

Output:
Hello World

Output:
ABC
PQR
XYZ

Expt. No. _____

Date _____
Page No. _____

Experiment - 1

- 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 address in multiple lines

```
#include <stdio.h>
int main () {
    printf ("ABC \n");
    printf ("PQR \n");
    printf ("XYZ");
    return 0;
}
```

Teacher's Signature

Output:

Enter your name: Sahaj

Enter age : 18

Hello, Sahaj! You are 18 years old.

Output:

Enter 1st number: 6

Enter 2nd number: 6

Sum is 12

Expt. No. _____

Date _____

Page No. _____

3) WAP that prompts the user to enter their name & age

#include <stdio.h>

int main () {

char name ;

int age ;

printf ("Enter name : ") ;

scanf ("%s", &name) ;

printf ("Enter age ") ;

scanf ("%d", &age) ;

printf ("\nHello, %s! You are %d years old.", name, age)

return 0 ;

}

(8)

20/25

13/25

4) WAP to add two numbers , take numbers from user

#include <stdio.h>

int main () {

int a, b ;

printf ("Enter 1st number:") ;

scanf ("%d", &a) ;

printf ("Enter 2nd number:") ;

scanf ("%d", &b) ;

printf ("Sum is %d", a+b) ;

return 0 ;

3

Teacher's Sign

Output:
Enter length: 10
Enter breadth: 15
Perimeter = 50
Area = 150

Output:
Enter Temp in celcius : 15
Temp (in Farhenheit) = 59.000000

Expt. No.

Date _____

Page No. _____

Experiment - 2

1) WAP to calculate area & perimeter of a rectangle based on its length & width

```
# include <stdio.h>
int main () {
    int length, breadth;
    printf ("Enter length : ");
    scanf ("%d", &length);
    printf ("Enter breadth : ");
    scanf ("%d", &breadth);
    printf ("Perimeter = %d \n", 2 * (length + breadth));
    printf ("Area = %d \n", (length * breadth));
    return 0;
}
```

2) WAP to convert temperature from Celcius to Farhenheit

```
# include <stdio.h>
int main () {
    float celcius;
    printf ("Enter Temp in celcius");
    scanf ("%f", &celcius);
    float Farhenheit;
    Farhenheit = (celcius * 9 / 5) + 32;
    printf ("Temp (in Farhenheit) = %f", Farhenheit);
    return 0;
}
```

Teacher's Signature _____

Output:
 Enter length 1 of triangle : 4
 Enter length 2 of triangle : 4
 Enter length 3 of triangle : 4
 It is valid triangle
 It is an equilateral triangle

Date _____
 Expt. No. _____
 Page No. _____

Experiment 3-1

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

```
#include <stdio.h>
int main() {
    int l1, l2, l3;
    printf("Enter l1");
    scanf("%d", &l1);
    printf("Enter l2");
    scanf("%d", &l2);
    printf("Enter l3");
    scanf("%d", &l3);

    if ((l1+l2)>l3 && (l2+l3)>l1 && (l1+l3)>l2)
    {
        printf("It is a triangle \n");
        if ((l1==l2) && (l2==l3))
            printf("It is an equilateral triangle");
        if ((l1==l2) || (l2==l3) || (l1==l3))
            printf("It is isosceles triangle");
        if ((l1 != l2) && (l2 != l3) && (l3 != l1))
            printf("It is scalene triangle");
    }
}
```

Teacher's Signature _____

```
if ((l1+l1 == l2+l3 + l3+l3) || (l2+l2 == l1+l1 + l3+l3) ||  
    (l3+l3 == l1+l1 + l2+l2)) {  
    printf ("It is right angled triangle");  
}
```

```
else {  
    printf ("It is not a triangle");  
}  
return 0;
```

- 2) WAP to compute the BMI index of the person and
print the BMI value as per the following ranges
 $BMI = \frac{\text{weight (Kg)}}{\text{height (mts)}^2}$

```
#include <stdio.h>  
int main() {  
    float weight, height;  
    printf ("Enter weight (Kg)");  
    scanf ("%f", &weight);  
    printf ("Enter height (mts)");  
    float BMI;  
    BMI = weight / height * height;  
    printf ("BMI index is : %f", BMI);  
}
```

Output:

Enter your weight (in Kg) : 70
Enter your height (in mts) : 1.79

BMI index is 21.6453240
Category : ideal

Expt. No. _____

Date _____
Page No. _____

```
if (BMI < 15) {  
    printf ("Category : starvation");  
} else if ( BMI >= 15.1 && BMI <= 17.5 ) {  
    printf ("Category : Anorexic");  
} 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 ("Category : Morbidity Obese");  
}  
else {  
    printf ("Invalid BMI, check your input");  
}  
return 0;  
}
```

