Index

Sr. No.	Name of the Experiment	Page No.	Date of Experiment	Date of Submission	Remarks
1-	Write a C program to print "Hello	1	06/08/2025	13/08/2025	
2-	" " " " " In line	2	06/08/2025	13/08/2025	
3-	" " " , their age & Name	3	06/08/2025	13/08/2025	
4-	add two number	4	06/08/2025	13/08/2025	
5-	length and width	5	06/08/2025	13/08/2025	
6-	C → F	6	06/08/2025	13/08/2025	
7-	Check the triangle Valid or Not	7	13/08/2025	21/08/2025	
8-	Use AND, NOT & NOR on bit level	8	13/08/2025	21/08/2025	
9-	Apply left & Right shift	9	13/08/2025	21/08/2025	
10-	find the highest value of perimeter	10	13/08/2025	21/08/2025	
11-	Find the BMI Value	11	13/08/2025	21/08/2025	
12-	Check Collinear	12	13/08/2025	21/08/2025	
13-	Leep year	13	13/08/2025	21/08/2025	
14-	Use ternary Operator	14	20/08/2025	27/08/25	
15-	Count pos, Neg, Zero.	15	20/08/2025	27/08/25	
16-	Use loop & find Num* (2Num	16	20/08/2025	27/08/25	
17-	Generate set Output	17	20/08/2025	27/08/25	
18-	Determine the population at the end of each year.	18	20/08/2025	27/08/25	
		19	20/08/2025	27/08/25	
19-	Ramanujan Number	20	20/08/2025	27/08/25	
20-	Store it in a single array. find second largest number	21	27/08/2025	04/09/25	
		22	27/08/25	04/09/25	
21-	frequency of number	23	27/08/25	04/09/25	

10 Ramu 13/8/

Index

Sr. No.	Name of the Experiment	Page No.	Date of Experiment	Date of Submission	Remarks
22-					
23.	~~find factorial=1~~				
23-	find factorial help to array.	24	27/08/25	04/09/25	
		2			
24-	Recursive fun fibo(num).	25	27/08/25	04/09/25	
25-	find Prime Number	26	27/08/25	04/09/25	
26-	~~Delete~~ Develop a fun Reverse.	27	27/08/25	04/09/25	
27	Exn-5(iv) array	28	27/08/25	04/09/25	
28	Exn-5(v) array	29	27/08/25	04/09/25	
29-	Exp-7(i) - Structures & Unio	30	04/09/25	11/09/25	
30.	Exp-7(ii)- "	32	04/09/25	11/09/25	
33 -	exp-7(iii)- " "	33	04/09/25	11/09/25	
34-	Exp-7(iv) :- "	34	04/09/25	11/09/25	
35.	Exp-8(i) - Pointers	35	11/09/25	18/09/25	
36	Exp 8(ii) - pointers.	36	11/09/25	18/09/25	
37.	Exp 8(iii) - "	37	11/09/25	18/09/25	
38.	Exp9(i) - file Handling in(	38	18/09/25	25/09/25	
39.	Exp9(ii) "	39	18/09/25	25/09/25	
40.	Exp9(iii) "	40.	18/09/25	25/09/25	
41.	Exp10(i) Dynamic memory	41.	25/09/25	09/10/25	
42.	Exp10(ii) "	42	25/09/25	09/10/25	
43.	Exp12(i) Preprocessor & Directives in C	43	09/10/25	16/10/25	

INDEX

S.No.	Name of the Experiment	Page No.	Date of Experiment	Date of Submission	Remarks
44.	Exp 12(ii) -	44.	09/10/25	16/10/25	
45.	Exp 13 (i). Macros in C	45	16/10/25	23/10/25	
46.	Exp 14 (i)- Static libing	46	23/10/25	30/10/25	
47.	Exp 14 (ii) "	47	23/10/25	30/10/25	
48.	Exp 15 (i). Shared lib in C	48	30/10/25	06/11/25	
49.	Exp 15 (ii) "	49	30/10/25	06/11/25	
50.	Exp 9 (i) Gobal & local	50	04/09/25	20/11/25	
51.	Exp 9 (ii) " "	51	04/09/25	20/11/25	
52.	Exp 9 (iii) " "	52	04/09/25	20/11/25	
53.	Exp (iv) " "	53	04/09/25	20/11/25	

Expt.No: 1

Date

Page No. 1

Installation, Environment Setup and starting with C language.

1- Write a C program to print "Hello World"

→ What is program??

A program is basically a set of Instruction written in a programming language that tells a computer what to do.

→ Why # is used in C programming?

- The # tells the compiler:
  "Hey, this is a command for the preprocessor, not normal C code."
- Everything starting with # is not translated into machine code directly – it's processed first.

→ Why stdio.h is used in C programming?

In C programming, stdio.h is a header file that stands for Standard Output.

We use it because it contains the declarations (definitions) of functions and macros that allow us to.

Teacher's Signature : _____

main.c

Share Run

```c

#include <stdio.h>

int main() {

    printf("Hello world");

    return 0;
}
```

Output

```
Hello world

=== Code Execution Successful ===
```

2- Write a C program to print the address in multiple line [re

→ What is Programming function?
A function in programming is a block of code that perform
a specific task and can be used ["or "called"]
whenever needed.

In C programming -
A function has:
1- Name - to identify it (e.g., printf, main, sum).
2- Parameters - values you pass into it [Optional].
3- Return type - the type of value it gives back [e.g., int,
void].
4. Body - the block of code {.... } that does the wor

→. \n :- In C programming, \n is called a newline e
sequence.
→ "Move the cursor to the next line before printing the nex

Teacher's Signature : _____

Output

```
Hello
I am Vibha Dubey
Lucknow, Uttar Pradesh
India


=== Code Execution Successful ===
```

Expt.No:

Date

Page No. 3

3. Write a program that prompts the user to enter their name and age.

→ Why do we use [;] in C Progamming?

In C Programming, the ; (semicolon) is used to mark the end of a statement.

It tells the compiler:

→ "This instruction is complete. Now move to the next one."

Teacher's Signature : _____

```c
#include <stdio.h>

int main ( ) {
char name [50];
int age;


printf ("Enter your name :  ");
 fgets (name, sizeof(name), stdin);


printf ("Enter your age :  ");
 scanf ("%d", &age);

printf ("\n Hello %s You are %d years old. \n ", name, age);

 return 0;
 }
```

Output

```
Enter your name : Vibha Dubey
Enter your age :  18

Hello Vibha Dubey
 You are 18 years old.
```

Expt.No:

Date

Page No. 9

4. Write a C program to add two numbers, take number fro user.

→ What %d means ??

