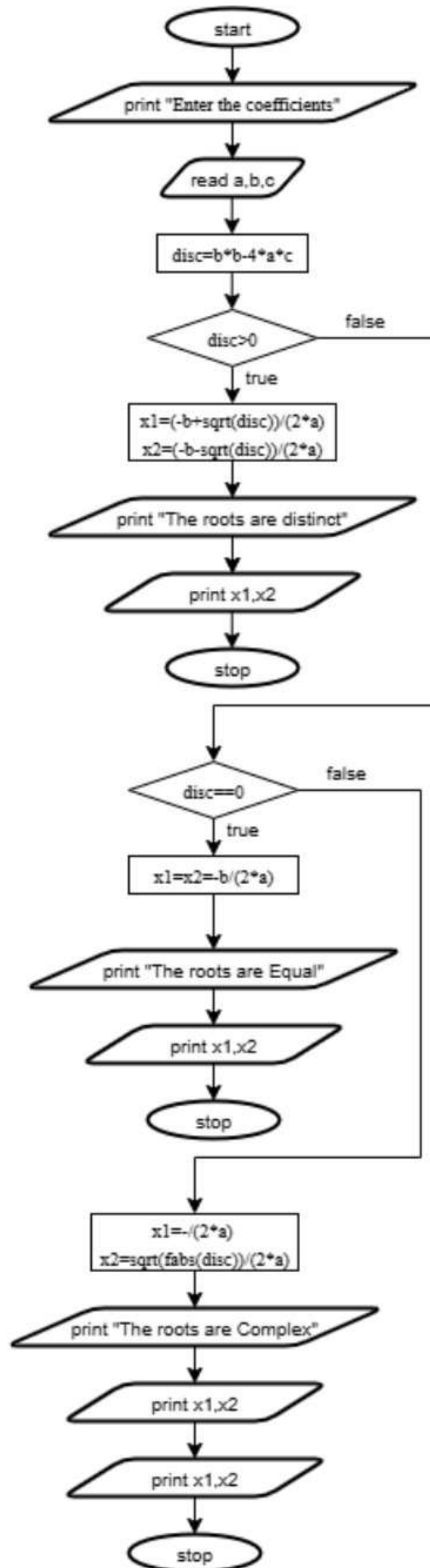
Signature of Lab Faculty In-charge/Co-Faculty

Program - 3

Develop a C Program to find the roots of quadratic equation for non-zero coefficients using if-else ladder construct.

Program:

```
#include<stdio.h>
#include<math.h>
void main( )
{
    float a,b,c,d,real,img,root1,root2;
    printf("\n Enter the values for a, b and c\n");
    scanf("%f%f%f",&a,&b,&c);
    d=b*b-4*a*c;
    printf("\n The value of Discriminant(d) is %f", d);
    if(d==0)
    {
        printf("\n The real and equal roots are:\n");
        root1 = root2 =-b/(2*a);
        printf("root1=root2=%.2f",root1);
    }
    else if(d>0)
    {
        printf("\n The real and distinct roots are:\n");
        root1=(-b+sqrt(d))/(2*a);
        root2=(-b - sqrt(d))/(2*a);
        printf("root1=%.2f and root2=%.2f",root1,root2);
    }
    else
    {
        printf("\n The complex roots are:\n");
        real =-b/(2*a);
        img=sqrt(fabs(d))/(2*a);
        printf("root1=%.2f+%.2fi\t",real,img);
        printf("root2 =%f-%.2fi",real,img);
    }
}
```

Flow Chart:

Test Cases:

Test No.	Input Parameters	Expected output	Obtained output	Remarks	Student remarks
1	a=1, b=6, c=9	Roots are real and equal Root1=-3 and Root2=-3	Roots are real and equal Root1 = -3.000 and Root2=-3.000	PASS	
2	a=1, b=-5, c=3	Roots are real and distinct Root1= 4.30 and Root2=0.69	Roots are real and distinct Root1= 4.303 and Root2=0.697	PASS	
3	a=1, b=4, c=7	Roots are real and imaginary Root1=-2+i1.73 Root2=-2-i1.73	Roots are real and imaginary Root1=2.000+i1.73 2 Root2=-2.000-i1.732	PASS	
Test for the following cases & record your observations					
4					
5					
6					

Rubrics for Continuous Evaluation of Experiments:

Continuous Evaluation for 25 Marks		Marks Allotted	Marks Obtained
A	Observation Write-up and Attendance	05	
B	Conduction of Experiment and Output	10	
C	Viva Voice	05	
D	Record Write up	05	
Total		25	

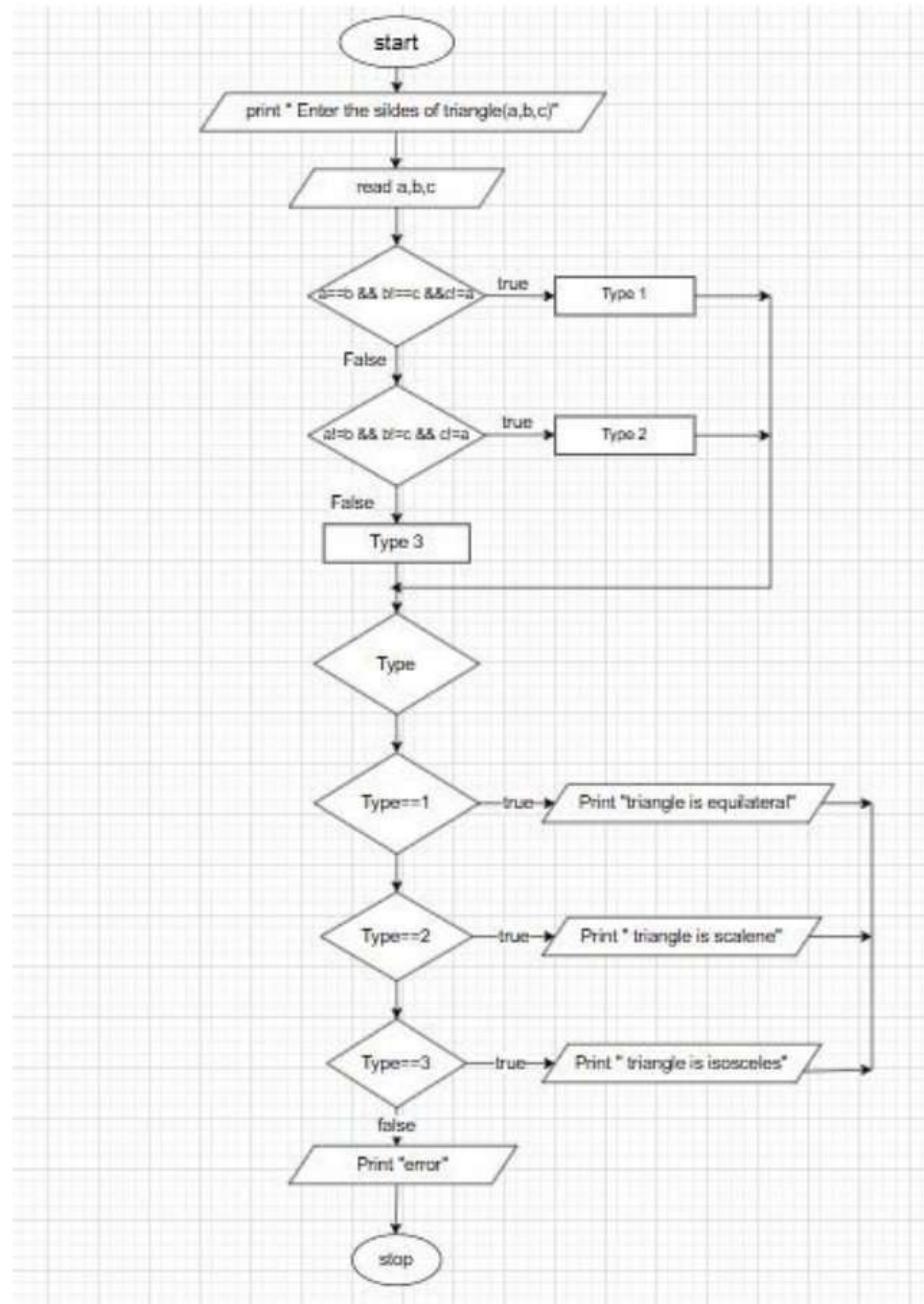
Signature of Lab Faculty In-charge/Co-Faculty