Teacher's Sig

Output 3:

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

Expt. No. _____

Date _____

Page No. _____

Q) WAP 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 1st coordinate ");  
scanf ("%d %d", &x1, &y1);  
printf (" Enter 2nd coordinate ");  
scanf ("%d %d", &x2, &y2);  
printf (" Enter 3rd coordinate ");  
scanf ("%d %d", &x3, &y3);
```

int slope ;

```
if ((y2 - y1) / (x2 - x1) == (y3 - y2) / (x3 - x2)) {  
    printf (" Points are collinear ");  
}
```

else {

```
    printf (" Points are non-collinear ");  
}
```

return 0 ;

Teacher's Signature

2) According to gregorian calendar, It was Monday on date 01/01/01 . If any year is input through the keyboard write a program to find out what is the day on 1st Jan of this year.

include < stdio.h >

```
int main () {  
    int year, day_of_week ;  
    int days = 0 ;
```

```
printf ("Enter year : \n");  
scanf ("%d", &year);
```

```
int i = 1
```

```
while (i < year) {
```

```
if ((i % 4 == 0 && i % 100 != 0) || (i % 400 == 0)) {  
    days = days + 366 ; }
```

```
else {
```

```
    days = days + 365 ; }
```

```
i++ ;
```

```
}
```

```
day_of_week = days % 7 ;
```

Output:

Enter the year : 2025
It's Wednesday

Expt. No. _____

Date _____

Page No. _____

```
if ( day_of_week == 0 )  
    printf ("It's Monday");  
  
else if ( day_of_week == 1 )  
    printf ("It's Tuesday");  
  
else if ( day_of_week == 2 )  
    printf ("It's Wednesday");  
  
else if ( day_of_week == 3 )  
    printf ("It's Thursday");  
  
else if ( day_of_week == 4 )  
    printf ("It's Friday");  
  
else if ( day_of_week == 5 )  
    printf ("It's Saturday");  
  
else if ( day_of_week == 6 )  
    printf ("It's Sunday");  
  
return 0;  
}
```

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 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 breadth of 3rd rectangle : ");

scanf ("%d", &b3) ;

int peri1, peri2, peri3, largest

peri1 = 2 * (l1+b1) ;

peri2 = 2 * (l2+b2) ;

peri3 = 2 * (l3+b3) ;

Output:

Enter length of 1st rectangle : 3
Enter breadth of 1st rectangle : 3
Enter length of 2nd rectangle : 4
Enter breadth of 2nd rectangle : 4
Enter length of 3rd rectangle : 2
Enter breadth of 3rd rectangle : 2

Expt. No. _____

Date _____

Page No. _____

longest = peri1 > peri2 ? ((Peri1 > Peri3) ? peri1 = peri3) :
(Peri2 > Peri3) ? peri2 = peri3);

printf("Longest Perimeter = %d , longest);

return 0;

3

Teacher's Si

Output :-

Enter number : 5
Do you want to continue ? : y

Enter number : 0

Do you want to continue ? : y

Enter number = -7

Do you want to continue ? : n

Positive numbers : 1

Negative numbers : 1

Zeroes : 1

Expt. No. _____ Date _____

Page No. _____

Experiment 3-2

Q.

1. WAP to enter numbers till the user wants. At the end, it should display the count of positive, negative and zeroes entered.

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

```
int main () {
```

```
    int num;
```

```
    int positive = 0, negative = 0, zero = 0
```

```
    char choice = 'y';
```

```
    while (choice == 'y' || choice == 'Y') {
```

```
        printf ("Enter number");
```

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

```
        if (num > 0)
```

```
            positive++;
```

```
        else if (num < 0)
```

```
            negative++;
```

```
        else
```

```
            zero++;
```

```
    printf ("Do you want to continue ?");
```

```
    scanf ("%c", &choice);
```

```
    printf ("Positive numbers → %d", positive);
```

```
    printf ("Negative numbers → %d", negative);
```

```
    printf ("Zeroes → %d", zero);
```

```
    return 0;
```

```
y
```

Teacher's Signature _____

Output :-

Enter the number : 5

The multiplication table of 5 is :

$$5 * 1 = 5$$

$$5 * 2 = 10$$

$$5 * 3 = 15$$

$$5 * 4 = 20$$

$$5 * 5 = 25$$

$$5 * 6 = 30$$

$$5 * 7 = 35$$

$$5 * 8 = 40$$

$$5 * 9 = 45$$

$$5 * 10 = 50$$

Expt. No. _____

Date _____

Page No. _____

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

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