- % → says "a variable's value will be inserted here."
- d → stands for decimal integer (whole numbers in base 1

→ In scanf

```
scanf ("%d", &d);
```

- % tells scanf to expect an integer input from the u
- &a gives the address of a where the entered inte will be stored.

→ In printf

```
printf ("sum = %d, c);
```

- %d tells printf to display an integer value.
- c is the integer whose value gets printed in place of

main.c

Share Run

```c
#include <stdio.h>

int main() {
    int a, b, c;

    printf("Enter first number: ");
    scanf("%d", &a);

    printf("Enter second number: ");
    scanf("%d", &b);

    c = a + b;  // Addition after inputs

    printf("Sum = %d\n", c);

    return 0;
}
```

Output

```
Enter first number: 25
Enter second number: 56
Sum = 81


=== Code Execution Successful ===
```

- Operators -

Operators :-

In C programming, an operator is a symbol that tells the compiler to perform a specific on data. [Variables or values].

Eg.
- +, -, *, /, %.
- ==, !=, >, <, >=, <=

1- WAP a C program to calculate the area and perimeter of a rectangle based on its length and width.

→ What is use ~~in~~ float ??

- float is a data type in C used to store decimal numbers. Example: 5.5, 12.0, 3.14
- It allows more precision than integers for real numbers.

→ Why use %.f in this program??

- %.f is a format specifier in C for float or doubles values. Used in -
  - scanf ("%.f", &variable); → to read a float value from the
  - printf ("%.f", variable); → to display a float value.

Teacher's Signature

main.c

Share Run

```c
#include <stdio.h>

int main() {
    float length, width, area, perimeter;

    

    printf("Enter the length of the rectangle: ");
    scanf("%f", &length);

    printf("Enter the width of the rectangle: ");
    scanf("%f", &width);


    area = length * width;
    perimeter = 2 * (length + width);

    printf("\nArea of rectangle = %.2f", area);
    printf("\nPerimeter of rectangle = %.2f\n", perimeter);

    return 0;
}

```

Output

```
Enter the length of the rectangle: 12
Enter the width of the rectangle: 12

Area of rectangle = 144.00
Perimeter of rectangle = 48.00


=== Code Execution Successful ===
```

Expt.No:

Date

Page No. 6

2- WAP a C program to convert temperature from Celsius to Fahrenheit using the formula : F = (C * 9/5) + 32.

Teacher's Signature : _____

Share Run

main.c

```c
#include <stdio.h>

int main() {
    float celsius, fahrenheit;

    printf("Enter temperature in Celsius: ");
    scanf("%f", &celsius);

    fahrenheit = (celsius * 9 / 5) + 32;

    printf("Temperature in Fahrenheit: %.2f\n", fahrenheit);

    return 0;
}
```

Output

```
Enter temperature in Celsius: 42
Temperature in Fahrenheit: 107.60

=== Code Execution Successful ===
```

Expt.No: 3.1 – Conditional Statements –

Date 13/08/2025
Page No. 7

Conditional statements = Decision-making statements.
They check conditions [ture/false] and decide which block of code should run.

1.) WAP to take check if the triangle is valid or not. If the validity is established, do check if the triangle is isosceles, equilateral, right angle, or

Teacher's Signature :

```
#include <stdio.h>
 int main( )
 {
  int a, b, c ;
printf ["Enter three sides of the triangle "];
scanf("%d %d %d", &a , &b, &c );
if ((a+b>c) && (a+c>b) && (b+c>a))
{
   printf ("Triangle is valid.\n");
if (a==b && b==c)
{
   printf ("It is an Equilater triangle \n");
 }
  else if (a==b || b==c || a==c )
{
   printf ("It is Scalene triangle \n");
 }
    if ((a*a + b*b == c*c) ||
       (a*a + c*c == b*b) ||
       (b*b + c*c == a*a))
{
    printf("It is also a Right -angled triangle \n");
}
}
 else
{
   printf (" Triangle is NOT Valid \n");
}
   return 0;
}
```

Output → Enter three sides of the triangles : 10, 12, 25
Triangle is NOT Valid.

Expt.No: 3.1

Date 13/08/2025

Page No. 10

5. WAP using ternary operator, the user should input the length and breadth of a rectangle, one has to find out which rectangle has to the highest perimeter. The minimum number of rectangle should be three.

```c
#include <stdio.h>
int main()
{
    int l1, b1, l2, b2, l3, b3;
    int P1, P2, P3;
printf("Enter length and breadth of rectangle 1: ");
scanf("%d %d", &l1, &b1);
printf("Enter length and breadth of rectangle 2: ");
scanf("%d %d", &l2, &b2);
printf("Enter length and breadth of rectangle 2: ");
scanf("%d %d", &l3, &b3);

P1 = 2*(l1+b1);
P2 = 2*(l2+b2);
P3 = 2*(l3+b3);

int max = (P1>P2)?((P1>P3)?P1:P3):((P2>P3)?P2:P3);

printf("Perimeter of Rectangle 1 = %d\n", P1);
printf("Perimeter of Rectangle 2 = %d\n", P2);
printf("Perimeter of Rectangle 3 = %d\n", P3);

printf("The highest perimeter is: %d\n", max);

return 0;
}
```

```
#include <stdio.h>
int main()
{
    int l1, b1, l2, b2, l3, b3;
    int P1, P2, P3;
printf("Enter length and breadth of rectangle 1: ");
scanf("%d %d", &l1, &b1);
printf("Enter length and breadth of rectangle 2: ");
scanf("%d %d", &l2, &b2);
printf("Enter length and breadth of rectangle 2: ");
scanf("%d %d", &l3, &b3);

P1 = 2*(l1+b1);
P2 = 2*(l2+b2);
P3 = 2*(l3+b3);

int max = (P1>P2)?((P1>P3){P1:P3}:((P2>P3)?P2:P3);

printf("Perimeter of Rectangle 1 = %d\n", P1);
printf("Perimeter of Rectangle 2 = %d\n", P2);
printf("Perimeter of Rectangle 3 = %d\n", P3);

printf("The highest perimeter is: %d\n", max);

return 0;
}
```

→ Output :-

Enter length and breadth of rectangle 1 : 12
12
Enter length and breadth of rectangle 2 : 13
13
Enter legth and breadth of rectangle 3 : 14
14
Perimeter of Rectangle 1 = 48
Perimeter of Rectangle 2 = 52
Perimeter of Rectangle 3 = 56

