INDEX

S. No.	Name of the Experiment	Page No.	Date of Experiment	Date of Submission	Remarks
1)	Experiment-1		22/8/25	22/8/25	
2)	Experiment-2		23/8/25	23/8/25	
3)	Experiment-3.1		25/8/25	25/8/25	
4)	Experiment-11		25/8/25	25/8/25	
5)	Experiment-3.2		28/8/25	28/8/25	
6)	Experiment-4		30/8/25	30/8/25	
7)	Experiment-5		14/9/25	14/9/25	
8)	Experiment-6		17/9/25	17/9/25	
9)	Experiment-8		30/9/25	30/9/25	
10)	Experiment-7		6/10/25	7/10/25	
11)	Experiment-9		11/10/25	11/10/25	
12)	Experiment-10		13/10/25	13/10/25	
13)	Experiment-12		20/10/25	20/10/25	
14)	Experiment-13		22/10/25	22/10/25	
15)	Experiment-14		23/10/25	23/10/25	
16)	Experiment-15		24/10/25	24/10/25	

Expt. No. 1 Page No. 1 Date 22/8/25

PROGRAMMING IN 'C'

Experiment - 1

Installation, Environmental Setup, Start with C

(1) Write a C program to print "Hello World".

```
#include <stdio.h>
int main() {

    printf("Hello World");
    return 0;

}
```

(2) Write a C program to print the address in multiple lines (new line).

```
#include <stdio.h>
int main() {

    printf("UPES\n");
    printf("via Premnagar\n");
    printf("Dehradun, Uttarakhand\n");
    return 0;

}
```

Teacher's Signature _____

Output:

Q1: Hellooo World

Q2: UPES
via Premnagar
Dehradun, Uttarakhand

Q3: Enter your name : Jahnvi
Enter your age : 19

name = Jahnvi
age = 19

```
#include <stdio.h>

int main () {
    int Num1, Num2, Sum;

    printf ("Enter Number 1 ");
    Scanf ("%d , & Num1);

    printf ("Enter Number2 ");
    Scanf (%d , & Num2 );

    Sum = Num1 + Num2 ;

    printf (" Sum = %d ", Sum);

    return 0 ;
}
```

Teacher's Signature _____

OUTPUT:

Enter Number 1 : 2

Enter Number 2 : 3

~~Sum = 2+3=5~~

Sum = 5

Expt. No. _____ Page No. ☐ Date ☐☐☐☐☐☐

(3) Write a program that promotes the user to enter their name and age.

```
#include <Studio.h>
int main() {

    Char name[50];
    int age;
    printf ("Enter your name");
    scanf ("%49s", name);
    printf ("Enter your age");
    scanf (%d, &age);
    printf ("name = %s \n", name);
    printf ("age = %d, age);
    return 0;

}
```

(4) Write C program to add two numbers, take number from user.

Teacher's Signature _____

Expt. No. _____ Page No. ☐ Date ☐☐☐☐☐☐

```
printf ("It is an equilateral triangle"); }
else if ((l1*l2 + l2*l2 = l3*l3) || (l2*l2 +
        l3*l3 = l1) || (l1*l1 + l3*l3 = l2*l2)) {
printf ("It is a right angle triangle");
}
else if ((l1 = l2) || (l2 = = l3) || (l3 = = l1)) {
    printf ("It is a isocelen triangle");
}
else {
    printf ("It is a scalen triangle");
}
return 0;
}
```

(2) Write a programme to compute the BMI of the person and print the BMI values as per the following ranges.

BMI = weight (Kgs) / Height (Mts) * Height (Mts)

```
#include <stdio.h>
int main() {
    float height, weight;
    printf ("Enter your weight in Kgs: ");
    scanf ("%d", &weight);
    printf ("Enter your height in mts: ");
```

Teacher's Signature _____

Expt. No. _____ Page No. ☐ Date ☐☐☐☐☐☐

Experiment - 3.1

1) Write a programe to check if the triangle is valid or not. If the validity is established, do check if the triangle is ~~isosceles~~ is or else, equilateral, right angle or scalene.
Take sides of the triangle as input form user.

```
#include <stdio.h>
int main() {
    int l1, l2, l3;
    printf("Enter length 1 : ");
    Scanf("%d", &l1);
    printf("Enter length 2 : ");
    Scanf("%d", &l2);
    printf("Enter length 3 : ");
    Scanf("%d", &l3);

if ((l1+l2) > l3 && (l2+l3) > l1 && (l1+l3) > l2) {
    printf("It is a valid triangle \n"); }
else {
    printf("It is not a valid triangle \n");
}

if ((l1 == l2) && (l2 == l3)) {
```

Teacher's Signature _____

OUTPUT:

Enter length 1 : 5
Enter length 2 : 6
Enter length 3 : 8

It is a valid triangle
It is a scalar triangle

Expt. No. _____ Page No. ☐ Date ☐☐☐☐☐☐

12) WAP a C program to Convert temperature from Celsius to Fahrenheit using the formula.
$F = (C * 9/5) + 32.$

```
#include <stdio.h>
int main () {
    int C, F;
    printf ("Enter the temperature in Celsius");
    scanf ("%d", &C);
    F = ((* 9/5) + 32);
    printf ("In Fahrenheit, the temp. is %d\n", F);
    return 0;
}
```