```
printf (" Enter the number ");
scanf (" %d ", num);
```

```
printf (" The Multiplication Table of %d is ", num);
for (int i=1, i<=10, i++) {
    printf ("%d * %d = %d ", num, i, num * i);
}
```

return 0;

}

Teacher's Signature _____

Output:

Enter number of rows : 3

1
2 3
4 5 6

Expt. No. _____ Date _____

Page No. _____

Q. WAP to generate the following output:

(a)

1

2 3

4 5 6

#include <stdio.h>

int main()

{

int n;

printf("Enter number of rows:");

scanf("%d", &n);

int num=1;

for(int i=1; i<=n; i++)

{
 for(int j=1; j<=i; j++)

 printf("%d", num);

 num++;

}

 printf("\n");

}

return 0;

}

Teacher's Signature _____

Output:

Input number of rows : 5

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

Expt. No. _____ Date _____

Page No. _____

3(b) Point :

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

#include <stdio.h>

int main()

{

int n;
printf("Input number of rows: ");

scanf("%d", &n);

for (int i=0; i<n; i++)

{

long coeff = 1;

for (int j=0; j<=i; j++)

{

printf("%d", coeff);

coeff = coeff * (i-j)/(j+1);

}

printf("\n");

}

return 0;

}

Teacher's Signature _____

Output:

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

Expt. No. _____

Date _____

Page No. _____

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. WAP to determine the population at the end of each year in the last decade.

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

```
float population = 100000;
```

```
float rate = 0.10;
```

```
int year;
```

```
printf("Year Population\n");
```

```
for (year = 1; year <= 10; year++) {
```

```
population = population + (population * rate);
```

```
printf("%d %.1f", year, population);
```

```
}
```

```
return 0;
```

```
}
```

Teacher's Signature _____

Output :

Enter the limit (L) : 20

Ramanujan numbers upto the cube of 20 are :

1729
4104
13832
20683
32832
39312
40033
46683
64232
65728

Expt. No. _____ Date _____

Page No. _____

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 a reasonable limit.
Example of Ramanujan number : 1729
 $12^3 + 1^3$ and $10^3 + 9^3$ for a number L=20 (that is limit)

```
#include < stdio.h >
int main() {
    int a, b, c, d;
    int limit;
    printf("Enter the limit (L) : ");
    scanf("%d", &limit);
    printf("Ramanujan numbers up to cube of %d are: ", limit);
    for (a=1; a<=limit; a++) {
        for (b=a+1; b<=limit; b++) {
            for (c=a+1; c<=limit; c++) {
                for (d=c+1; d<=limit; d++) {
                    if ((a*a*a + b*b*b) == (c*c*c + d*d*d)) {
                        printf("%d = %d*d^3 + %d*d^3 = %d*d^3 + %d*d^3",
                               a*a*a + b*b*b, a, b, c, d);
                    }
                }
            }
        }
    }
    return 0;
}
```

Teacher's Signature _____

Output :

Inside main () : count = 10

Inside displayCount () : count = 10

Inside modifyCount () : count changed to 15

Back in main () after modification : count = 15

Expt. No. _____

Date _____

Page No. _____

Experiment 4

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

```
#include < stdio.h >  
int count = 10;
```

```
void displayCount();  
void modifyCount();
```

```
int main () {  
    printf (" Inside main () : count = %d ", count );  
    displayCount ();  
    modifyCount ();  
    printf (" Back in main () after modification = %d ", count );  
    return 0;  
}
```

```
void displayCount () {  
    printf (" Inside displayCount () : count = %d ", count );  
}
```

```
void modifyCount () {
```

```
    count += 5;
```

```
    printf (" Inside modifyCount () : count changed to %d ", count );  
}
```

Teacher's Signature _____

Output:

Inside main():
local Main = 50 (local variable)
global Var = 100 (global variable)

Inside demoFunction():

local Demo = 200 (local variable)
global Var = 100 (global variable accessible Inside function)
Back in main(), globalVar = 150 (still accessible)

Expt. No. _____

Date _____

Page No. _____

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

```
#include <stdio.h>
int globalVar = 100;
```

```
void demoFunction();
```

```
int main() {
    int localMain = 50;
    printf("Inside main():");
    printf(" localMain = %d (local variable)", localMain);
    printf(" globalVar = %d (global variable)", globalVar);

    demoFunction();
    printf(" Back in main(), globalVar = %d (still accessible)", globalVar);
    return 0;
}
```

```
void demoFunction() {
```

```
    int localDemo = 200;
```

```
    printf(" Inside demoFunction:");
    printf(" localDemo = %d (local variable)", localDemo);
    printf(" globalVar = %d (global variable)", globalVar);
```

```
    globalVar += 50;
}
```

Teacher's Signature _____

Output
↓
Error

Expt. No. _____

Date _____

Page No. _____

3. Declare variables within different code blocks (enclosed by curly braces) and test their accessibility within & outside these blocks.

#include < stdio.h >