The highest perimeter is : 56

Expt.No: 8.1

Date 13/09/2025

Page No. 11

2. WAP to compute the BMI Index of the person and print the BMI values as per the following ranges. You can use the following formula to compute BMI = weight (kgs) / Height (Mts) * Height(M

$$BMI = \frac{\text{Weight (Kgs)}}{\text{Height * Height (mts)}}$$

```c
#include <stdio.h>
int main()
{
    float weight, height, bmi;
    printf("Enter weight :");
    scanf("%f", &weight);

    printf("Enter height: ");
    scanf("%f", &height);

    bmi = weight / (height * height);
    printf("your BMI = %.2f\n", bmi);

if (bmi<15) {
    printf("Category starvation\n");
}
  else if (bmi >= 15.1 && bmi<=17.5) {
      printf("Category : Anorexic\n");
  }
  else if (bmi >=17.6 && bmi <=24.9) {
      printf("Category : Underweight\n");
  }
  else if (bmi >=25 && bmi <=25.9) {
      printf("Category : Ideal\n");
  }
else if (bmi >= 25 && bmi<=25.9) {
    printf("Category: Overweight");
}
else if (bmi >= 30 && bmi<=39.9) {
    printf("Category : obese\n");
}
```

```
28.  }
29.  else if (bmi >=40.0) {
30.      printf ("category: Morbidly obese\n");
31.  }
32.  else
33.  {
34.      printf ("Invalid BMI Range\n");
35.  }
36.    return 0;
37.  }
```

Output :-

```
Enter the height in meter
    5.5
Enter the weight in kg
    68
BMI = 68.000
category: Morbidly obese
```

Teacher's Signature :

Expt.No: 3.1

Date 13/08/2025

Page No. 12

3. WAP to check if three points (x_1, y_1), (x_2, y_2) and (x_3, y_3) are collinear or not.

If area of triangle formed by 3 points is 0, then points are collinear.

$$\text{Area} = \frac{1}{2} \times [x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2)]$$

If area = 0 → Points are collinear

Else → Not collinear.

```
1.  #include<stdio.h>
2.  int main()
3.  {
4.  int x1, y1, y2, x2, x3, y3;
5.  int area;
6.  printf("Enter the point 1 (x1, y1): ");
7.  scanf("%d", &x1, &y1);
8.  printf("Enter the point 2 (x2, y2): ");
9.  scanf("%d", &x2, &y2);
10. printf("Enter the point 3 (x3, y3): ");
11. scanf("%d", &x3, &y3);

12. area = x1*(y2-y3) + x2*(y3-y1) + x3*(y1-y2)

13. if(area == 0)
14. {
15. printf("The points are collinear.\n");

16. else
17. {
18. printf("The points are NOT collinear.\n");
19. return 0;
20. }
```

→

Output :-

Enter the value of x_1 : 4
Enter the value of x_2 : 2
Enter the value of x_3 : 3
Enter the value of y_1 : 7
Enter the value of y_2 : 2
Enter the value of y_3 : 3

→ Not collieax

Expt.No: 3.1

Date

Page No. 13

4. According to the gregorian calendar, it was Monday on the date 01/01/01. If Any year is Input through the keyboard write a program to find out what is the day on 1st January of this year.

Teacher's Signature : _____

```
#include<stdio.h>
int main() {
  printf("Enter year");
  scanf("%d", &year);

  for(int y=1; y<year; y++) {
      if(isLeap(y))
          days += 366;
      else
          days += 365;
  }
    int day = days % 7;
    char [] = {"Mon", "Tue", "Wed", "Thus", "fri",
                "Sata", "Sun"};
  printf("1st January of year %d is %s\n", year, week
          Days[dayIndex]);

      return 0;
  }
```

→ Output

Enter the year : 2025

The day on 1st January 2025 is : Wednesday

Expt.No: 3.1

Date

Page No. 17

5- WAP using ternary operator the user should input the length & breadth of a rectangle, one has to find out which rectangle has the highest perimeter. The minimum number of rectangle should be three.

→ Ternary Operator :-

condition ? expression : expression2 ;

Teacher's Signature : _____

```c
#include <stdio.h>

int main() {
    float l1, b1, l2, b2, l3, b3;
    float p1, p2, p3, max;
    printf("Enter length and breadth of Rectangle 1: ");
    scanf("%f %f", &l1, &b1);
    printf("Enter length and breadth of Rectangle 2: ");
    scanf("%f %f", &l2, &b2);
    printf("Enter length and breadth of Rectangle 3: ");
    scanf("%f %f", &l3, &b3);
    p1 = 2 * (l1 + b1);
    p2 = 2 * (l2 + b2);
    p3 = 2 * (l3 + b3);
    max = (p1 > p2) ? ((p1 > p3) ? p1 : p3) : ((p2 > p3) ? p2 : p3);
    if (max == p1)
        printf("Rectangle 1 has the highest perimeter: %.2f\n", p1);
    else if (max == p2)
        printf("Rectangle 2 has the highest perimeter: %.2f\n", p2);
    else
        printf("Rectangle 3 has the highest perimeter: %.2f\n", p3);

    return 0;
}
```

Output :-

Enter length and breadth of Rectangle 1 : 5 3
Enter length and breadth of Rectangle 2 : 6 4
Enter length and breadth of Rectangle 3 : 2 10
Rectangle 3 has the highest perimeter : 24

Expt.No: 3.2

Loops

Page No. 15

1- WAP to enter numbers till the user wants. At the end, It should display the count of positive, negative & zero entered.

Loop :- A loop is used to execute a set of statements repeatedly until a certain condition is met.

→ Why loops are used :

loops help when you want to do something again & again, like..

- Printing numbers from 1 to 100.

```c
#include <stdio.h>
int main() {
    int num, positive = 0, negative = 0, zero = 0;
    char choice = 'y';
    while (choice == 'y' || choice == 'Y') {
        printf("Enter a number: ");
        scanf("%d", &num);
        if (num > 0)
            positive++;
        else if (num < 0)
            negative++;
        else
            zero++;
        printf("Do you want to enter another number? (y/n): ");
        scanf(" %c", &choice);
    }
    printf("\nTotal Positive numbers: %d\n", positive);
    printf("Total Negative numbers: %d\n", negative);
    printf("Total Zeroes: %d\n", zero);

    return 0;
}
```

Output -

Enter a number : 12 3 5 0 -9 -7 -2 4

Do you want to enter another number? (Y/n)

n

count of ~~Physics~~ Positive numbers : 4

count of Negative number : 3

count of zero : 1

Teacher's Signature :

2. WAP to print the multiplication table of the number entered by the user. It should be in the correct formatting. Num * 1 = Num