Teacher's Signature _____

OUTPUT :

Enter the temperature in Celsius : 20
In fahrenheit, the temp. is 52

Experiment-2

Operators

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

```
#include <stdio.h>
int main() {

    int length, width, area, perimeter;
    printf ("Enter the length");
    scanf ("%d", &length);

    printf ("Enter the width");
    scanf ("%d", &width);

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

    printf ("Area = %d\n", area);
    printf ("Perimeter = %d\n", perimeter);

    return 0;
}
```

Teacher's Signature

OUTPUT :

Enter the length : 10
Enter the width : 15
Area = 50
Perimeter = 30

Expt. No. _____ Page No. ☐ Date ☐☐☐☐☐☐

Q4- Write a program using ternary operator, the user should input the length and breadth of a rectangle, one has to find out which rectangle has the highest perimeter. The minimum no. of rectangles should be three.

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

  printf("Enter length of 2nd rectangle:");
  scanf("%d", &l2);
  printf("Enter breadth of 2nd rectangle:");
  scanf("%d", &b2);

  printf("Enter length of 3rd rectangle:");
  scanf("%d", &l3);
  printf("Enter ~~long~~ breadth of 3rd rectangle:");
  scanf("%d", &b3);
```

Teacher's Signature _____

Expt. No. _____ Page No. ☐ Date ☐☐☐☐☐☐

Q3- Write a program to check if three points (x_1, y_1) (x_2, y_2) & (x_3, y_3) are collinear or not.

```
#include <stdio.h>
int main() {
    int x1, y1, x2, y2, x3, y3;
    printf("Enter coordinates of point1(x1, y1): ");
    scanf("%d %d", &x1, &y1);

    printf("Enter coordinates of point 2(x2, y2): ");
    scanf("%d %d", &x2, &y2);

    printf("Enter coordinates of point3(x3, y3): ");
    scanf("%d %d", &x3, &y3);

    int a;
    a = x1 * (y2-y3) + x2 * (y3-y1) + x3 * (y1-y2);

    if (a == 0) {
        printf("The points are collinear"); }
    else {
        printf("The points are not collinear");
    }
    return 0;
}
```

Teacher's Signature _____

OUTPUT :

Enter coordinates of points (x_1, y_1) : 1 2
Enter coordinates of point 2 (x_2, y_2) : 3 4
Enter coordinates of point 3 (x_3, y_3) : 5 6
The points are collinear

```
scanf ("%d", &height);
float bmi;
bmi = weight / height * height;
printf ("BMI Index is : %.f", bmi);
if bmi <15 {
    printf ("Category : Starvation"); }
else if bmi >= 15.1 && bmi <= 17.5 {
    printf ("Category : Underweight");
}
else if bmi >= 17.6 && bmi <= 18.5 {
    printf ("Category : Underweight");
}
else if bmi >= 18.6 && bmi <= 24.9 {
    printf ("Category : Ideal");
}
else if bmi >= 25 && bmi <= 29.9 {
    printf ("Category : Overweight");
}
else if bmi >= 30 && bmi <= 39.9 {
    printf ("Category : Obese");
}
else if bmi >= 40 {
    printf ("Catery : Morbidity obese");
}
else {
    printf ("Invalid BMI, Check input");
}
return 0;
}
```

Teacher's Signature _____

OUTPUT:

Enter your weight in Kgs : 70
Enter your height in mts : 1.70
BMI Index is : 24.221453
Category : Ideal.

OUTPUT:

1729 = 1^3 + 12^3 and 9^3 + 10^3
4104 = 2^3 + 16^3 and 9^3 + 15^3
...

```
float rate = 0.10;
int years = 10;
for(int i = 1; i <= years; i++){
population = population * (1+rate);
printf("End of year %d population will be
    = %d", i, population);
}
return 0;
}
```

(8) Ramanujan Number is the smallest no. that can be expressed as the sum of two cubes in two different ways. Write a programe to print all such number up to a resonable limit

Ex-> of Ramanujan number : 1729
12^3+1^3 & 10^3+9^3 for a no. L=20
(that is limit)

```
# include <stdio.h>
# include <math.h>
# define LIMIT 100000

int main(){
int count_sums[LIMIT] = {0};
```

Teacher's Signature _____

OUTPUT:

Enter the length : 12

Enter the width : 5

Area = 60

Perimeter = 34

Expt. No. _____ Page No. ☐ Date ☐☐☐☐☐☐

```
#include <stdio.h>
int main() {
int num = 1 ;
for (int i = 1 ; i <= 4 ; i++ ) {
for (int k=1 ; k <= 4 - i ; k++) {
        printf ("") ;
    }
for (int j = 1 ; j <= i ; j++ ) {
        printf ("%d", num ) ;
        num += 1 ;
    }
printf ("\n") ;
}
return 0;
}
```

(4) The population of a town is 1,00,000. The population has increased steadily at a rate of 10% for 10 years. Write a programme to determine the population at each year in the last decade.