Program – 4

Develop a C Program to implement to check type of triangle (equilateral triangles, isosceles triangles, and scalene triangles) using switch construct. Display appropriate messages for invalid case.

Program:

```
#include <stdio.h>
int main()
{
    int a, b, c, type = 0;
    printf("Enter the sides of the triangle (a, b, c):\n");
    scanf("%d%d%d",&a,&b,&c);
    if (a==b && b==c && c==a)
        type=1;
    else if (a!=b && b!=c && c!=a)
        type=2;
    else
        type=3;
    switch (type)
    {
        case 1:    printf("The triangle is Equilateral.\n");
                   break;
        case 2:    printf("The triangle is Scalene.\n");
                   break;
        case 3:    printf("The triangle is Isosceles \n");
                   break;
        default:   printf("Error: Unexpected case.\n");
    }
    return 0;
}
```

Flow Chart:

Test Cases:

Test No	Input Parameters	Expected output	Obtained output	Remarks	Student Remarks
1	Enter the sides of the triangle (a, b, c) 2,2,2	The triangle is Equilateral.	The triangle is Equilateral.	PASS	
2	Enter the sides of the triangle (a, b, c) 2,3,3	The triangle is Isosceles.	The triangle is Isosceles.	PASS	
3	Enter the sides of the triangle (a, b, c) 2,1,3	The triangle is Scalene.	The triangle is Scalene.	PASS	
Test for the following cases & record your observations					
3					
4					

Rubrics for Continuous Evaluation of Experiments:

Continuous Evaluation for 25 Marks		Marks Allotted	Marks Obtained
A	Observation Write-up and Attendance	05	
B	Conduction of Experiment and Output	10	
C	Viva Voice	05	
D	Record Write up	05	
Total		25	

Signature of Lab Faculty In-charge/Co-Faculty

Program – 5

Develop a C Program to find the GCD & LCM of two integers using Euclid's algorithm.

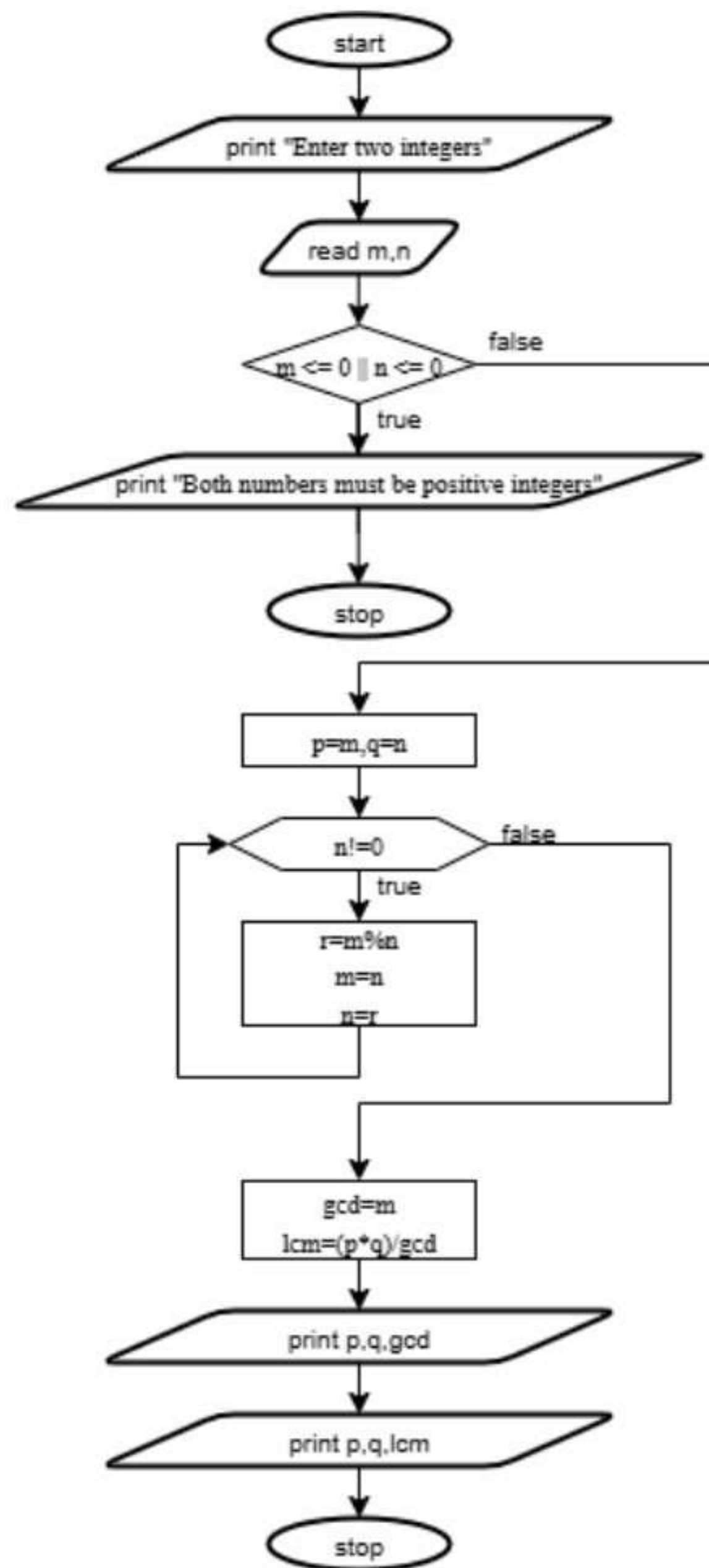
Program:

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int m,n,p,q,temp,r,gcd,lcm;

    printf("Enter two integers:\n");
    scanf("%d%d",&m,&n);

    if (m <= 0 || n <= 0)
    {
        printf("Both numbers must be positive integers.\n");
        return 1;
    }
    p=m,q=n;
    while(n!=0)
    {
        r=m%n;
        m=n;
        n=r;
    }
    gcd=m;
    lcm=(p*q)/gcd;
    printf("GCD of %d and %d is: %d\n", p, q, gcd);
    printf("LCM of %d and %d is: %d\n", p, q, lcm);
    return 0;
}
```

Flow Chart:

Test Cases:

Test No.	Input Parameters	Expected output	Obtained output	Remarks	Student Remarks
1	m=20,n=20	GCD of 20 and 10 is: 10 LCM of 20 and 10 is: 20	GCD of 20 and 10 is: 10 LCM of 20 and 10 is: 20	PASS	
2	m=5,n=8	GCD of 5 and 8 is: 1 LCM of 5 and 8 is: 40	GCD of 5 and 8 is: 1 LCM of 5 and 8 is: 40	PASS	
Test for the following cases & record your observations					
3					
4					

Rubrics for Continuous Evaluation of Experiments:

Continuous Evaluation for 25 Marks		Marks Allotted	Marks Obtained
A	Observation Write-up and Attendance	05	
B	Conduction of Experiment and Output	10	
C	Viva Voice	05	
D	Record Write up	05	
Total		25	

Signature of Lab Faculty In-charge/Co-Faculty

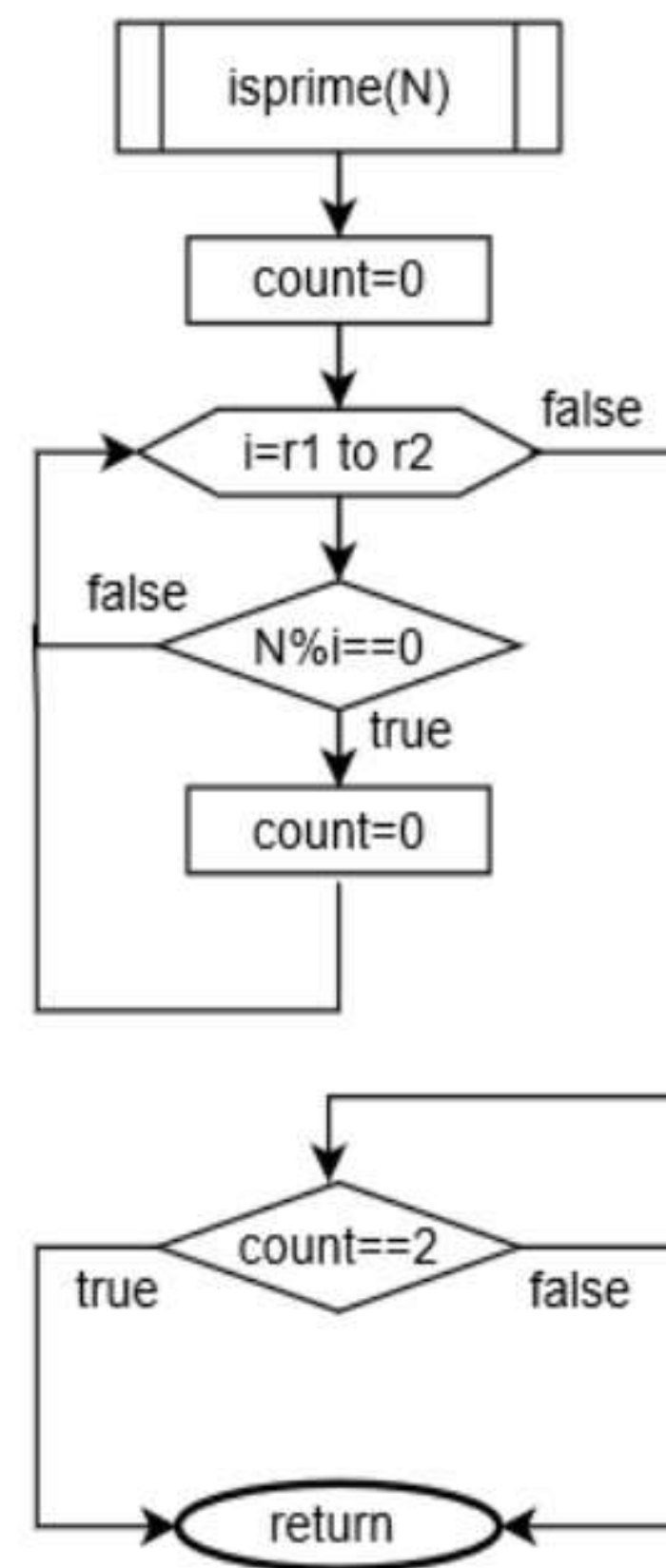
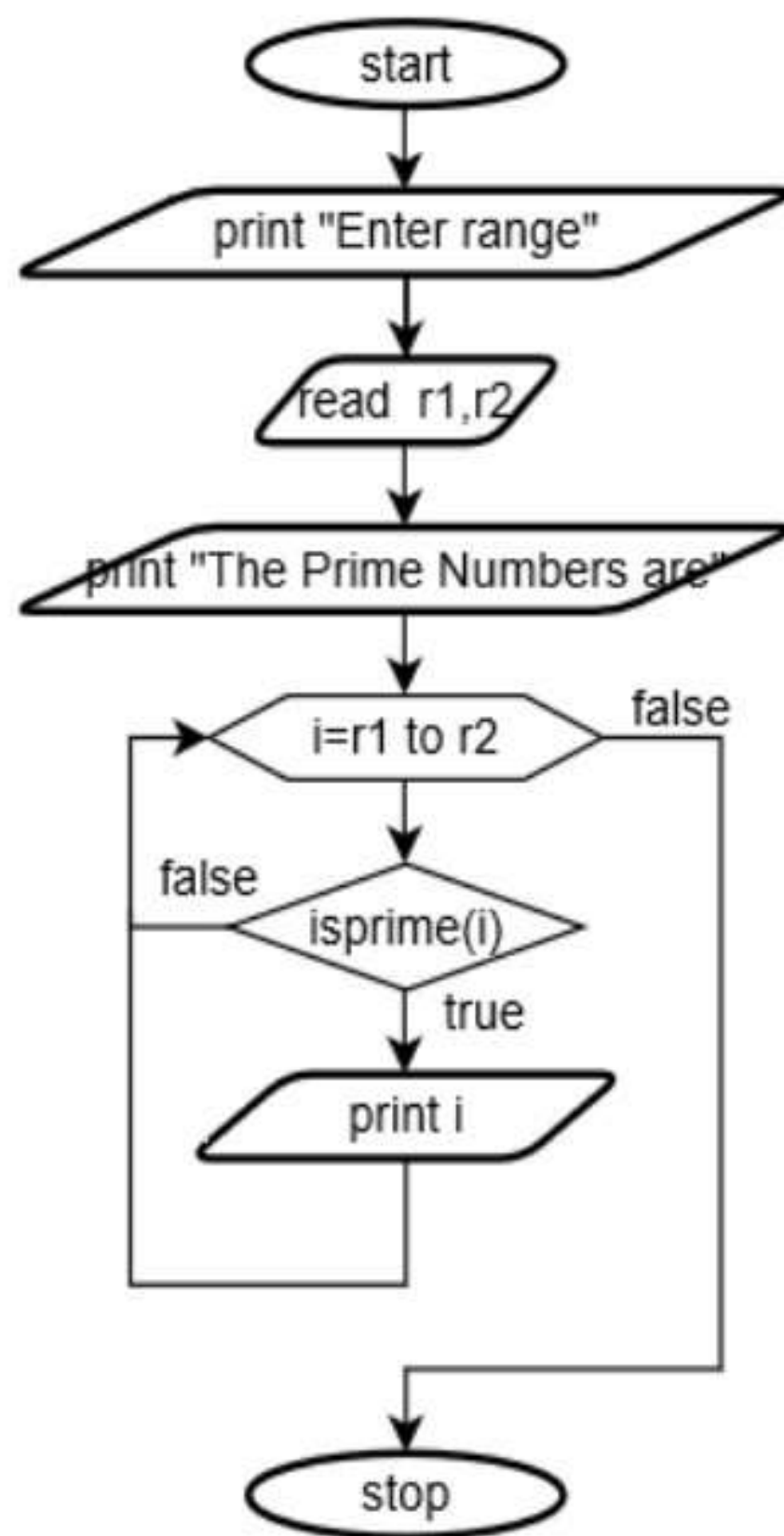
Program – 6

Develop a C Program to generate the Prime numbers between the ranges m & n using functions. Also, print the number of prime numbers generated.