→ For loop :- Used when you know how many times you want to repeat.

```
for(initialization ; condition ; increment/decrement) {
    // code to repeat
}
```

Teacher's Signature : _____

```c
#include <stdio.h>
int main(){
    int n,i;
    printf("Enter the value of n\n");
    scanf("%d",&n);
    for(int i=1;i<=10;i++)
    {
        printf("%d*%d=%d\n",n,i,n*i);
    }
    return 0;
}
```

Output:

```
Enter the value of n: 5
5*1=5
5*2=10
5*3=15
5*4=20
5*5=25
5*6=30
5*7=35
5*8=40
5*9=45
5*10=50
```

Expt.No: 3-2

Page No. 17

3- WAP to generate the following set of output:

a.
```
1
2 3
4 5 6
```

```c
#include <stdio.h>

int main() {
    int i, j, num = 1;
    for(i = 1; i <= 3; i++) {
        for(j = 1; j <= i; j++) {
            printf("%d ", num);
            num++;
        }
        printf("\n");
    }
    return 0;
}
```

Output :-

1

2 3

4 5 6

Expt.No:

Date

Page No. 18

4. The population of a town is 100000. The population has increased steadily at the rate of 10% per year for the last 10 years. Write a program to determine the population at the end of each year in the last decade.

Teacher's Signature :

```c
#include <stdio.h>

int main() {
    float population = 100000;
    int year;
    printf("Year\t\tPopulation\n");
    for (year = 1; year <= 10; year++) {
        population = population + (population * 0.10);
        printf("%d\t\t\t%.0f\n", year, population);
    }
    return 0;
}
```

Output :-

Year	Population
1	110000
2	121000
3	133100
4	146410
5	161051
6	177156
7	194872
8	214359
9	235795
10	259374

5. Ramanujan Number is the smallest number that can be expressed as the sum of two cubes in two different ways. WAP to print all such numbers up to reasonable limit.

Ramanujan Number:-

Teacher's Signature :

```c
#include <stdio.h>
int main() {
    int L, a, b, c, d;
    printf("Enter the limit L: ");
    scanf("%d", &L);
    printf("Ramanujan numbers upto %od are:\n", L);
    for (int num = 1; num <= L; num++) {
        int count = 0;
        for (a = 1; a*a*a < num; a++) {
            for (b = a + 1; b*b*b < num; b++) {
                if (a*a*a + b*b*b == num)
                    count++;
            }
        }
        if (count == 2)
            printf("%d ", num);
    }
    return 0;
}
```

Output → 1.)

Enter the limit L: 20

Ramanujan numbers upto 20 are:

2.)

2000

1729

Expt.No: 5

Date

Page No. 21

-'Array'-

1. WAP to read a list of integers and store it in a single dimensional array. Write a C program to print the second largest integer in a list of integers.

→ Array :- An array is a collection of similar data items stored at consecutive memory locations and accessed using a single name.

⇒ Why use arrays ?

- To store many values under one name
- To easily access or modify data using an index
- To avoid using multiple separate variables like int a, b, c, d, e.

Teacher's Signature :

```c
#include <stdio.h>

int main() {
    int n, i, j, temp;
    int arr[100];

    printf("Enter the number of elements: ");
    scanf("%d", &n);
    printf("Enter %d integers:\n", n);
    for (i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }
    for (i = 0; i < n - 1; i++) {
        for (j = 0; j < n - i - 1; j++) {
            if (arr[j] > arr[j + 1]) {
                temp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = temp;
            }
        }
    }
    printf("\nThe second largest number is: %d\n", arr[n - 2]);
    return 0;
```

Expt.No: 5

Date

Page No. 22

2- WAP to read a list of integers and store it in a single dimensional array. Write a C program to count & display positive, negative, odd & even numbers in an array.

Teacher's Signature :

```c
#include <stdio.h>
int main() {
int Pos=0, Neg=0, odd=0, Even=0;
int a[50], n, i;
printf("Enter number of element ");
scanf("%d", &n);
printf("Enter %d integers \n", n);
for(i=0; i<n; i++) {
   scanf("%d", &a[i]);
}
  for(i=0; i<n; i++) {
  if(a[i] >=0)
     Pos++;
  else
       Neg++;
   if(a[i]%2==0)
     Even++;
   else
       odd++;
  }
  printf(" Pasitive numbers : %d\n", Pos);
  printf(" Negative numbers : %d\n", Neg);
  printf(" Even numbers : %d\n", Even);
  printf(" Odd numbers : %d\n", Odd);

  return 0;
}
```

Output :- Enter number of element : 5

Enter5integers: 1, -3 2 -4 5

Pos = 3

Neg = 2

Even = 2

Odd = 3

3- WAP to read a list of integers and store it in a single dimensional array. Write a program to find the frequency of a particular number in a list of integers.

Teacher's Signature : _____

```c
#include <stdio.h>
int main() {
    int a[100], n, i, num, count = 0;
    printf("Enter the number of elements: ");
    scanf("%d", &n);
    printf("Enter %d integers:\n", n);
    for (i = 0; i < n; i++) {
        scanf("%d", &a[i]);
    }
    printf("Enter the number whose frequency you want to find: ");
    scanf("%d", &num);
    for (i = 0; i < n; i++) {
        if (a[i] == num)
            count++;
    }
    printf("The frequency of %d is: %d\n", num, count);
    return 0;
}
```

Output :-

```
Enter the number of elements : 6
Enter 6 integers:
1 1 3 3 4 1
Enter the number whose frequency you want to
find : 1

The frequency of 1 is: 3
```

Expt.No: 5

Page No. 27

4. WAP that reads two matrics A(mxn) & B(pxq) and computes the product A and B. Read matix A and matix B in row major order respectively. Print both the input matrices and resultant matix with suitable headings and output should be in matrix format only. Program must check the compatibillity of orders of the matrices for multiplication. Report appropriate message in case of incompatibillity.

```
#include <stdio.h>
int main ( ) {
int A[10][10], B[10][10], C[10][10];
int m, n, p, q;
int i, j, k;
print ("Enter the number of rows and columns of maxtrix A:");
Scanf ("%d %d", &m, &n);
print ("Enter the number of rows and columns of matrix B:");
scanf ("%d %d", &p, &q);
if (n! =p) {
printf("\n Matrix multiplication not possible!\n");
printf (" Reason : Number or columns of A (%d)! = Number of rows
        of (%d)\n", n,p);
return 0;
}
printf ("\n Enter elements of matrix A (%d x %d) in row
        major order:\n", m,n);
for(i = 0 ; i<m; i++) {
for (j=0 ; j<n ; j++) {
  scanf ("%d", &B[i][j]);
}
}
  for (i=0 ; i<m; i++) {
   for (j=0 ; j<q ; j++) {
    c[i][j] = 0;
   for(k = 0; k<n; k++) {
    c[i][j] + = A[i][k] * B[k][j];
}
}
}
```

```
printf("\n Matrix A: \n");
for(i=0; i<m; i++) {
  for(j=0; j<n; j++) {
   printf("%5d", A[i][j]);
  }
   printf("\n");
}
printf(" Resultant Matrix (AxB): \n");
for(i=0; i<m; i++) {
for(j=0; j<q; j++) {
  printf("%5d", C[i][j]);
}
  printf("\n");
}
  return 0;
}
```