```
#include <stdio.h>
int main() {
float population = 100000 ;
```

Teacher's Signature _____

OUTPUT :

```
    1
  2  3
4  5  6
7 8 9 10
```

Expt. No. _____ Page No. ☐ Date ☐☐☐☐☐☐

```
scanf("%c", &choice);
}
while (choice == 'y' || choice == 'Y');
printf("Positive numbers = %d \n", positive);
printf("Negative numbers = %d \n", negative);
printf("zero = %d \n", zero);
return 0;
}
```

2) Write a programme to print the multiplication table of the numbers entered by the user. It should be in the correct format => Num * 1 = Num.

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

3) Write a programme to generate the following set of output

```
    1
  2   3
4   5   6
```

Teacher's Signature _____

OUTPUT:

Enter a number: 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. _____ Page No. ☐ Date ☐☐☐☐☐☐

Experiment - 3.2

1) Write a program to enter numbers till the user wants. At the end, it should display the count of +ve, -ve & zeroes entered.

```
#include <stdio.h>
int main() {
    int a;
    int positive = 0, negative = 0, zero = 0;
    char choice;
    do {
        printf("Enter a no. : \n");
        scanf("%d", &a);
        if (a > 0) {
            positive ++;
        }
        else if (a < 0) {
            negative ++;
        }
        else {
            zero ++;
        }
        printf("Do you want to continue ?(Y/N));
```

Teacher's Signature _____

OUTPUT:

Enter a no.:
15
Do you want to continue ? (Y/N) : Y

Enter a no.:
-8
Do you want to continue ? (Y/N) : Y

Enter a no.:
0
Do you want to continue ? (Y/N) : Y

Enter a no.:
2
Do you want to continue ? (Y/N) : N

Positive numbers = 2
Negative numbers = 1
Zero = 1

Experiment-1

Write a programe to apply bitwise OR, AND and NOT operators in bit label.

```
#include <stdio.h>
int main() {
  int a = 5;
  int b = 7;
  printf(" bitwise a OR b is %d \n", a|b);
  printf(" bitwise a AND b is %d \n", a&b);
  printf(" bitwise a not of a is %d \n", ~a);
  return 0;
  }
```

Write a programe to apply left shift and right shift operators.

```
#include <stdio.h>
int main () {
int a=10;
printf ("Left shift operator, %d \n", a>>2);
printf( "right shift operator , %d \n", a<<2);
return 0;
  }
```

Teacher's Signature

OUTPUT :

bitwise OR is 7

bitwise AND is 5

bitwise NOT of a is -6

OUTPUT :

Left Shift operator , 2
Right Shift operator , 40

Expt. No. _____ Page No. ☐ Date ☐☐☐☐☐

```
else if (day == 6) {
printf("Sunday");
}
return 0;
}
```

OUTPUT:

Enter year
2025
Thursday

Expt. No. _____ Page No. ☐ Date ☐☐☐☐☐☐

```
for(i=1 ; i<year ; i++) {
 if(i%4 == 0 && i%100 != 0 || i%400 == 0){
total_days = total_days + 366;
}
else {
total_days = total_days + 365;
}
}

day = total_days%7 ;

if (day == 0) {
    printf ("monday"); }
else if (day == 1) {
    printf ("Tuesday");
}
 else if (day == 2) {
 printf ("Wednesday");
}
else if (day == 3) {
printf ("Thursday");
}
else if (day == 4) {
printf ("friday");
}
else if (day == 5) {
printf ("saturday");
}
```

Teacher's Signature _____

Expt. No. _____ Page No. ☐ Date ☐☐☐☐☐☐

```
int per1, per2, per3, Largest;
per1 = 2*(l1+b1);
per2 = 2*(l2+b2);
per3 = 2*(l3+b3);

largest = per1 > per2 ? ((per1 > per3) ? per1 : per3) :
          ((per2 > per3) ? per2 : per3);
printf("Largest perimeter = %d", Largest);

return 0;
}
```

5- According to the gregorian calender, 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 in that year.

```
#include <stdio.h>
int main() {
int year, total_days, day, i;

total_days = 0;
printf("Enter year");
Scanf("%d", &year);
```

Teacher's Signature _____

OUTPUT :

Enter length of 1st rectangle : 10
Enter breadth of 1st rectangle : 5
Enter length of 2nd rectangle : 8
Enter breadth of 2nd rectangle : 15
Enter length of 3rd rectangle : 12
Enter breadth of 3rd rectangle : 10
Largest perimeter = 46

OUTPUT:

Value of intVar : 10
Value of floatVar : 20.50
Value of charVar : A

Value of intPtr (address of intVar) : 0x... (address of intVar)
Value of floatPtr (address of floatVar) : 0x... (address of floatVar)
Value of charPtr (address of charVar) : 0x... (address of charVar)

Value pointed to by intPtr : 10
Value pointed to by floatPtr : 20.50
Value pointed to by charPtr : A

Expt. No. _____ Page No. ☐ Date ☐☐☐☐☐☐

Experiment -8

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

```
#include <stdio.h>
int main(){
int intVar = 10;
float floatVar = 20.5f;
Char charVar = 'A';

int *intPtr;
float* floatPtr;
Char* charPtr;

int Ptr = &intVar;
float Ptr = &floatVar;
Char Ptr = &charVar;

printf("Value of intVar : %d \n", intVar);
printf("Value of floatVar : %.2f \n", floatVar);
printf("Value of charVar : %.C \n \n", charVar);
```

Teacher's Signature _____