```
#include <stdio.h>
#include <stdbool.h>

int isPrime(int N)
{
    int count=0,i;
    for(i=1;i<=N;i++)
    {
        if(N%i==0)
            count++;
    }
    if(count==2)
        return 1;
    else
        return 0;
}

int main()
{
    int r1,r2,i;
    printf("Enter range \n");
    scanf("%d%d",&r1,&r2);
    printf("The Prime Numbers are\n");
    for(i=r1;i<=r2;i++)
    {
        if (isPrime(i))
            printf("%d\n",i);
    }
    return 0;
}
```

Flow Chart:

Test Cases:

Test No.	Input Parameters	Expected output	Obtained output	Remarks	Student Remarks
1	Enter range r1=10,r2= 20	The Prime Numbers are 11,13,17,19	The Prime Numbers are 11,13,17,19	PASS	
2	Enter range r1=5,r2= 10	The Prime Numbers are 5,7	The Prime Numbers are 5,7	PASS	
Test for the following cases & record your observations					
3					
4					

Rubrics for Continuous Evaluation of Experiments:

Continuous Evaluation for 25 Marks		Marks Allotted	Marks Obtained
A	Observation Write-up and Attendance	05	
B	Conduction of Experiment and Output	10	
C	Viva Voice	05	
D	Record Write up	05	
Total		25	

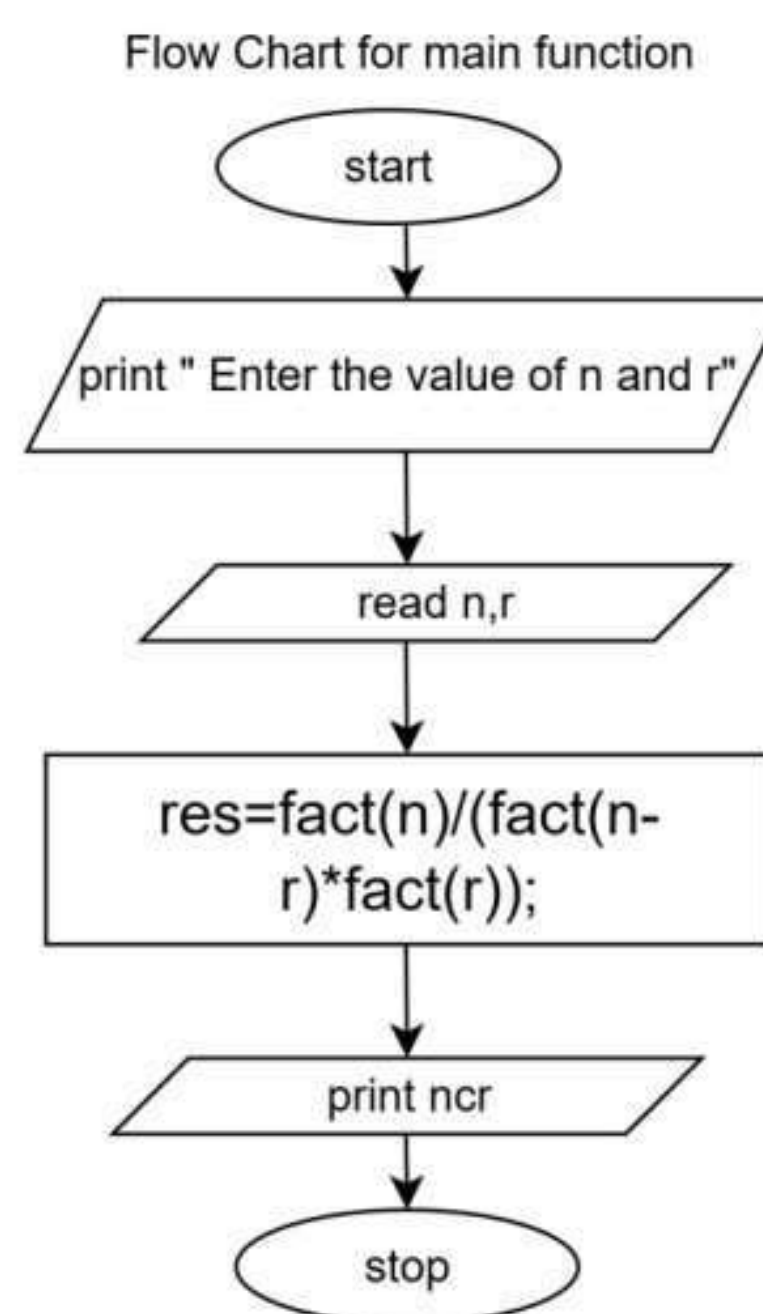
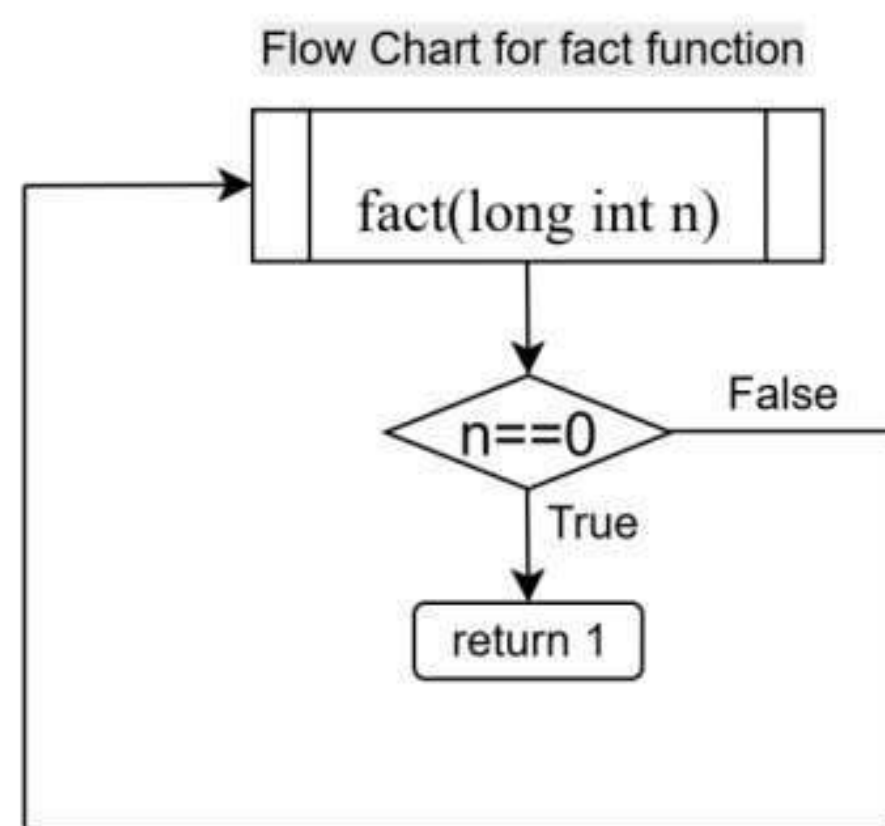
Signature of Lab Faculty In-charge/Co-Faculty

Program – 7

Develop a recursive C function to find the factorial of a number, $n!$, defined by $\text{fact}(n)=1$, if $n=0$. Otherwise $\text{fact}(n)=n*\text{fact}(n-1)$. Using this function, develop a C program to compute the Binomial coefficient nCr . Perform input validation as well.

Program:

```
#include<stdio.h>
#include<conio.h>
long int fact(long int n)
{
    if(n==0)
        return 1;
    else
        return (n*fact(n-1));
}
int main()
{
    long int n,r,res;
    printf("Enter the value of n and r\n");
    scanf("%ld%ld",&n,&r);
    res=fact(n)/(fact(n-r)*fact(r));
    printf("The NCR is = %ld",res);
}
```

Flow charts:

Test Cases:

Test No.	Input Parameters	Expected output	Obtained output	Remarks	Student Remarks
1	n = 10, r=10	The NCR is = 1	The NCR is = 1	PASS	
2	n = 10, r=2	The NCR is = 45	The NCR is = 45	PASS	
Test for the following cases & record your observations					
3					
4					

Rubrics for Continuous Evaluation of Experiments:

Continuous Evaluation for 25 Marks		Marks Allotted	Marks Obtained
A	Observation Write-up and Attendance	05	
B	Conduction of Experiment and Output	10	
C	Viva Voice	05	
D	Record Write up	05	
Total		25	

Signature of Lab Faculty In-charge/Co-Faculty

Program – 8

Develop a C program to read n characters and sort the characters using Bubble sort technique and reverse the sorted array. Print the input array, sorted array and the reverse array with suitable messages.

Program:

```
#include <stdio.h>

void bubbleSort(char arr[], int n)
{
    int i, j;
    char temp;
    for (i = 0; i < n - 1; i++)
    {
        for (j = 0; j < n - 1 - i; j++)
        {
            if (arr[j] > arr[j + 1])
            {
                temp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = temp;
            }
        }
    }
}

void reverseArray(char arr[], int n)
{
    int i;
    char temp;
    for (i = 0; i < n / 2; i++)
    {
        temp = arr[i];
        arr[i] = arr[n - 1 - i];
        arr[n - 1 - i] = temp;
    }
}

int main()
{
    int n, i, a[10];
    printf("Enter the number of characters: ");
    scanf("%d", &n);

    char arr[n];

    printf("Enter %d characters:\n", n);
    // Use a space before %c to skip any newline/space characters
```

```
for (i = 0; i < n; i++)
{
    scanf(" %c", &arr[i]);
}

printf("Input array: ");
for (i = 0; i < n; i++)
{
    printf("%c ", arr[i]);
}
printf("\n");
bubbleSort(arr, n);
printf("Sorted array: ");
for (i = 0; i < n; i++)
{
    printf("%c ", arr[i]);
}
printf("\n");
reverseArray(arr, n);
printf("Reversed array: ");
for (i = 0; i < n; i++)
{
    printf("%c ", arr[i]);
}
printf("\n");
return 0;
}
```

Test Cases:

Test No.	Input Parameters	Expected output	Obtained output	Remarks	Student Remarks
1	Enter the number of characters n=4 x,b,a,y	Input array: x,b,a,y Sorted array:a,b,x,y Reversed sorted array:y,x,b,a	Input array: x,b,a,y Sorted array:a,b,x,y Reversed sorted array:y,x,b,a	PASS	
2	Enter the number of characters n=3 x,a,m	Input array: x,a,m Sorted array:a,m,x Reversed sorted array: x,m,a	Input array: x,a,m Sorted array:a,m,x Reversed sorted array: x,m,a	PASS	
Test for the following cases & record your observations					
3					
4					

Rubrics for Continuous Evaluation of Experiments:

Continuous Evaluation for 25 Marks		Marks Allotted	Marks Obtained
A	Observation Write-up and Attendance	05	
B	Conduction of Experiment and Output	10	
C	Viva Voice	05	
D	Record Write up	05	
Total		25	

Signature of Lab Faculty In-charge/Co-Faculty

Program – 9

Develop a C program to read two matrices A (m x n) and B (p x q) and compute the product of the two matrices. Print both the input matrices and resultant matrix with suitable headings and output should be in matrix format only. Program must check the compatibility of orders of the matrices for multiplication. Report appropriate message in case of incompatibility.

Program:

```
#include<stdio.h>
void main()
{
    int a[10][10],b[10][10],c[10][10];
    int m,n,p,q,i,j,k;
    printf("\n enter the order of the matrix A\n");
    scanf("%d%d",&m,&n);
    printf("\n enter the order of the matrix B\n");
    scanf("%d%d",&p,&q);
    if(n!=p)
    {
        printf("\n Invalid inputs, Multiplication is not Possible \n");
    }
    else
    {
        printf("\n enter the elements of matrix A\n");
        for(i=0;i<m;i++)
        {
            for(j=0;j<n;j++)
            {
                scanf("%d",&a[i][j]);
            }
        }
        printf("\n enter the elements of matrix B\n");
        for(i=0;i<p;i++)
        {
            for(j=0;j<q;j++)
            {
                scanf("%d",&b[i][j]);
            }
        }
        /* Multiplication */
        for(i=0;i<m;i++)
        {
            for(j=0;j<q;j++)
            {
                c[i][j]=0;
                for(k=0;k<p;k++)
                {
```