```
int sum();  
int sub();
```

```
int main()  
{
```

```
printf("1.d' 1.d' 1.d' 1.d' , a,b,c,d);  
printf ("1.d' \n 1.d' "); sum(), sub());  
return 0;
```

}

```
int sum()  
{
```

```
int a=6 , b=4;  
return a+b;
```

}

```
int sub()  
{
```

```
int c=5 , d=4;  
return c-d;
```

}

Teacher's Signature _____

Expt. No. _____

Date _____

Page No. _____

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

#include <stdio.h>

```
int sum() {
```

```
    static int a = 0;  
    int sum = a+5;
```

```
    a = sum;  
    printf("%d", sum);  
    return a;
```

```
}
```

```
int main()
```

```
{
```

```
    sum();
```

```
    sum();
```

```
    sum();
```

```
    sum();
```

```
    return 0;
```

```
}
```

Teacher's Signature _____

Experiment 5

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

```
#include <stdio.h>
int main() {
    int arr[100], n;
    int longest, secondLongest;
    printf("Enter the number of elements:");
    scanf("%d", &n)
```

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

```
    if (arr[0] > arr[1]) {
        longest = arr[0];
        secondLongest = arr[1];
    }
}
```

```
for (int i = 2; i < n; i++) {
    if (arr[i] > longest) {
        secondLongest = longest;
        longest = arr[i];
    }
}
```

Teacher's Signature _____

Output :-

Enter the number of elements : 6

Enter 6 integers :

12 45 23 67 45 30

The second largest number is : 45

Expt. No. _____

Date _____

Page No. _____

else if (arr[i] > secondLargest && arr[i] != largest)
 secondLargest = arr[i];

3

3

printf ("The second largest number is : ", secondLargest);
return 0;

3

Teacher's Signature _____

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

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

```
do {
```

```
    printf ("Enter the no. of elements in array");
    scanf ("%d", &n);
}
```

```
while (n <= 0);
```

```
int arr[n];
int positive = 0, negative = 0, odd = 0, even = 0;
```

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

```
}
```

```
for (int i = 0; i < n; i++) {
    if (arr[i] > 0)
        positive++;
    else if (arr[i] < 0)
        negative++;
}
```

Output :

Enter the number of element in the array : 6

Enter 6 integers :

Element 1 : 10

Element 2 : -5

Element 3 : 0

Element 4 : 7

Element 5 : -8

Element 6 : 3

Total Positive numbers = 2

Negative numbers : 2

Even numbers : 3

Odd numbers : 3

Expt. No.

Date

Page No.

```
if ( arr[i] % 2 == 0 )
```

```
    even++;
```

```
else
```

```
    odd++;
```

3

```
printf (" Total Positive numbers = %d ", positive);  
printf (" Negative numbers : %d ", negative);  
printf (" Even numbers : %d ", even);  
printf (" Odd numbers : %d ", odd);
```

```
return 0;
```

3

Output :

Enter the number of elements in array : 6

Enter 6 integers :

Element 1 : 5
Element 2 : 2
Element 3 : 7
Element 4 : 5
Element 5 : 1
Element 6 : 5

Enter the number to find its frequency : 5

The number 5 appears 5 times

Expt. No. _____

Date _____

Page No. _____

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