```
printf("Reversed string: %s\n", mystring);

return 0;
}
```

OUTPUT :

Enter a String : Hello
Original string : Hello
Reversed string : olleH

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

void REVERSE (char str[]){
  int length = strlen (str);
  int start = 0;
  int end = length - 1;
  char temp;
while ( start < end ){
  temp = str [start] ;
  str [ start ] = str [end];
  str[end ] = temp ;
  start ++;
  end -- ;
  }
}
int main(){
  char mystring[100];
printf ("Enter a string :");
fgets (my string , size of (my string ), stdin);
my string [ strcspn (my string , "\n") ] = 0;
printf ( "Original. String : %s \n", my string );

REVERSE ( my string );
```

Expt. No. _____ Page No. ☐ Date ☐☐☐☐☐☐

```
int main(){
int start_range, end_range;
printf("Enter the starting number of the range:");
Scanf("%d", &start_range);
printf("Enter the ending number of the range:");
Scanf("%d", &end_range);

if (start_range > end_range){
int temp = start_range;
 start_range = end_range;
 end_range = temp;
 }
printf("Prime numbers b/w %d & %d are:\n",
    start_range, end_range);
for(int i = start_range, ic = end_range; i++){
 if (ISPRIME(i)){
    printf("%d\n", i);
 }
}
return 0;
}
```

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

Teacher's Signature _____

OUTPUT :

```
Enter the starting number of the range : 10
Enter the ending number of the range : 50

Prime numbers between 10 & 50 are :
11
13
17
19
23
29
31
37
41
43
47
```

Expt. No. _____ Page No. ☐ Date ☐☐☐☐☐☐

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 invoke this function to generate prime numbers b/w the given ranges.

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

int ISPRIME (int num){
 if (num <= 1){
   return 0;
  }
if (num == 2){
  return 1;
}
if (num %. 2 == 0){
   return 0;
}
for(int i = 3; i < sqrt(num); i += 2){
 if (num % i == 0){
return 0;
 }
}
 return 1;
}
```

Teacher's Signature _____

```
    return 1;
  } else {
    return fibo(num-1) + fibo(num-2);
  }
}

int main() {
  int n, i;
  printf("Enter the number of terms for the fibonacci
    Sequence:");
  scanf("%d", &n);
  if (n<0) {
  printf("Please enter a non-negative number of
    terms.\n");
  } else {
    printf("fibonacci sequence up to %d terms:\n", n);
    for(i=0; i<n; i++) {
      printf("%d", fibo(i));
    }
    printf("\n");
  }
  return 0;
}
```

Teacher's Signature _____

Expt. No. _____ Page No. ☐ Date ☐☐☐☐☐☐

```
int main() {
int number1, number2;

printf("Enter the first integer:");
Scanf("%d", &number1);

printf("Enter the second integer:");
Scanf("%d", &number2);

printf("The greatest common division of %d and
%d is: %d \n", number1, number2,
GCD(number1, number2));

return 0;
}
```

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.

```
#include <stdio.h>
int FIBO(int num) {
  if (num == 0) {
    return 0;
  } else if (num == 1) {
```

Teacher's Signature _____

Expt. No. _____ Page No. ☐ Date ☐☐☐☐☐☐

```
for(i=0; i<sizeof(n_values)/sizeof(n_values[0]); i++){
for(j=0; j<sizeof(r_values)/sizeof(r_values[0]); j++){
int n = n_values[i];
int r = r_values[j];
if(r<=n){
printf("%2d %1d %7lld \n", n, r, binomial_
coefficient(n, r));
}
}
}
return 0;
}
```

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

```
#include <stdio.h>
int GCD(int num1, int num2){
if(num2==0){
return num1;
}
else{
return GCD(num2, num1 % num2);
}
}
```

Teacher's Signature _____

Expt. No. _____ Page No. ☐ Date ☐☐☐☐☐☐

```
long long int binomial_coefficient(int n, int r) {
if (r<0 || r>n) {
  return 0;
  }
return fact_non_recursive(n) / (fact_non_recursive(r) *
    fact_non_recursive(n-r));
  }
  int main(){
  int n_values[] = {5, 6, 7, 8, 9};
  int r_values[] = {2, 3, 4};
   int i, j;
printf("...factorial calculation Examples ---\n");
printf("Number | Recursive fact | Non-Recursive fact \n");
printf("-------------\n");
 for (i=0; i<5; i++){
    int num = n_values[i];
printf("%d | %14lld | %18Ld \n", num, fact_recursive(num),
              fact_non_recursive(num));
      }
printf("\n");
printf("... Binomial coefficients(n,r) Tabulation...\n");
printf("n |r| C(n,r)\n");
printf("-------\n");
```

Teacher's Signature _____

Expt. No. _____ Page No. ☐ Date ☐☐☐☐☐☐

Experiment-6

1) Develop a recursive and non-recursive function FACT(num) to find the factorial a number i.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 diff. values of n & r with suitable messages.