```
                c[i][j] = c[i][j]+a[i][k]*b[k][j];
            }
        }
    }
    printf(" \n The Resultant Matrix is: \n");
    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
        {
            printf("%d\t",c[i][j]);
        }
        printf("\n");
    }
}
```

Test Cases:

Test No.	Input Parameters	Expected output	Obtained output	Remarks	Student Remarks
1	Matrix A Size=2,2 Matrix B Size= 2, 2 Elements= 12,56,34,78	The product of two matrix 19 22 43 50	The product of two matrix 19 22 43 50	PASS	
2	Matrix A Size=2,3 Matrix B Size=2,2	Invalid input	Multiplication not possible	PASS	
Test for the following cases & record your observations					
3					
4					

Rubrics for Continuous Evaluation of Experiments:

Continuous Evaluation for 25 Marks		Marks Allotted	Marks Obtained
A	Observation Write-up and Attendance	05	
B	Conduction of Experiment and Output	10	
C	Viva Voice	05	
D	Record Write up	05	
Total		25	

Signature of Lab Faculty In-charge/Co-Faculty

Program – 10

Develop a C program to accept a matrix of order m x n. Implement the following functions:

- a. Find the sum of each row
- b. Find the sum of each column
- c. Search an element in a matrix

Program:

```
#include <stdio.h>
```

```
void input_matrix(int matrix[][100], int m, int n)
{
    printf("Enter the elements of the matrix (%d x %d):\n", m, n);
    for (int i = 0; i < m; i++)
    {
        for (int j = 0; j < n; j++)
        {
            scanf("%d", &matrix[i][j]);
        }
    }
}
```

```
void sum_of_rows(int matrix[][100], int m, int n)
{
    int row_sum;

    printf("Sum of each row:\n");
    for (int i = 0; i < m; i++)
    {
        row_sum = 0;
        for (int j = 0; j < n; j++)
        {
            row_sum += matrix[i][j];
        }
        printf("Sum of row %d: %d\n", i + 1, row_sum);
    }
}
```

```
void sum_of_columns(int matrix[][100], int m, int n)
{
    int col_sum;
    printf("Sum of each column:\n");
    for (int j = 0; j < n; j++)
    {
        col_sum = 0;
        for (int i = 0; i < m; i++)
        {
```

```
        col_sum += matrix[i][j];
    }
    printf("Sum of column %d: %d\n", j + 1, col_sum);
}
}

void search_element(int matrix[][100], int m, int n, int element)
{
    int found = 0;
    for (int i = 0; i < m; i++)
    {
        for (int j = 0; j < n; j++)
        {
            if (matrix[i][j] == element)
            {
                printf("Element %d found at position (%d, %d)\n",
                    element, i + 1, j + 1);
                found = 1;
                break;
            }
        }
        if (found) break;
    }
    if (!found)
    {
        printf("Element %d not found in the matrix.\n", element);
    }
}

int main()
{
    int matrix[100][100], m, n, element;
    printf("Enter the number of rows (m) and columns (n): ");
    scanf("%d %d", &m, &n);

    input_matrix(matrix, m, n);
    sum_of_rows(matrix, m, n);
    sum_of_columns(matrix, m, n);

    printf("Enter an element to search for: ");
    scanf("%d", &element);
    search_element(matrix, m, n, element);

    return 0;
}
```

Test Cases:

Test No.	Input Parameters	Expected output	Obtained output	Remarks	Student Remarks
1	Enter the elements of the matrix (2 x 2): 1 2 3 4	Sum of each row: Sum of row 1: 3 Sum of row 2: 7 Sum of each column: Sum of column 1: 4 Sum of column 2: 6 Enter an element to search for: 2 Element 2 found at position (1, 2)	Sum of each row: Sum of row 1: 3 Sum of row 2: 7 Sum of each column: Sum of column 1: 4 Sum of column 2: 6 Enter an element to search for: 2 Element 2 found at position (1, 2)	PASS	
2	Enter the number of rows (m) and columns (n): 1 23	Sum of each row: Sum of row 1: 24 Sum of each column: Sum of column 1: 1 Sum of column 2: 23 Enter an element to search for: 25 Element 25 not found in the matrix.	Sum of each row: Sum of row 1: 24 Sum of each column: Sum of column 1: 1 Sum of column 2: 23 Enter an element to search for: 25 Element 25 not found in the matrix.	PASS	
Test for the following cases & record your observations					
3					

Rubrics for Continuous Evaluation of Experiments

Continuous Evaluation for 25 Marks		Marks Allotted	Marks Obtained
A	Observation Write-up and Attendance	05	
B	Conduction of Experiment and Output	10	
C	Viva Voice	05	
D	Record Write up	05	
Total		25	

Signature of Lab Faculty In-charge/Co-Faculty

Program – 11

Develop a C program to Check whether a String is Palindrome or not.

Program:

```
#include <stdio.h>
#include <string.h>

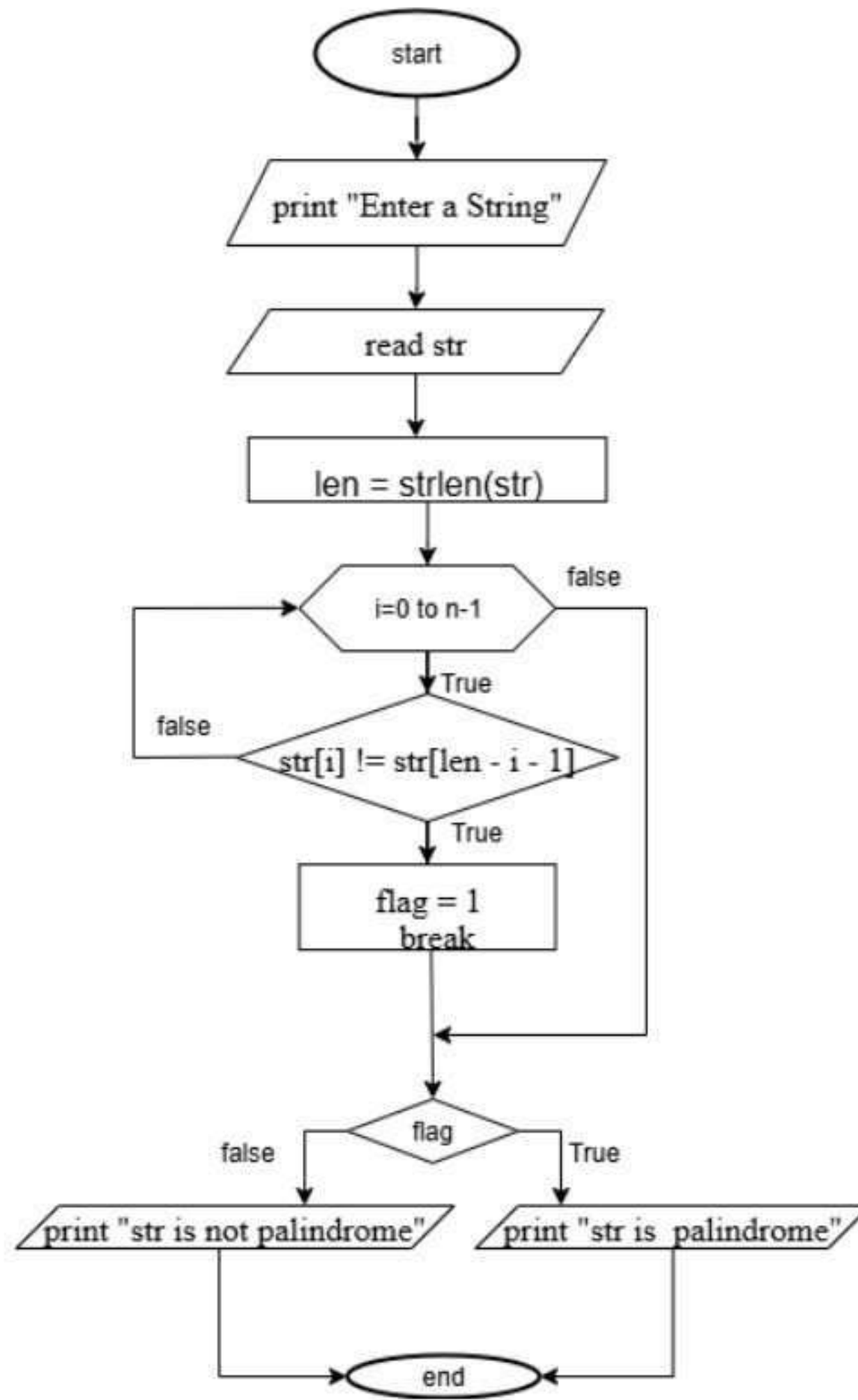
int main()
{
    char str[100];
    int i, len, flag = 0;
    printf("Enter a String\n");
    scanf("%s",str);

    len = strlen(str);

    for (i = 0; i < len; i++)
    {
        if (str[i] != str[len - i - 1])
        {
            flag = 1;
            break;
        }
    }

    if (flag)
        printf("%s is not palindrome", str);
    else
        printf("%s is palindrome", str);

    return 0;
}
```

Flow chart:

Test Cases:

Test No.	Input Parameters	Expected output	Obtained output	Remarks	Student Remarks
1	Enter a string: LIRIL	"LIRIL" is a palindrome.	"LIRIL" is a palindrome.	PASS	
2	Enter a string: ROHITH.	"ROHITH" is not a palindrome.	"ROHITH" is not a palindrome.	PASS	
Test for the following cases & record your observations					
3					
4					

Rubrics for Continuous Evaluation of Experiments

Continuous Evaluation for 25 Marks		Marks Allotted	Marks Obtained
A	Observation Write-up and Attendance	05	
B	Conduction of Experiment and Output	10	
C	Viva Voice	05	
D	Record Write up	05	
Total		25	

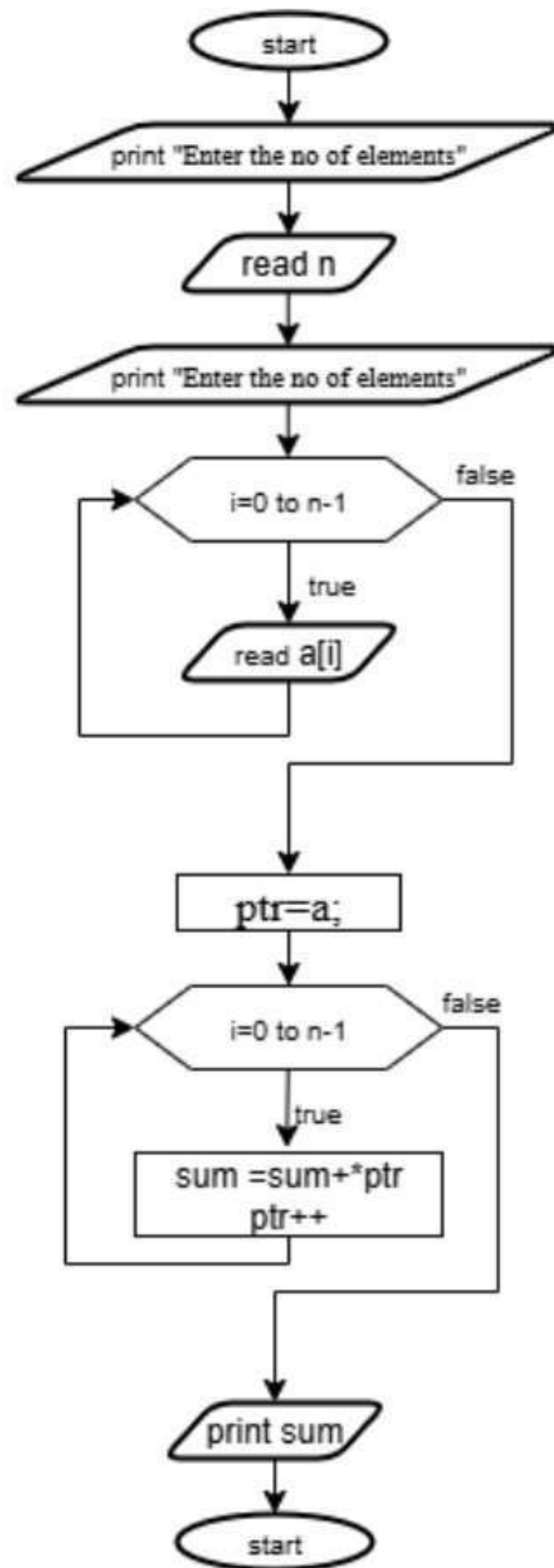
Signature of Lab Faculty In-charge/Co-Faculty

Program - 12

Develop a C program to find the sum of all the elements of an integer array using pointer.

Program:

```
#include<stdio.h>
#include<math.h>
int main()
{
    float a[10], *ptr, mean, std, sum=0;
    int n,i;
    printf("\n Enter the no of elements\n");
    scanf("%d",&n);
    printf("\n Enter the array elements\n");
    for(i=0;i<n;i++)
    {
        scanf("%f",&a[i]);
    }
    ptr=a;
    for(i=0;i<n;i++)
    {
        sum=sum+ *ptr;
        ptr++;
    }
    printf("\n Sum=%.3f\t",sum);
    return 0;
}
```

Flow Chart:

Test Cases:

Test No.	Input Parameters	Expected output	Obtained output	Remarks	Student Remarks
1	n=5 Array elements 1 5 9 6 7	sum=28	sum=28.000	PASS	
2	n=4 Array elements 2.3 1.1 4.5 2.7 8	sum=10.68	sum=10.680	PASS	
Test for the following cases & record your observations					
3					
4					

Rubrics for Continuous Evaluation of Experiments:

Continuous Evaluation for 25 Marks		Marks Allotted	Marks Obtained
A	Observation Write-up and Attendance	05	
B	Conduction of Experiment and Output	10	
C	Viva Voice	05	
D	Record Write up	05	
Total		25	

Signature of Lab Faculty In-charge/Co-Faculty

Program – 13

Program to read marks of three students in 3 subjects. Calculate the total marks scored, student wise and subject wise using structure.

Program:

```
#include <stdio.h>

struct Student
{
    int marks[3];
};

int main()
{
    struct Student students[3];
    int i, j;
    int totalStudent[3] = {0};
    int totalSubject[3] = {0};

    for (i = 0; i < 3; i++)
    {
        printf("Enter marks for student %d in 3 subjects:\n", i + 1);
        for (j = 0; j < 3; j++)
        {
            printf("Subject %d: ", j + 1);
            scanf("%d", &students[i].marks[j]);
        }
    }
    for (i = 0; i < 3; i++)
    {
        for (j = 0; j < 3; j++)
        {
            totalStudent[i] += students[i].marks[j];
            totalSubject[j] += students[i].marks[j];
        }
    }

    printf("\nTotal marks scored by each student:\n");
    for (i = 0; i < 3; i++)
    {
        printf("Student %d: %d\n", i + 1, totalStudent[i]);
    }

    printf("\nTotal marks scored in each subject:\n");
    for (j = 0; j < 3; j++)
    {
        printf("Subject %d: %d\n", j + 1, totalSubject[j]);
    }
}
```

```

}
return 0;
}

```

Test Cases:

Test No.	Input Parameters	Expected output	Obtained output	Remarks	Student Remarks
1	Enter marks for student 1 in 3 subjects: Subject 1: 100 Subject 2: 20 Subject 3: 45	Total marks scored by each student: Student 1: 165 Student 2: 544 Student 3: 93	Total marks scored by each student: Student 1: 165 Student 2: 544 Student 3: 93 Total marks scored in each subject: Subject 1: 150 Subject 2: 519 Subject 3: 133	PASS	
Test for the following cases & record your observations					
2					

Rubrics for Continuous Evaluation of Experiments

Continuous Evaluation for 25 Marks		Marks Allotted	Marks Obtained
A	Observation Write-up and Attendance	05	
B	Conduction of Experiment and Output	10	
C	Viva Voice	05	
D	Record Write up	05	
Total		25	

Signature of Lab Faculty In-charge/Co-Faculty

Program -14

Write a C program to copy the contents of one file to another using `fscanf()` and `fprintf()` functions.

Program:

```
#include <stdio.h>

#define BUFFER_SIZE 1024

int main()
{
    FILE *sourceFile, *destinationFile;
    char sourceFilename[256], destinationFilename[256];
    char buffer[BUFFER_SIZE];

    printf("Enter source filename: ");
    scanf("%s", sourceFilename);
    printf("Enter destination filename: ");
    scanf("%s", destinationFilename);

    sourceFile = fopen(sourceFilename, "r");
    if (sourceFile == NULL)
    {
        perror("Error opening source file");
        return 1;
    }

    destinationFile = fopen(destinationFilename, "w");
    if (destinationFile == NULL)
    {
        perror("Error opening destination file");
        fclose(sourceFile);
        return 1;
    }

    while (fscanf(sourceFile, "%1023s", buffer) != EOF)
    {
        fprintf(destinationFile, "%s ", buffer);
    }

    fclose(sourceFile);
    fclose(destinationFile);

    printf("File copied successfully.\n");

    return 0;
}
```

Test Cases:

Test No.	Input Parameters	Expected output	Obtained output	Remarks	Student Remarks
1	Enter source filename:1.txt Enter destination filename:2.txt	File copied successfully.	File copied successfully.	PASS	
2	Enter source filename:11.txt Enter destination filename:22.txt	Error opening source file	Error opening source file	PASS	
Test for the following cases & record your observations					
3					
4					

Rubrics for Continuous Evaluation of Experiments

Continuous Evaluation for 25 Marks		Marks Allotted	Marks Obtained
A	Observation Write-up and Attendance	05	
B	Conduction of Experiment and Output	10	
C	Viva Voice	05	
D	Record Write up	05	
Total		25	

Signature of Lab Faculty In-charge/Co-Faculty

Viva Voice Questions

1. What is an algorithm?
2. What are the characteristics of an algorithm?
3. What are the notations used while writing an algorithm?
4. What is flowchart?
5. List the symbols used while writing flowchart.
6. What is pseudocode?
7. What is ASCII?
8. What is high level language?
9. What is compiler?
10. What are tokens?
11. What are keywords? How many keywords are there in C programming language?
12. What is a variable?
13. What is the significance of a variable?
14. What are the rules to be followed while declaring a variable?
15. What is a constant?
16. What is a datatype? What are the different data types?
17. What are the basic or primary or fundamental datatypes supported by C?
18. What are escape sequence characters?
19. What are backslash constants? Name some constants.
20. List the size and range of basic data types.
21. What is the difference between a character and string containing a single character?
22. What is the meaning of associativity of an operator?
23. What is left associativity and right associativity?
24. What is implicit type conversion and explicit type conversion (type casting)?
25. What is precedence of an operator means?
26. List the precedence of all the types of operators along with associativity.
27. List the formatted input and output functions.
28. What is an expression?
29. What are the different types of expressions?
30. What is function?
31. What are the advantages of functions?
32. What are the different types of functions?
33. What are the elements of functions?
34. What is a library function?
35. What is calling function and called function?
36. What is the meaning of actual parameter and formal parameter?
37. What is function prototype or function declaration?
38. What is a function call?
39. Which are the logical operators?
40. Define decision making statement?
41. What are the varieties of if-statements?
42. What is the purpose of switch statement? Explain with syntax
43. What is loop? List the differences between pre-test and post-test loop
44. What are the advantages of loops?

45. What is control statement? What are the various types of control statements available in C
46. Explain for loop with syntax.
47. What is the difference between while and do-while loop?
48. What are unconditional control statements?
49. What is the use of break statement?
50. What is recursion? What are the advantages and disadvantages of recursion?
51. What is the difference between an ordinary variable and an array variable?
52. How an array will be initialized? Explain with an example.
53. How to declare an array variable?
54. What is bubble sort?
55. What is binary search?
56. What is multi-dimensional array?
57. What are the differences between recursion and iteration?
58. What is string? How strings are represented?
59. How the strings are stored in memory?
60. What is the difference between a character and string containing a single character?