```
#include <stdio.h>
int main () {
    int n, search, count = 0;
```

```
do {
```

```
    printf ("Enter no. of elements in array.");
    scanf ("%d", &n);
```

```
    } while (n <= 0);
```

```
int arr [n];
```

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

```
printf ("Enter the no. to find its frequency");
scanf ("%d", &search);
```

```
for (int i = 0; i < n; i++) {
    if (arr[i] == search)
        count++;
```

```
}
```

```
printf ("The number %d appears %d times", search, count);
return 0;
```

Teacher's Signature _____

Output :-

Enter the no. of rows & columns for Matrix A(m,n) : 2 3
Enter the no. of rows & columns for Matrix B(p,q) : 3 2

Enter elements of Matrix A (2x3):

$$A[0][0] = 1$$

$$A[0][1] = 2$$

$$A[0][2] = 3$$

$$A[1][0] = 4$$

$$A[1][1] = 5$$

$$A[1][2] = 6$$

Enter elements of Matrix B (3x2):

$$B[0][0] = 7$$

$$B[0][1] = 8$$

$$B[1][0] = 9$$

$$B[1][1] = 10$$

$$B[2][0] = 11$$

$$B[2][1] = 12$$

Matrix A (2x3):

$$\begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$$

$$\begin{bmatrix} 4 & 5 & 6 \end{bmatrix}$$

Matrix B (3x2):

$$\begin{bmatrix} 7 & 8 \end{bmatrix}$$

$$\begin{bmatrix} 9 & 10 \end{bmatrix}$$

$$\begin{bmatrix} 11 & 12 \end{bmatrix}$$

Resultant Matrix (AxB) (2x2):

$$\begin{bmatrix} 58 & 64 \end{bmatrix}$$

$$\begin{bmatrix} 139 & 154 \end{bmatrix}$$

Expt. No.

Date

Page No.

1. WAP that reads two matrices A(m,n) and B(p,q) and computes the product A and 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 for multiplication. Report appropriate message in case of incompatibility.

#include <stdio.h>

int main () {

int m,n,p,q;

printf(" Enter no. of rows & columns for Matrix A(m,n) : ");
scanf ("%d %d", &m, &n);

printf(" Enter the no. of rows & columns for Matrix B(p,q) : ");
scanf ("%d %d", &p, &q);

if (n != p) {

printf(" Matrix multiplication not possible");

printf(" Reason: Number of columns in A(%d) is not equal to
the number of rows in B(%d), n, p);

return 0;

int A[m][n], B[p][q], C[m][q];

printf (" Enter elements of Matrix A (%d x %d) : ", m, n);

Teacher's Signature _____

Expt. No. _____

Date _____

Page No. _____

```
for (int i=0 ; i<m ; i++) {  
    for (int j=0 ; j<n ; j++) {  
        printf ("%d", A[i][j]);  
        if (j != n-1) printf (" ");  
    }  
}
```

```
for (int i=0 ; i<m ; i++) {  
    for (int j=0 ; j<n ; j++) {  
        for (int k=0 ; k<p ; k++) {  
            C[i][j] += A[i][k] * B[k][j];  
        }  
    }  
}
```

```
printf (" Matrix A (%d x %d) : ", m, n);  
for (int i=0 ; i<m ; i++) {  
    for (int j=0 ; j<n ; j++) {  
        printf ("%d", A[i][j]);  
        if (j != n-1) printf (" ");  
    }  
    printf ("\n");  
}
```

```
printf (" Matrix B (%d x %d) : ", p, n);  
for (int i=0 ; i<p ; i++) {  
    for (int j=0 ; j<n ; j++) {  
        printf ("%d", B[i][j]);  
        if (j != n-1) printf (" ");  
    }  
    printf ("\n");  
}
```

Teacher's Signature _____

Expt. No. _____

Date _____

Page No. _____

```
printf (" Resultant Matrix [AxB] (1d x 1d : ,m,n) :  
for (int i=0 ; i<m ; i++) {  
    for (int j=0 ; j<n ; j++) {  
        printf ("%d", C[i][j]);  
    }  
    printf ("\n");  
}  
return 0;
```

Output:

Enter the value of a : 4

Factorial = 24

Expt. No. _____

Date _____

Page No. _____

Experiment 6

1. Develop a recursive and non recursive function FACT(num) to find the factorial of a number, $n!$, defined by $\text{FACT}(n)=1$, if $n=0$. otherwise $\text{FACT}(n) = n \cdot \text{FACT}(n-1)$. Using the 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>
int fact (int a)
{
    int fact = 1;
    for (int i=1 ; i <= a ; i++)
    {
        fact = fact * i
    }
    int main()
    {
        int a;
        printf (" Enter value of a ");
        scanf ("%d", &a);
        int b = fact (a);
        printf (" Factorial = %d ", b);
```

Teacher's Signature _____

Output:

Enter the number : 10 20

$$\text{GCD} = 10$$

Expt. No. _____

Date _____

Page No. _____

- 2- Develop a recursive function GCD(num1, num2) that accepts two integers arguments. WAP that involves this function to find the greatest common divisor of given integers

```
# include <stdio.h>
int gcd (int a, int b)
{
    if (b == 0)
        return a;
    else
        return gcd (b, a % b);
}

int main()
{
    int a, b;
    printf ("Enter numbers:");
    scanf ("%d %d", &a, &b);
    int c = gcd (a, b);
    printf ("GCD = %d", c);
}
```

Teacher's Signature _____

Output :

Enter the number : 6
0 1 1 2 3 5 8

Expt. No. _____

Date _____

Page No. _____

Q. Develop a recursive fibo(num) that accepts an integer argument. WAP that involves this function to generate the fibonacci sequence upto num.

```
# include <stdio.h>
int fibo(int num)
{
    if (num == 0)
        return 0;
    if (num == 1)
        return 1;
    else
        return fibo(num-2) + fibo(num-1);
}
```

```
int main()
{
    int num;
    printf("Enter number");
    scanf("%d", &num);
    for (int i=0, i<=num; i++)
    {
        int t = fibo(i);
        printf("%d", t);
    }
}
```

Teacher's Signature _____

Output:

Enter lower limit : 10
Enter higher limit : 30

Prime numbers b/w 10 and 30 are:
11, 13, 17, 19, 23, 29

Expt. No. _____

Date _____

Page No. _____

4. Develop a C function is_prime(num) that accepts an integer argument & returns 1 if the argument is prime, 0 otherwise.