```
#include <stdio.h>
long long int fact_recursive(int n){
  if (n==0){
    return 1;
  } else {
      return (long long int) n* fact_recursive(n-1);
  }
}
long long int fact_non_recursive(int n){
  long long int result = 1;
  int i;
  for(i=1; i<=n; i++){
    result *=1;
  }
  return result;
}
```

Teacher's Signature _____

t. No. _____ Page No. ☐ Date ☐☐☐☐☐☐

```
            resultMatrix[i][j] += matrixA[i][k] * matrixB[k][j];
        }
    }
}

printf("\n MatrixA:\n");
for(int i=0; i<m; i++) {
for(int j=0; j<n; j++) {
printf("%d\t", matrixA[i][j]);
}
printf("\n");
}

printf("\n Matrix B:\n");
for(int i=0; i<p; i++) {
for(j=0; j<q; j++) {
printf("%d", matrix B[i][j]);
}
printf("\n");
}

printf("\n resultant matrix(AxB):\n");
for(int i=0; i<m; i++) {
for(int j=0; j<q; j++) {
printf("%d\t", resultmatrix[i][j]);
}
printf("\n");
}
return 0;
}
```

Teacher's Signature _____

No. _____ Page No. ☐ Date ☐☐☐☐☐☐

```
int matrix A[m][n];
int matrix B[p][q];
int result_matrix[m][q];

printf("\n Enter elements of matrix A (%d x %d):\n", m,n);
for( int i=0; i<m; i++) {
for (int j=0; i<n ; j++) {
Scanf ("%d", &matrix A[i][j]);
}
}

printf("\n Enter elements of matrix B(%d x %d):\n", p,q);
for(int i=0; i<p; i++){
for(int j=0 ; j<q; j++){
Scanf ("%d", &matrix B[i][j];
}
}

for(int i=0; i<m ; i++) {
for(int j=0; j<q; j++){
result matrix[i][j]=0;
}
}

for(int i=0; i<m; i++){
for(int j=0; j<q; j++){
for(int k=0; k<n ; k++ ){
```

Teacher's Signature _____

search_num, frequency);

return 0;
}

Q4- Write a program to read two matrices A(m×n) and B(p×q) and computes the product A & B. Read matrix A and matrix B in row major order respectively. Print both the input matrices & resultant matrix with suitable headings & 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.

```
#include <stdio.h>
int main() {
int m, n, p, q;

printf("Enter the number of rows & columns for
    matrix A: ");
Scanf("%d %d", &p, &q);

if (n != q) {
printf("Matrix multiplication is not possible. The
number of columns in the first matrix must be equal
to the number of rows in the second matrix.\n");
return 1;
}
```

Teacher's Signature

Expt. No. _____ Page No. ☐ Date ☐☐☐☐☐☐

```
#include <stdio.h>
int main(){
int n, i;
printf("Enter the number of elements in the
        array:");
scanf("%d", &n);

int arr[n];

printf("Enter %d integers:\n", n);
for(i=0; i<n; i++){
printf("Enter element %d:", i+1);
scanf("%d", &arr[i]);
}
int search_num;
printf("Enter the no. to find its frequency:");
scanf("%d", &search_num);

int frequency=0;
for(i=0; i<n, i++){
if(arr[i]==search_num){
    frequency++;
}
}
printf("The no. %d appears %d times in the array\n",
```

Teacher's Signature _____

```
else if (arr[i] < 0) {
negative_count++;
}
if (arr[i] % 2 == 0) {
    even_count++;
} else {
        odd_count++;
    }
}
printf("\n--- Analysis of numbers ---\n");
printf("Positive numbers: %d\n", positive_count);
printf("Negative numbers: %d\n", negative_count);
printf("Odd numbers: %d\n", odd_count);
printf("Even numbers: %d\n" even_count);

    return 0;
}
```

3) Write a programme to read a list of integers and store it in a single dimensional array. Write a C programme to find the frequency of a particular number in a list of integer.

Teacher's Signature _____

Expt. No. _____ Page No. ☐ Date ☐☐☐☐☐☐

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

```
#include <stdio.h>
int main(){
 int size, i;
int positive_count=0;
int negative_count=0;
int odd_count=0;
int even_count=0;

printf("Enter the size of the array:");
scanf("%d", &size);

int arr[size];

printf("Enter %d integers:\n", size);
for(i=0; i<size; i++){
scanf("%d", &arr[i]);
}
for(i=0; i<size; i++){
 if(arr[i]>0){
    positive_count++;
 }
```

Teacher's Signature _____

```
printf ("Enter %d integers : \n", n);
for ( int i=0 ; i<n ; i++ ) {
printf (" Element %d : ", i+1 );
Scanf (" %d", &arr[i]);
  }

int first largest = INT_MIN;
int second largest = INT_MIN;

for ( int i=0 ; i<n ; i++ ) {
if ( arr[i] > first largest ) {
    second largest = first largest;
    first largest = arr [i];
} else if ( arr[i] > second largest && arr[i]
        = first largest ) {
    second largest = arr[i];
  }
}

if ( second largest == INT_MIN ) {
  printf ("There is no distinct second largest element");
} else {
printf ("The second largest element in the array is:
          %d \n", second largest);
  }
return 0;
}
```

Teacher's Signature _____

Expt. No. _____ Page No. ☐ Date ☐☐☐☐☐☐

Experiment-5

Q1 = Write a programe to read a list of integers and store it in a single dimensional array. Write a C programming to print the second largest integer in a list of integers.

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

int main() {
int n;
printf("Enter the number of elements in the
        array:");
Scanf("%d", &n);

if (n<2) {
printf("Invaid input: The array must contain
        at least two elements to find the second
        largest.\n");
return 1;
}
int arr[n];
```