Teacher's Signature :

Output :-

```
Enter rows and columns of Matrix A: 2 3
Enter rows and columns of Matrix B: 3 2

Enter matrix A:
1 2 3
4 5 6
Enter matrix B:
7 8
9 10
11 12

Resultant Matrix (A x B):
58 64
139 154
```

Expt.No: 6-

- Function -

Date

Page No. 25

1- Develop a recursive & non-recursive function fact(num) 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, write a C program to compute the binomial coefficient. Tabulate the results for different values of n and r with suitable messages.

```
#include <stdio.h>
long long fact_recursive(int n) {
  if(n==0)
   return 1;
  else
  return n*fact_recursive(n-1);
  }
long long fact_nonrecursive(int n) {
   long long f=1;
  for(int i = 1 ; i<=n; i++)
     f *= i;
  return f;
 }
long long binomial(int n, int r) {
  return fact_recursive(n) / (fact_nonrecursive(r)*fact_nonrecursive
                                                    (n-r));
  }
int main() {
  int n,r;
printf("Enter value of n:");
  scanf("%d", &n);
  printf("Ente value of r");
  scanf("%d", &r);
  if (r>n) {
  printf("Invalid Input ! r cannot be greater than n\n");
  return 0;
}
printf("Using recursive and non_recursive fact fun:\n");
Printf("Factorial of %d = %lld\n", n, fact_recursive(n));
Printf("Factorial of %d = %lld\n", n, fact_nonrecursive(n));
```

```
long long result = binomial(n,r);
printf("\n Binomial coefficient C(%d,%d) = %d\n", n, r, result);
printf("\n Table of C(n,r) for given n:\n");
printf("-------\n");
for (int i=0; i<=n; i++)
    printf("%d\t%lld\n", i, binomial(n,i));
return 0;
}
```

➡ Output -

```
Using recursive and non-recursive factorial functions:
    factorial of 5 = 120
    factorial of 5 = 120

Binomial coefficient C(5,2) = 10

Table of C(n,r) for given n:
```

r	C(n,r)
0	1
1	5
2	10
3	10
4	5
5	1

Teacher's Signature : _____

Expt.No: 6

Date

Page No. 24

2. Develop a recursive function GCD (num1, num2) that accepts two integer arguments. Write a C program that invokes this function to find the greatest common divisor of two given integers.

```c
#include <stdio.h>
int GCD(int num1, int num2) {
    if (num2 == 0)
        return num1;
    else
        return GCD(num2, num1 % num2);
}
int main() {
    int a, b, result;
    printf("Enter two integers: ");
    scanf("%d %d", &a, &b);
    if (a < 0) a = -a;
    if (b < 0) b = -b;
    result = GCD(a, b);
    printf("\nThe GCD of %d and %d is: %d\n", a, b, result);
    return 0;
}
```

utput -

Enter two integers: 4 8

The GCD of 4 & 8 is : 4

Expt.No:

Date

Page No. 27

3- Develop a recursive function fibo (num) that accepts an integer argument. Write a C program that invokes this function to generate the fibonacci sequence up to num.

Teacher's Signature : _____

```c
#include <stdio.h>
int FIBO(int num) {
    if (num == 0)
        return 0;
    else if (num == 1)
        return 1;
    else
        return FIBO(num - 1) + FIBO(num - 2);
}
int main() {
    int n, i;
    printf("Enter the number of terms: ");
    scanf("%d", &n);
    printf("\nFibonacci sequence up to %d terms:\n", n);
    for (i = 0; i < n; i++) {
        printf("%d ", FIBO(i));
    }
    printf("\n");
    return 0;
}
```

Expt.No: 5

Date

Page No. 28

4- Develop a C function ISPRIME (num) that accepts an integer argument and returns 1 if the argument is prime, a 0 otherwise. Write a C Program that invokes this function to generate prime numbers between the given ranges.

```c
int ISPRIME(int num) {
    int i;
    if (num <= 1)
        return 0;
    for (i = 2; i <= num / 2; i++) {
        if (num % i == 0)
            return 0;
    }
    return 1;
}
int main() {
    int start, end, i;
    printf("Enter the starting range: ");
    scanf("%d", &start);
    printf("Enter the ending range: ");
    scanf("%d", &end);
    printf("\nPrime numbers between %d and %d are:\n", start,
        end);
    for (i = start; i <= end; i++) {
        if (ISPRIME(i))
            printf("%d ", i);
    }
    printf("\n");
```

Expt.No: 6

Date

Page No. 23

5. Develop a function REVERSE (str) that accepts a string arguments. Write a C program that invokes this function to find the reverse of a given string

Teacher's Signature :

```c
#include <stdio.h>
#include <string.h>
void REVERSE(char str[]) {
    int i, j;
    char temp;
    int len = strlen(str);
    for (i = 0, j = len - 1; i < j; i++, j--) {
        temp = str[i];
        str[i] = str[j];
        str[j] = temp;
    }
}
int main() {
    char str[100];
    printf("Enter a string: ");
    gets(str);
    REVERSE(str);
    printf("\nReversed string: %s\n", str);
    return 0;
}
```

Expt. No. — Structures and Union —

Date

Page No. 30

1.1 Write a C program that uses function to perform the following operations:

a.) Reading a complex number.

b.) Writing a complex number.

c.) Addition and subtraction of two complex numbers

→ Structure:- A container that holds different types of data together.

Example - A student has

- Name (string)
- Age (int)
- Marks (float)

→ Union: A union allows you to store different data ty but only one at a time, because all members share the same memory location.

Example -

id → 4 bytes

marks → 4 bytes

name → 20 bytes.