WAP that involves this function to generate prime nos b/w the given ranges.

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

```
void main()
{
```

```
    int low, high, i;
    printf("Enter lower limits: ");
    scanf("%d", &low);
    printf("Enter higher limit: ");
    scanf("%d", &high);
```

Teacher's Signature _____

```
printf ("Prime numbers b/w %d and %d are: ", low, high);
```

{

```
if (ISPRIME [i])
```

```
    printf ("%d", i);
```

}

}

5 Develop a ~~main~~ function REVERSE(str) that accepts a string argument. WAP that invokes this function to find the reverse of a given string.

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

void Reverse (char str[100])
{
    for (int i = strlen(str)-1 ; i >= 0 ; i--)
    {
        printf("%c", str[i]);
    }
}

int main()
{
    char str[100];
    printf("Enter string");
    fgets(str, sizeof(str), stdin);
    if (str[strlen(str)-1] == '\n')
    {
        str[strlen(str)-1] = '\0';
    }
    Reverse(str);
    return 0;
}
```

Teacher's Signature _____

Experiment - 7

1. WAP that uses function to perform the following operations :

- Reading a complex number
- Writing a Complex number
- Addition & subtraction of two complex numbers

```
#include <stdio.h>
struct complex {
    int rp, ip;
} c;
struct complex read()
{
    struct complex c;
    printf("Enter the real part");
    scanf("%d", &c.rp);
    printf("Enter the imaginary part");
    scanf("%d", &c.ip);
    return c;
}
void write(struct complex c)
{
    printf("%d + %d i\n", c.rp, c.ip);
}
struct complex add(struct complex a, struct complex b)
{
    struct complex res;
    res.rp = a.rp + b.rp;
```

Teacher's Signature _____

```
res.ip = a.ip + b.ip ;  
return res ;  
}
```

```
struct complex sub( struct complex a, struct complex b )
```

```
{ struct complex res ;
```

```
res.rp = a.rp - b.rp ;
```

```
res.ip = a.ip - b.ip ;
```

```
return res ;
```

```
}
```

```
int main()
```

```
{
```

```
struct complex c1, c2, c3, c4 ;
```

```
printf ("Enter first complex number") ;
```

```
c1 = read () ;
```

```
printf ("Enter second complex number") ;
```

```
c2 = read () ;
```

```
write (c1) ;
```

```
printf ("\n") ;
```

```
write (c2) ;
```

```
c3 = add(c1, c2)
```

```
c4 = sub(c1, c2)
```

```
        printf("\n");
        write(c2);
        write(c4);
        return 0;
    }
```

2. WAP to compute the monthly pay of 100 employees using each employee's name, basic pay. The DA is computed as 52% of the basic pay.

```
#include <stdio.h>
struct Employee
{
    char name[100];
    int basic;
    float gross;
};

int main()
{
    struct Employee emp[100];
    for(int i = 0; i < 100; i++)
        printf("Enter name:");
        scanf("%s", &emp[i].name);
        printf("Enter basic pay:");
        scanf("%d", &emp[i].basic);
}
```

Teacher's Signature

```

float = 0.52 * emp[i].basic;
emp[i].gross = emp[i].basic + DA
}
printf("\n")
for (int i=0; i<100; i++)
{
    printf("%s", emp[i].name);
    emp[i] = gross;
}
return 0;
}

```

3. Create a Book structure containing book_id, title, author name & price. WAP to pass a structure as a function argument & print the book details

```

struct book
{
    int id, price;
    char title[50], author[50];
}
void details( struct book a )
{
    printf("Id : %d\n", a.id);
    printf("Title : %s\n", a.title);
    printf("Author : %s\n", a.author);
    printf("Price : %d\n", a.price);
}

```

Teacher's Signature

```
int main()
```

{

```
    struct book a;  
    printf("Enter book id :");  
    scanf("%d", &a.id);
```