Teacher's Signature _____

Declare a static local variable inside a function. Observe how its value persists across function calls.

```
#include <stdio.h>

void counterfunction() {
    static int callcount = 0;
    callcount++;
    printf("function called %d times \n",
        callcount);
}
int main() {
    counterfunction();
    counterfunction();
    counterfunction();
    return 0;
}
```

OUTPUT :

function called 1 times
function called 2 times
function called 3 times

Expt. No. _____ Page No. ☐ Date ☐☐☐☐☐☐

3) Declare variable with different code blocks (enclosed by curly braces) and test their accessibility within & outside those blocks.

```
# include <stdio.h>

int main ( ){
int global_to_main = 10;
printf ("Inside main block : global_to_main = %d
\n", global_to_main );
{
int inner_block_var1 = 20;
printf ("inside inner block 1 : global_to_main
= %d , inner_block_var1 = %d \n",
global_to_main , inner_block_var1 );

int inner_block_var 2 = 30 ;
printf ("inside inner block 2 ; global_to_main
= %d , inner_block_var1 = %d ; inner_block_
var 2 = %d \n ", global_to_main , inner_block_var1
inner_block_var2 );
}
}
return 0;
}.
```

Teacher's Signature _____

OUTPUT :

Inside main block : global to main = 10
Inside inner block 1 : global to main = 10, inner_block_var 1 = 20
inside inner block 2 : global to main = 10, inside_block_var 1 = 20,
inner_block_var 2 = 30

```
access_global ();
global_variable = 30;
printf ("Inside main() : global_variable
  after direct modification = %d \n",
  global_variable);
return 0;
}
```

(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.

```
#include <stdio.h>

int global_var = 10;
void my_function () {
    int local_var = 20;
    printf ("inside my_function : \n");
    printf ("Accessing global_var : %d \n", global_var);
    printf ("Accessing local_var : %d \n", local_var);
}
int main() {
printf ("inside main : \n");
printf ("Accessing global var : %d \n", global_var);
my_function();
return 0;
}
```

Teacher's Signature _____

OUTPUT:

Inside main(): initial global variable = 100
Inside modify_global(): global-variable = -100
Inside main(): global-variable after modify global() call = -100
Inside access global(): global-variable = -100
Inside main(): global-variable after direct modification = 30

OUTPUT:

inside main :
Accessing global var: 10
inside my function :
Accessing global var: 10
Accessing local var: 20

Expt. No. _____ Page No. ☐ Date ☐☐☐☐☐☐

Experiment-4

17) Declare a global variables outside as function and use it inside various functions to understand its accessibility.

```
#include <Stdio.h>
int global_variable = 100;
void modify_global() {
    global_variable = 200;
printf("Inside modify_global(): global variable = %d
                                    global_variable);
}

void accessglobal(){
 printf("Inside access global(): global_variable
    = %d \n", global_variable);
}
int main() {
 printf("inside main() : Initial global variable
    = %d \n", global variable);
 modify global();
printf("inside main(): global_variable
    after modify_global() Call = %d \n",
     global_variable);
```

Teacher's Signature _____

```
int sum_pairs[LIMIT][2] = {0};
int i, j, sum;

for (i = 1; i < 47; i++) {
  sum = i * i * i + j * j * j;
  if (sum < LIMIT) {

if (count_sums[sum] == 1) {
printf("%d^3 = %d^3 + %d^3 and %d^3
    + %d^3 \n", sum, sum_pairs[sum
    [0], sum_pairs[sum][1], i, j);
  }
if (count_sums[sum] == 0) {

  sum_pairs[sum][0] = i;
  sum_pairs[sum][1] = j;
  }
  count_sums[sum]++;
   }
  }

}
return 0;
  }
```

Experiment No. _____ Name: _____

PAGE NO.:
DATE: / /

```
while ((a=fgetc(f))!=EOF)){
    printf("%c",a);
}
fclose(f);
}
```

(2)

```
#include<stdio.h>
int main(){
FILE *f;
char a[100];
f=fopen("abc.txt","r");
if(f==NULL){
    printf("Not able to open"); return 0;
}
while(fgets(a,sizeof(a),f)!=NULL){
    printf("%s",a);
}
fclose(f);
}
```

Teacher's Signature: _____

Experiment No. _____ Name: _____

PAGE NO.:
DATE: / /

Experiment 9

①
```
#include <stdio.h>

int main() {
FILE *f;
char a[100];
f = fopen("abc.txt", "w");

if (f == NULL) {
printf("Not able to open");
return 0; }

fgets(a, sizeof(a), stdin);
fputs(a, f);
fclose(f);
}
```

②
```
#include <stdio.h>
int main() {
    FILE *f;
    char a;
    f = fopen("abc.txt", "r");
    if (f == NULL) {
        printf("Not able to open);
}
```

Teacher's Signature: _____

Experiment No. _____ Name: _____

PAGE NO.:
DATE: / /

(3)
```
#include <Stdio.h>

struct book {
    Char book_id [50];
    Char title [50];
    Char author_name [50];
    Char price [10];

    int main ( ) {
    Struct book A = { "10, "C language", "Tahmi",
                      "500" };
        printbook (a);
        return 0;
    }
void printbook ( struct book b );
printf (" %s, %d, %s, %s, b.book_id,
        b.title, b.author_name, b-price );
}
```