Teacher's Signature :

```
#include <stdio.h>
struct complex
{
    float real;
    float imag;
};
struct complex c;
printf("Enter real part:");
scanf("%f", &c.real);
print("Enter imaginary part:");
scanf("%f", &c.imag);
return 0;
}

void write complex(struct complex c)
{
  if(c.imag >= 0)
printf("%.2f + %.2f i\n", c.real, c.imag);
} else
printf("%.2f - %.2f i\n", c.real, -c.imag);

struct complex add complex(struct complex a, struct
{

struct complex result;
result.real = a.real + b.real;
result.imag = a.imag + b.image
return result;
}

int main()
{
```

```
    struct complex c1, c2, sum, diff;
    printf("Enter first complex number: \n");
    c1 = read complex();
  printf("Enter second complex numbers \n");
    c2 = read complex();
    sum = add complex(c1, c2);
    diff = subtract complex(c1, c2);

  printf("\n first complex Number: ");
    write complex(c1);
  printf("\n second complex Number: ");
    write complex(c2);
  printf("\n Addition: ");
    write complex(sum);
  printf("Subtraction: ");
    write complex(diff);
    return 0;
}.
```

2- Write a C program to complete the monthly pay of 100 employees using each employee's name, basic pay. The DA is computed as 52% of the basic pay. Gross salary (basic pay + DA). Print the employees name and gross salary.

Structure	Union
All members have separate memory → can store all values at once.	All members share the same memory → can store only one value at a time.

Teacher's Signature :

```c
//2.    Write a C program to compute the monthly pay of 100 employees using each employee_s name, basic pay. The DA is com
#include <stdio.h>
struct emp {
    char name[50];
    float basic_pay;
    float da;
    float gross_sal;
};
int main() {
    struct emp a[100];
    for(int i = 0; i < 100; i++) {
        printf("Enter name of employee %d: ", i+1);
        scanf("%s", a[i].name);
        printf("Enter basic pay of employee %d: ", i+1);
        scanf("%f", &a[i].basic_pay);
        a[i].da = 0.52 * a[i].basic_pay;
        a[i].gross_sal = a[i].basic_pay + a[i].da;
    }
    printf("\n--- Employee Salary Details ---\n");
    for(int i = 0; i < 100; i++) {
        printf("\nEmployee %d\n", i+1);
        printf("Name        : %s\n", a[i].name);
        printf("Basic Pay   : %.2f\n", a[i].basic_pay);
        printf("DA (52%%)    : %.2f\n", a[i].da);
        printf("Gross Pay   : %.2f\n", a[i].gross_sal);
    }
    return 0;
}
```

Output -

Enter name of Employee 1 : Ramesh
Enter basic pay of Employee 1 : 350
Enter name of Employee 2 : Mukesh
Enter basic pay of Employee 2 : 320
Enter name of Employee 3 : Gukesh
Enter basic pay of Employee 3 : 300

--- Employee Salary Details ---

Employee 1
Name : Ramesh
Basic Pay : 350.00
DA (52%) : 182.00
Gross Pay : 532.00

Employee 2
Name : Mukesh
Basic Pay : 320.00
DA (52%) : 166.40
Gross Pay : 486.40

Employee 3
Name : Gukesh
Basic pay : 300.00
DA (52%) : 156.00
Gross Pay : 456.00

3- Create a Book structure containing book_id, title, author name and price. Write a C program to pass a structure as a function argument and print the book's details.

→ When should you use a union??

Use Union when:

- You want to save memory.
- You need variables that will be used one at a time.

Teacher's Signature :

exp7c3.c > ...

```c
//3.    Create a Book structure containing book_id, title, author name and price. Write a C program to pass a structure a
#include <stdio.h>
struct Book {
    int book_id;
    char title[40];
    char author_name[30];
    int price;
};
void printDetails(struct Book b);
int main() {
    struct Book b1 = {5900, "Math", "Ravish Gupta", 450};
    printDetails(b1);
    return 0;
}
void printDetails(struct Book b)
{
    printf("\nBook Details:\n");
    printf("Book ID       : %d\n", b.book_id);
    printf("Title         : %s\n", b.title);
    printf("Author Name   : %s\n", b.author_name);
    printf("Price         : %d\n", b.price);
}
```

Output -

Book Details : ς

Book ID : 5900
Title : Math
Author Name: Ravish Gupta
Price : 450

Expt.No: 7

Date

Page No. 34

a. Create a union containing 6 strings : name, home_address, hostel_address, city, state and zip. Write a C program to display your present address.

Teacher's Signature : _____

Output :

Present Address :

Belwai

Sultanpur

Uttar Pradesh

228171

Expt.No:- 8

Date

Page No. 35

POINTERS -

1.1 Declare different types of pointer (int, float, char) and initialize them with the addresses of variables. Print the values of both the pointers and the variables they point to.

>> Pointer :- Address storing variable.
Instead of storing a value, it stores where the value is located in memory.

Pointer	stores memory address.
&a	Address of variable a.
*p	Value at address stored in pointer

>> Why use ??
Fast, efficient, dynamic memory, data structures.

Teacher's Signature : _____

Output -

Values of Variables :

a = 10

b = 3.14

c = X

Values stored in pointers [addresses]:

P_1 = 0061FF1B

P_2 = 0061FF0C

P_3 = 0061FF0B

Values accessed through pointers:

$*P_1$ = 10

$*P_2$ = 3.14

$*P_3$ = X

2. Perform pointer arithmetic (increment and decrement) on pointers of different data types. Observe how the memory addresses change and the effects on data access.

⇛ Why do we use pointers in a program?

i). Accessing memory directly.
2.) Passing large data to function efficiently
3.) Dynamic Memory Allocation
4.) Creating complex data structures.
→ linked list
→ Trees.
→ Graphs
5) Array handling.

Teacher's Signature :

C exp8c2.c > ...

```c
//2.    Perform pointer arithmetic (increment and decrement) on pointers of different data types. Observe how the
#include <stdio.h>
int main() {
    int a = 10;
    float b = 5.5;
    char c = 'A';
    int *ip = &a;
    float *fp = &b;
    char *cp = &c;
    printf("Original addresses:\n");
    printf("ip = %p\n", ip);
    printf("fp = %p\n", fp);
    printf("cp = %p\n", cp);
    ip++;
    fp++;
    cp++;
    printf("\nAfter increment:\n");
    printf("ip = %p\n", ip);
    printf("fp = %p\n", fp);
    printf("cp = %p\n", cp);

    return 0;
}

```

Output:

Original addresses:

ip = 0061FF10

fp = 0061FF0C

Cp = 0061FF0B

After increment:

ip = 0061FF14

fp = 0061FF10

fp = 0061FF0C

Expt.No: 8

Date

Page No. 37

3- Write a function that accepts pointers as parameters. Pass variables by reference using pointers and modify their values within the function.