```
    printf("Enter book title");  
    scanf("%s", &a.title);
```

```
    printf("Enter author's name");  
    scanf("%s", &a.author);
```

```
    printf("Enter book price");  
    scanf("%d", &a.price);
```

```
    details(a);
```

```
    return 0;
```

}

Teacher's Signature

4.

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

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

union address

{

char name [50];

char home_address [100], hotel_address [100]; city [50], state [50],

zip [10];

} ;

int main()

{

union address a;

printf ("Enter your present hotel address : ");

gets (a.hotel_address, size_of(a.hotel_address, stdin));

printf ("Your present address is : ");

printf ("%s", a.hotel_address);

return 0;

}

Teacher's Signature

Output :

Values of variables :
 $a = 10$, $b = 5.50$, $c = A$

Addresses stored in pointers :
 $p1 = 0x7ffee7fb300c$, $p2 = 0x7ffee7fb300d$, $p3 = 0x7ffee7fb3008$

Values accessed using pointers :-

$*p1 = 10$, $*p2 = 5.50$, $*p3 = A$

Expt. No. _____

Date _____

Page No. _____

Experiment 8

1. Define different types of pointers (int, float, char) and initializing 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 a = 10;

float b = 5.5;

char c = 'A';

int *p1 = &a;

float *p2 = &b;

char *p3 = &c;

printf (" Values of variables : ");

printf (" a = %d, b = %.2f, c = %c ", a, b, c);

printf (" Addresses stored in pointers ");

printf (" p1 = %p, p2 = %p, p3 = %p " , (void *)p1, (void *)p2, (void *)p3);

printf ("

printf ("

return 0;

}

Teacher's Signature _____

Output :

Original addresses :
 $p_i = 0x7ffe77b39fc$, $p_f = 0x7ffe77b39f8$, $p_c = 0x7ffe77b39f7$

After incrementing pointers :
 $p_i = 0x7ffe77b3000$, $p_f = 0x7ffe77b39fc$, $p_c = 0x7ffe77b39f8$

After decrementing back :

$p_i = 0x7ffe77b39fc$, $p_f = 0x7ffe77b39f8$, $p_c = 0x7ffe77b39f7$

Expt. No. _____

Date _____

Page No. _____

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

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

```
int main() {
```

```
int x = 100;
```

```
float y = 25.75;
```

```
char z = 'Z';
```

```
int *pi = &x;
```

```
float *pf = &y;
```

```
char *pc = &z;
```

```
printf("After incrementing pointers");
```

```
printf(" pi = %p, pf = %p, pc = %p", (void*)pi, (void*)pf, (void*)pc);
```

```
pi -= 2;
```

```
pf -= 2;
```

```
pc -= 2;
```

```
printf("After decrementing back");
```

```
printf(" pi = %p, pf = %p, pc = %p", (void*)pi, (void*)pf, (void*)pc);
```

```
return 0;
```

```
}
```

Teacher's Signature _____

Before function call : num = 5 , val = 3.50

After function call : num = 15 , val = 7.00

Expt. No. _____

Date _____

Page No. _____

Q: WAP that accepts pointers as parameters . Pass variables by reference using pointers & modify their values within the function.

#include < stdio.h >

void modifyValues (int * a , float * b) {

* a += 10 ;

* b *= 2 ;

}

int main () {

int num = 5 ;

float val = 3.5 ;

printf (" Before function call : num = %d , val = %.2f , num , val) ;

modifyValues (&num , &val) ;

printf (" After function call : num = %d , val = %.2f , num , val) ;

return 0 ;

}

Teacher's Signature _____

Experiment - 9

1. WAP to create a new file & write text to it.

```
#include <stdio.h>
int main() {
    FILE *fp
    fp = fopen("file1.txt", "w")
    if (fp == NULL) {
        printf("File could not be created")
        return 1
    }
    fprintf(fp, "Hello! This is my first file in C")
    fclose(fp)
    printf("Data written successfully")
    return 0
}
```

Output : (file1.txt)

Hello! This is my first
file in C

2. Open an existing file & read its content character by character, & then close the file.

```
#include <stdio.h>
int main() {
    FILE *fp
    char ch
    fp = fopen("file1.txt", "r")
    if (fp == NULL) {
        printf("File not found")
        return 1
    }
    while ((ch = fgetc(fp)) != EOF) {
        printf("%c", ch)
    }
    fclose(fp)
    return 0
}
```

Output :

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

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

```
int main() {  
    FILE * fp;  
    char line[100];
```

```
fp = fopen("file1.txt", "r");
```

```
if (fp == NULL) {  
    printf("File can't be opened");  
    return 1;  
}
```

```
printf("File content:\n");  
while (fgets(line, sizeof(line), fp)) {  
    printf("%s", line);  
}
```

```
fclose(fp);
```

```
return 0;
```

```
}
```

Teacher's Signature _____

Experiment - 10

1. WAP to create a simple linked list in C using pointers & structures

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

struct Node {
    int data;
    struct Node *next;
};

int main() {
    struct Node *head, *second, *third;
    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;
    struct Node *temp = head;
    while (temp != NULL) {
        printf(" %d -> ", temp->data);
        temp = temp->next;
    }
    printf("NULL");
    return 0;
}
```