(4)
```
#include <stdio.h>
union person { Char name [50]; char home_add[100];
Char hostel_add [100]; char city[50]; Char state [50];
Char zip [10]; };
int main() { union person a = { "Tahmi", 21>
Bhagwati kardan", "Bhidoli", "GZB", "UP", "201002" }
printf ("%s, %s %s, a.home_add, a.city,
        a.state, a.zip); }
```

Teacher's Signature: _____

Experiment No. _____ Name: _____ PAGE NO.: DATE: / /

```
printf ("Enter details of 100 employees : \n");
for (i=0 ; i<100 ; i++) {
    printf ( "\n employee %.d )n", i+1 );
    printf ( "Enter name : ");
    Scanf ("%.s", emp[i].name);
    printf ( "Enter basic pay: ");
    Scanf ("%f", & emp[i].basic_pay);
    float DA = emp[i].basic_pay * 0.52;
    emp[i].gross_salary = emp[i].basic_pay +DA;
}
printf ("\n\n -- Employee Gross Salary list --\n");
for (i=0 ; i<100 ; i++) {
    printf ("\nName : %.s", emp[i].name);
    printf ("\n Gross Salary: %.2f\n", emp[i].gross_salary);
}
return 0;
}
```

Teacher's Signature: _____

OUTPUT :

Enter details of 100 employees :-

Employee 1

Enter name : Rahul
Enter basic pay : 20000

...

(continues till employee 100)

Experiment No. Name:

PAGE NO.:
DATE: / /

```
printf("\n\n Addition Result : ");
printComplex(sum);

printf("\n Subtraction Result : ");
printComplex(diff);

return 0;
}
```

2) Write a C programme to compute the monthly pay of 100 employees using each employee's basic pay. The DA is computed as 32% of the basic pay. Gross salary (basic pay + DA). Print the employees name and gross salary.

```
#include <stdio.h>

struct Employee {
    char basic_pay;
    float basic_pay;
    float gross_salary;
};

int main() {
    struct Employee emp[100];
    int i;
```

Teacher's Signature:

OUTPUT :

first Complex number : 5.00 + 3.00i
Second Complex Number : 2.00 + 4.00i

Addition Result : 7.00 + 7.00i
Subtraction Result : 3.00 - 1.00i

Experiment No. _____ Name: _____

PAGE NO.:
DATE: / /

```
{
printf ("%.2f + %.2fi", c.real, c.imag);
}
Struct Complex subtract_Complex (Struct Complex a,
Struct Complex b) {
 Struct Complex result;
result.real = a.real - b.real;
result.img = a.img - b.imag;
return result;
}
int main() {
 Struct Complex c1, c2, sum, diff;

printf (" Enter first Complex number : \n");
 c1 = readcomplex();
printf ("\n Enter second Complex number : \n");
c2 = read Complex();
sum = add Complex (c1, c2);
diff = Subtractcomplex (c1, c2);

printf ("\n first Complex number : ");
printcomplex (c1);

printf ("\n Second Complex Number : ");
printcomplex (c2);
```

Teacher's Signature: _____

Experiment No. _____ Name: _____

PAGE NO.:
DATE: / /

Experiment-7

Q:- Write a C programme that uses functions to perform the following operations:-
(a) Reading a complex number.
(b) Writing a complex number
(c) Addition and subtraction of two complex no.

```
#include <stdio.h>
struct complex
{
    float real;
    float image;
};
struct complex read complex()
{
    Struct complex c;
    printf (" Enter real part : \n");
    scanf ("%f", &c.real);
    printf (" Enter imaginary part :");
    Scanf ("%f", &c.image);
    return c;
}

void write complex (struct complex c)
{
return 0;
}
```

Teacher's Signature: _____

```
printf("p1 = %p, %d\n", p1, *p1);
printf("p2 = %p, %f\n", p2, *p2);
printf("p3 = %p, %c\n", p3, *p3);
return 0;
}
```

Q1 Write a f" that accepts pointers as parameters. Pass variable by reference using pointers & modify their values within the function.

⇒ When we pass variable as pointers, the function can directly modify their values in memory.

```
#include <stdio.h>
void swap (int* x, int *y){
int temp = *x;
        *x = *y;
        *y = temp;
}
int main()
{
int a=5, b=10;
printf("Before swap\n");
printf("a = %d, b = %d\n", a, b);

swap(&a, &b);
printf("\n After swap:\n");
printf("a = %d, b = %d\n", a, b);
return 0;
}
```

Teacher's Signature _____

Expt. No. _____ Page No. ☐ Date ☐☐☐☐☐☐

21) Perform pointer arithmetic (increment & decrement) on pointers of different data types. Observe how the memory address change and the effect on data access.

```
#include <stdio.h>
int main(){
int a[3] = { 10,20,30};
float b[3] = {1.1,2.2,3.3};
Char c[3] = {'A','B','C'};
int* p1 = a;
float* p2 = b;
Char* p3 = c;
printf("Original addresses :\n");
printf("p1 = %p\n", p1);
printf("p2 = %p\n", p2);
printf("p3 = %p\n", p3);
p1++; p2++, p3++;
printf("After Increment :\n");
printf("p1 = %p , %d \n; p1, *p1);
printf("p2 = %p , %f\n; p2, *p2);
printf("p3 = %p , %c \n; p3, *p3);
p1--,
p2--,
p3--,
printf("\n After Decrement \n");
```

Teacher's Signature _____

OUTPUT:

Original addresses:
p1 = 0x ... (address of a[0])
p2 = 0x .... (address of b[0])
p3 = 0x ... (address of c[0])

After Increment:
p1 = 0x ... (+4 bytes from original), *p1 = 20
p2 = 0x ... (+4 bytes from original), *p2 = 2.206000
p3 = 0x ... (+1 byte from original), *p3 = B

After Decrement:
p1 = 0x ... (original address of a[0]), *p1 = 10
p2 = 0x ... (original address of b[0]), *p2 = 1.100000
p3 = 0x ... (original address of c[0]), *p3 = A

Expt. No. _____ Page No. ☐ Date ☐☐☐☐☐☐

```
printf("\n Value of int ptr (address of int var) : %p\n",
                    (void *) int ptr);
printf("Value of float ptr (address of float var) : %p\n", (void *)
    float ptr);
printf("Value of char ptr (address of char ptr) : %p\n", (void *)
        char ptr);

printf("Value pointed to by int ptr : %d\n", *int ptr);
printf("Value pointed to by float ptr : %.2f\n", *float ptr);
printf("Value pointed to by char ptr : %c\n", *char ptr);

return 0;
}
```

Teacher's Signature _____

Experiment No. _____ Name: _____ PAGE NO.: DATE: / /

```
    temp = temp → next ;
    }
new Node → next = temp → next ;
temp → next = new Node ;
}
int main ()
{
Struct Node * head = NULL , * second = NULL ;
    * third = NULL ;
head = ( struct Node * ) malloc ( size of ( struct Node ));
second = ( struct Node * ) malloc ( size of ( struct Node ));
third = ( struct Node * ) malloc ( size of ( 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 ");
insert Middle ( head , 15, 1 );
printf ( " Updated list : ");
Struct Node * temp = head ;
while ( temp != NULL ) {
    printf ( " %d → ", temp → data );
    temp = temp → next ;
}
printf ( " Null \n ");
}
```

Teacher's Signature: _____

```
    }
    printf ("NULL\n");
    return 0;
    }
```

(2)

```
#include <stdio.h>
#include <stdlib.h>
Struct Node
{
int data;
Struct Node *next;
};
Void insertMiddle (Struct Node * head,
                   int data, int position)
  {
  Struct Node * newNode
        = (Struct Node*) malloc (Size of
                  (Struct Node));
newNode -> data = data;
 Struct Node *temp = head;
 int i;
 for (i=1 ; i< position ; i++)
  {
    if (temp == NULL)
    {
    printf ("Position out of range!
    return;
  }
```

Experiment No. _____ Name: _____

PAGE NO.:
DATE: / /

Experiment-10

①
```
#include <stdio.h>
#include <stdlib.h>
    struct node {
      int a;
      struct node *b;
    };
int main() {
struct node * head = NULL ;
* second = NULL * third = NULL ;

  head = (struct Node*) Malloc (sizeof(struct Node));
  second = (struct Node*) Malloc (size of (struct Node));
  third = (struct node*) Malloc (size of (struct Node));
head → data = 10;
head → next = second;
second → data = 20;
second → data next = third ;
third → data = 30;
third → next = NULL;
printf ("Linked list : ");
struct Node * temp = head;
while (temp != NULL)
{
 printf ("%d → ", temp → data);
 temp = temp → next ;
 }
```

Teacher's Signature: _____

Experiment No. _____ Name: _____

PAGE NO.:

DATE : / /

Experiment-14

(1) 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;
}
```

Teacher's Signature: _____

Experiment No. _____ Name: _____

PAGE NO.:
DATE: / /

~~Experiment-14~~

```
return 1;
}
printf("File Content: \n");
while((ch=fgets(fp))!=EOF)
{
    putchar(ch);
}
fclose(fp);
    return 0;
}
```

Experiment No. _____ Name: _____ PAGE NO.: DATE: / /

Experiment - 13

1) ~~Write a f~~

```
#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) ((a)/(b))
int main()
{
    int x=20, y=4;
    printf("Addition = %d\n", ADD(x,y));
    printf("Subtraction = %d\n", SUB(x,y));
    printf("Division = %d\n", DIV(x,y));

    return 0;
}
```

2)

```
#include <stdio.h>
int main()
{
    FILE *fp;
    char ch;
    fp = fopen("myfile.txt", "r");
    if (fp == NULL)
    printf("file not found!");
```

Teacher's Signature: _____

Experiment No. _____ Name: _____

PAGE NO.:
DATE : / /

Experiment-12

(1)

```
# include < Stdio.h >
# define PI 3.14
# define MAX 100
# define MIN 1

int main ()

{
printf ("PI = %.2f \n", PI);
printf ("MAX = %d \n", MAX);
printf ("MIN = %d \n", MIN);

return 0;
}
```

(2)

```
# include < stdio.h >
# define SQUARE (x) ((x)*(x))
  int main()
{
  int num = 5;
  printf ("Square of %d is %d \n", num,
                SQUARE (num));
return 0;
}
```

Teacher's Signature: _____

Experiment No. _____ Name: _____

PAGE NO.:
DATE: / /

Experiment-15

1)

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

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;
}
```

Teacher's Signature: _____

Experiment No. _____ Name: _____

PAGE NO.:

DATE: / /

Q)

```
# 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;
}
```

Teacher's Signature: _____