Teacher's Signature :

Output:

Before function call:

a = 5 :

b = 3.50

After function call:

a = 15

b = 7.00

Expt.No: - 9

Date

Page No. 38

File Handling in C

1.) Write a program to create a new file write text into it.

» File handling :- Saving data to files and retrieving it later using C programs.

» Why use ??

Because Variables store data temporarily (in Ram).

But files store data permanently on disk

Signature :

C exp9c1.c > main()

```c
//1.    Write a program to create a new file and write text into it.
#include <stdio.h>
int main() {
    FILE *fp;
    fp = fopen("myfile.txt", "w");
    if (fp == NULL) {
        printf("Error! File cannot be created.\n");
        return 1;
    }
    fprintf(fp, "This is a sample text written into the file.\n");
    fprintf(fp, "File Handling Experiment in C.");
    fclose(fp);
    printf("File created and text written successfully!\n");
    return 0;
}
```

Output :

File created and text written successfully !

Expt.No: 9

Date

Page No. 39

2-> Open an existing file and read its content character by character, and then close the file.

Use cases -

- Saving records [student, employees, expenses]
- Reading configuration files
- Logging errors
- Creating reports

Teacher's Signature :

Output :

This is a sample text written into the file.
File Handling Experiment in C.

Expt.No:

Date

Page No. 40

3.) Open a file, read its content line by line, and display each line on the console.

» File Operations in C

(i)- Opening a file
Using : fopen()

(ii)- Reading from a file
Using: fgetc(), fgets(), fscanf()

(iii)- Writing to a file.
Using: fputc(), fputs(), fprintf()

(iv)- Closing the file
Using: fclose()

Mode	Meaning
"r"	Read
"w"	Write
"a"	Append
"r+"	Read + Write
"w+"	Read + Write (overwrite)

Teacher's Signature : _____

1.)
```c
#include <stdio.h>
int main()
{
FILE *file;
char line[256];
File = fopen("Error Opening file!\n");
return 1;
}
While (fgets(line, sizeof(line), file)) {
    printf ("%.s", line);
}
fclose (file);
return 0;
}
```

Output :

```c
#include <stdio.h>
int main()
{
printf(" Hello world");
return 0:
}
```

Expt.No:- 10

Dynamic Memory Allocation

Date

Page No. 41

1.) Write a program to create a simple linked list in C using pointer and structure.

» Dynamic memory allocation : Memory is given when the program is running, not before.

→ Why use dynamic memory allocation ??

- We don't know the required memory size before running the program.
- The size may change based on user input.
- We want to use memory efficiently and avoid wasting space.

Teacher's Signature :

1.)
```c
#include <stdio.h>
#include <stdlib.h>

  struct Node {
      int data ;
   Struct Node* next;
};
int main() {
struct Node *head = NULL, * second = NULL, * third = NULL;
  head = (struct Node*) malloc (sizeof (struct Node));
  second = (struct Node*) malloc (sizeof (struct Node));
  third = (struct Node* ) malloc (sizeof (struct Node));

   head -> data = 20;
   head -> next = second;
   head -> data = 20;
   second -> next = third;
   third -> data = 30;
   third -> next = NULL;
   Struct Node* temp = head;
   printf ("Linked List: ");
   while (temp != NULL) {
      printf ("%d -> ", temp -> data );
    temp = temp -> next;
  }
   printf (" NULL ");
  return 0;
  }
```

Output –

Linked list : 10 -> 20 -> 30 -> NULL

Expt.No: 16

Date

Page No. 42

2. Write a program to insert item in middle of the linked list.

>> Function used in Dynamic memory allocation :-

Function	Propose
malloc ( )	Allocates memory
calloc ( )	Allocates memory + initialises with 0
realloc ( )	Change (increases / decreases) allocated m
free ( )	Releases memory back to system

```
#include <stdio.h>
#include <stdlib.h>
 struct Node {
  int data;
  struct Node * next;
};
Void insertMiddle  Node* head, int data, int position) {
struct Node * newNode = (struct Node*) malloc (sizeof (Struct Node));
 newNode -> data = data;
 struct Node* temp = head;
  int i = 1;
 While (i< position && temp != NULL) {
     temp =  temp -> next;
       i++;
   }
    if (temp == NULL) {
     printf ("Position out of range!\n");
    return 0;
    }
    newNode -> Next = temp -> next;
     temp -> next = newNode;
    }
     int main () {
      Struct Node * head = NULL, * second = NULL , * third = NULL;
     head = (Struct Node*) malloc (sizeof (Struct Node));
     second = (struct Node* ) malloc (sizeof (Struct Node));
     third  = (Struct Node* ) malloc (sizeof(Struct Node));
    head -> data = 10;  head -> next = second;
    second -> data = 20; second -> next = third;
    third -> data = 30;  third -> next = NULL;
    printf ("Original List: 10 -> 20 -> 30 -> NULL\n");
    insertMiddle (head, 25, 2);
     struct Node * temp = head;
     printf (" Updated List :");
     While (temp != NULL) {
      Printf ("%d -> ",
```

```
    temp = temp -> next;
}
    printf ("NULL");
    return 0;
}
```

Output ➡

Original list : 10 → 20 → 30 → NULL

Updated list : 10 → 20 → 30 → NULL

Expt.No: 11 : — Bitwise Operator —

Date 13/08/2025
Page No. 8

Bitwise operators are special operators in programming (like C, C++ ...) that work directly on the binary [bit level] representation of numbers.

Types of Bitwise Operators :

1.) AND (&)
2.) OR (|)
3.) NOT (~)

1.) Write a program to apply bitwise OR, AND and NOT operators on bit level.

⇒ NOT -

for NOT = -(n+1)

Teacher's Signature : _____

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      int a, b;
5.      printf("Enter two integers:");
6.      scanf("%d %d", &a, &b);
7.      printf("a & b = %d \n", a & b);
8.      Printf("a | b = %d \n", a | b);
9.      printf("~a = %d \n", ~a);
10.     printf("~b = %d \n", ~b);
11.     return 0;
12. }
```

Output :-

Enter two integers : 7", 10

a & b = 2

a | b = 15

~a = -8

~b =

Expt.No: 11

Date 13/08/2025

Page No. 9

2- Write a program to apply left shift and right shift operator.

Ques. What is Left shift and Right shift ??

Left shift [<<] : Moves all bits to the left by given positions.

Example: 00000101
00001010 - 1st shift
00010100 - 2nd shift

Right shift [>>]: Moves all bits to the right by given positions.

Example: 00000101
00000010 - 1st shift
00000001 - 2nd shift

Teacher's Signature :

```c
#include <stdio.h>
int main()
{
    int a = 7;
    int b = 10;
printf("Left shift operator of a %d\n", a>>1);
printf("Left shift operator of b %d\n", b>>2);
printf("Right shift operator of a %d\n", a<<1);
printf("Right shift operator of b %d\n", b<<2);

    return 0;
}
```

Output :-

```
Left shift operator of a 3
Left shift operator of b 2
Right shift operator of a 14
Right shift operator of b 40
```

Expt.No: - 12

Date

Page No. 23

Preprocessor and Directives in C

1.> Write a program to define some constant variable in preprocessor.

» Preprocessor : The preprocessor is a program that run before the compiler.

- It modifies the source code by handling:
  - Macros
  - Header files
  - Conditional compilation
  - file inclusion.

Teacher's Signature : _____

exp12c1.c > ...