Experiment - 10 (Part 2)

2. WAP to insert item in middle of linked list.

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

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

```
struct node
```

```
{ int data ;
```

```
 struct node* next ;
```

```
 } ,
```

```
 int main()
```

```
{
```

```
 struct node* head = NULL , *temp , *new ;
```

```
 int arr[] = { 10, 20, 30, 40 } ;
```

```
 int n = 4 ;
```

```
 for ( int i = 0 ; i < n ; i++ )
```

```
{
```

```
 new = ( struct node* ) malloc ( sizeof ( struct node ) ) ;
```

```
 new -> data = arr [ i ] ;
```

```
 new -> next = NULL ;
```

```
 if ( head == NULL )
```

```
 { head = new ; }
```

```
 else
```

```
 { temp = head ; }
```

```
 while ( temp -> next != NULL )
```

Teacher's Signature _____

```
{ temp = temp->next;  
temp->next = new;  
}  
}
```

```
int count = 0;  
temp = head;  
while (temp != NULL)  
{ count++;  
temp = temp->next; }
```

```
int mid = count/2;  
new = (struct node*) malloc(sizeof(struct node));
```

```
new->data = 99;  
temp = head;  
for (int i=1; i<mid; i++)  
{  
temp = temp->next; }
```

```
new->next = temp->next;  
temp->next = new;  
temp = head;  
while (temp != NULL)
```

```
{  
printf("%d *->", temp->data);  
temp = temp->next; }
```

```
printf("\nNULL");  
return 0;
```

Teacher's Signature _____

tab

Q

W

E

R

P

T

Y

I

7

8

9

Output - 1

bitwise a OR b is 7
 bitwise a AND b is 5
 bitwise not of a is -6

Output - 2

left shift operator 2
 Right shift operator 40

Expt. No. _____

Date _____

Page No. _____

Experiment 11

1) WAP to apply bitwise OR, AND & NOT operations on bit level

```
#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;
}
```

2) WAP to apply left shift & right shift operator.

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

Teacher's Signature _____

Experiment-12

1. WAP to define some constant variable in preprocessor.

```
#include <stdio.h>
#define PI 3.14159
int main() {
    float radius = 5;
    float area = PI * radius * radius;
    printf("Radius : %.f", radius);
    printf("Area : %.f", area);
    return 0;
}
```

Output:

Radius : 5.00

Area : 78.54

2. WAP to define a function in directives

```
#include <stdio.h>
#define SQUARE(x) ((x)*(x))
int main() {
    int num = 6;
    printf("Number: %d", num);
    printf("Square : %d", SQUARE(num));
    return 0;
}
```

Output:

Number : 6

Square : 36

Experiment - 13

1. WAP to define multiple macros to perform arithmetic functions

```
#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= 10;
    printf("Addition : %d", ADD(x,y));
    printf(" Subtraction : %d", SUB(x,y));
    printf(" Multiplication : %d", MUL(x,y));
    printf(" Division : %d", DIV(x,y));
    return 0;
}
```

Output :

Addition : 30

Subtraction : 10

Multiplication : 200

Division : 2

Experiment - 14

1. WAP to create a static library for performing arithmetic functions

file 1 ar.h

```
int add(int a, int b);
int sub(int a, int b);
int product(int a, int b);
int div(int a, int b);
```

file 2 ar.c

```
#include "ar.h"
int add(int a, int b)
{
    return a+b;
}
int sub(int a, int b)
{
    return a-b;
}
```

```
int product(int a, int b)
{
    return a*b;
}
```

```
int div(int a, int b)
{
    return a/b;
}
```

file 3 ar main.c

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

```
int main()
```

```
int a = 20, b = 10;
printf("Sum : %d\n", add(a,b));
printf("Difference : %d, sub(a,b));
printf("Product : %d, product(a,b));
printf("Division : %d, div(a,b));
```

```
return 0;
}
```

Teacher's Signature _____

2. WAP to use static library in other program

```
#include <stdio.h>
#include "arith.h"
int main()
{
    int a=5, b=5;
    printf(" sum : %d ", add(a,b));
    printf(" difference : %d ", sub(a,b));
    printf(" Product : %d ", product(a,b));
    printf(" division : %.2f ", div(a,b));
    return 0;
}
```

Teacher's Signature

Experiment-15

1. WAP to create a shared library for performing arithmetic function

Commands to run static library:

- ① gcc -c arith.c
- ② ar rcs libarith.a arith.o
- ③ gcc main.c -L -larith -o main
- ④ ./main

Commands to run shared library:

- ① gcc -fPIC -c arith.c
- ② gcc -shared -o libarith.so arith.o
- ③ gcc main.c -L -larith -o main
- ④ ./main