```c
//1.    Write a program to define some constant variable in preprocessor
#include <stdio.h>
#define PI 3.14
#define MAX 100
#define COUNTRY "India"

int main() {
    printf("Value of PI = %.2f\n", PI);
    printf("Maximum Limit = %d\n", MAX);
    printf("Country = %s\n", COUNTRY);

    return 0;
}
```

Experiment No. 12 Name:

PAGE NO.: 44
DATE: / /

2- Write a program to define a function in directives.

Teacher's Signature:

exp12c2.c > ...

```c
//2.    Write a program to define a function in directives.
#include <stdio.h>
#define SQUARE(x) ((x) * (x))
#define ADD(a, b) ((a) + (b))
int main() {
    int num = 5;
    int a = 10, b = 20;

    printf("Square of %d = %d\n", num, SQUARE(num));
    printf("Addition of %d and %d = %d\n", a, b, ADD(a, b));

    return 0;
}
```

Output:

Square of 5 = 25

Addition of 10 and 20 = 30

Experiment No. 13 Name: MACROS IN C

PAGE NO.: 45
DATE: / /

1. Write a program to define multiple macro to perform arithmetic functions.

Teacher's Signature:

exp13c1.c > ...

```c

//1.    Write a program to define multiple macro to perform arithmetic functio
#include <stdio.h>
#define ADD(a, b) ((a) + (b))
#define SUB(a, b) ((a) - (b))
#define MUL(a, b) ((a) * (b))
#define DIV(a, b) ((b) != 0 ? ((a) / (b)) : 0)
#define MOD(a, b) ((a) % (b))
int main() {
    int x, y;
    printf("Enter two numbers: ");
    scanf("%d %d", &x, &y);
    printf("\n--- Arithmetic using Macros ---\n");
    printf("Addition = %d\n", ADD(x, y));
    printf("Subtraction = %d\n", SUB(x, y));
    printf("Multiplication = %d\n", MUL(x, y));
    printf("Division = %d\n", DIV(x, y));
    printf("Modulus = %d\n", MOD(x, y));
    return 0;
}

```

Experiment No. 14 Name: Static library in C

PAGE NO.: 96
DATE: / /

1.> Write a program to create a static library for performing arithmetic functions.

Teacher's Signature:

arith.h

```
int add (int, int);
int sub (int, int);
int mul (int, int);
int divide (int, int);
```

arith.c

```
int add (int a, int b)
{
    return a+b;
}
int sub (int a, int b)
{
    return a-b;
}
int mul (int a, int b)
{
    return a*b;
}
int divide (int a, int b)
{
    return a/b;
}
```

Steps to create static library

1.) Compile the source file

gcc -c arith.c

2.) Create the static library

ar rcs libarith.a arith.o

3.) Static library created

libarith.a

Experiment No. 14 Name: _____

2. Write a program to use static library in other program

FILE : main_static.c

```
#include <stdio.h>
#include "arith.h"
int main()
{
    int a=10, b=5;
    printf("Add = %d\n", add(a,b));
    printf("Sub = %d\n", sub(a,b));
    printf("Mul = %d\n", mul(a,b));
    printf("Div = %d\n", divide(a,b));
    return 0;
}
```

Compilation -

```
gcc main_static -c.    -larith    -o static-app
```

Run

```
./static-app
```

Experiment No. 15 Name: Shared Library in C

PAGE NO.: 40
DATE: / /

1. Write a program to create a shared library for performing arithmetic functions.

FILE 1 : arith_shared.h

```
int add (int , int);
int sub (int , int);
int mul (int , int);
int divide (int, int);
```

FILE 2 : arith_shared.c

```
int add (int a , int b)
{
    return a+b;
}
  int sub (int a , int b)
{
    return a-b;
}
  int mul (int a, int b)
{
  return a*b;
}
  int divide (int a, int b)
{
    return a/b;
}
```

~~Setup~~

Steps to create shared library :

Compile with position - independent code

gcc -fPIC -c arith_shared.c

2) Create shared library -

gcc -shared -o libarith.so arith_shared.o

Shared library created

libarith.so

Teacher's Signature :

Experiment No. 15 Name:

PAGE NO.: 49
DATE: / /

2. Write a program to use shared library in other program.

FILE : main_shared.c

```
#include <stdio.h>
# include "arith_shared.h"
int main()
{
   int  a = 10  , b = 5;
printf( "Add = %d\n", add(a,b));
printf("Sub = %d\n",  sub(a,b));
printf ("Mul = %d\n",  mul(a,b));
printf ("Div = %d\n",  divide(a,b));
  return 0;
}
```

Compilation

```
gcc  main_shared.c  -L   -larith   -o  shared-app
```

Run

```
export LD_LIBRARY_PATH = .
   . /shared_app
```

Output :-

```
$ gcc -c arith.c
$ ar rcs libarith.a arith.o
$ gcc main_static.c -L. -larith -o static_app
$ ./static_app
Add = 15
Sub = 5
Mul = 50
Div = 2
```

Teacher's Signature :

Experiment No. 4 Name: Variable and Scope of variable

PAGE NO.: 50
DATE: / /

1- Declare a global variable outside all functions and use it inside various functions to understand accessibity.

Teacher's Signature: _____

Output:

20
50
50

Experiment No. 4 Name: DATE: / /

2. Declare a local variable inside a function and try to access it outside the function. Compare this with accessing the global variable from within the function.

Teacher's Signature:

C exp4c2.c > ...

```c
//2.    Declare a local variable inside a function and try to access it outside the function. Compare
#include <stdio.h>
void greet();

int a=50;
int main()
{
    int a=20;
    printf("%d\n",a);
    greet ();
    printf("%d\n",a);
    return 0;
}
void greet()
{

    int x=50;
    printf("%d\n",a);
}
```

Output :

20
50
20