



DEVYANSH
DEOPA

B.TECH
(COMPUTER
SCIENCE)

DevyanshDeopa.25479
@stu.upes.ac.in

BATCH :
20

SAP ID :-

SUBMITTED TO

590025479 MR.RAHUL
PRASAD SIR

Experiment No.	Title	Signature
1	Installation, Environment Setup and Starting with C Language	_____
2	Operators	_____
3.1	Conditional Statements	_____
3.2	Loops	_____
4	Variable and Scope of Variable	_____
5	Array	_____
6	Functions	_____
8	Pointers	_____
9	bitwise operator	_____

10 STRUCTURES

11 FILE HANDLING IN C

12 DYNAMIC MEMORY ALLOCATION

13 PREPROCESSOR

14 STATIC LIBRARY

15 SHARED LIBRARY

Outputs of All C Programs

exp1.c

```
Output:  
Hello, World!
```

exp2.1.c

```
Input:  
Length = 5, Width = 3  
  
Output:  
Area of the rectangle: 15.00  
Perimeter of the rectangle: 16.00
```

EXP2.2.C

```
Input:  
Celsius = 25  
  
Output:  
Temperature in Fahrenheit: 77.00
```

EX3.3.C

```
Input:  
(x1, y1) = (1, 1)  
(x2, y2) = (2, 2)  
(x3, y3) = (3, 3)  
  
Output:  
The points are collinear.
```

EX3.5.C

```
Input:  
Rectangle 1: 4, 5  
Rectangle 2: 3, 6  
Rectangle 3: 2, 9  
  
Output:  
The highest perimeter among the three rectangles is: 22.00
```

array1.c

```
Input:  
5  
10 20 35 40 25  
  
Output:  
Second largest number = 35
```

array2.c

```
Input:  
6  
10 -4 7 -2 0 8  
  
Output:  
Count of positive numbers: 3  
Count of negative numbers: 2  
Count of even numbers: 3  
Count of odd numbers: 3
```

array3.c

```
Input:  
6  
2 3 4 2 5 2  
Number to find = 2  
  
Output:  
The number 2 appears 3 times.
```

array4.c

```
Input:  
Matrix A (2x2)  
1 2  
3 4  
Matrix B (2x2)  
5 6  
7 8  
  
Output:  
Result:  
19 22  
43 50
```

Outputs of All C Programs (Set 2)

exp2.c

Output:
My Home Address:
123, Green Valley Apartment,
MG Road,
Dehradun, Uttarakhand,
India - 248001

exp3.c

Input:
Name = Devyansh
Age = 18

Output:
Hello Devyansh! You are 18 years old.

exp3.1.c

Input:
Sides = 3, 4, 5

Output:
The triangle is valid.
The triangle is Right Angled.

EXP3.2.C

Input:
Weight = 70 kg
Height = 1.75 m

Output:
Your BMI is: 22.86
You have Normal weight.

EXP3.4.C

Input:
Year = 2025

Output:
On 1st January 2025, it was Wednesday.

exp4.c

Input:
First number = 12
Second number = 8

Output:
The sum of 12 and 8 is 20.

func1.c

```
Input:  
n = 5, r = 2  
  
Output:  
Using Recursive Function: nCr = 10  
Using Non-Recursive Function: nCr = 10
```

func2.c

```
Input:  
Numbers = 36, 60  
  
Output:  
GCD of 36 and 60 is 12
```

func3.c

```
Input:  
Number of terms = 7  
  
Output:  
Fibonacci series: 0 1 1 2 3 5 8
```

variable3.c

Output:

10

20

variable4.c

Output:

count = 1

count = 2

count = 3

Outputs of All C Programs (Set 3)

func4.c

Input:
Start = 1, End = 10

Output:
Prime numbers between 1 and 10 are:
2 3 5 7

func5.c

Input:
String = HELLO

Output:
Reversed string: OLLEH

loops.c

Input:
Numbers: 5, -3, 0, 8, -1

Output:
Count of Positive numbers: 2
Count of Negative numbers: 2
Count of Zeroes: 1

loops2.c

Input:
Number = 5

Output:
Multiplication Table of 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

loops3.c

Output:
1
2 3
4 5 6

loops3.2.c

Output:
Pascal's Triangle (5 rows):

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

loops4.c

Output:
Year■Population

1■110000
2■121000
3■133100
4■146410
5■161051
6■177156
7■194871
8■214358
9■235793
10■259372

loops5.c

Input:
Limit = 20

Output:
Ramanujan Numbers up to limit 20 are:
 $1729 = 1^3 + 12^3 = 9^3 + 10^3$

Outputs of All C Programs (Set 4)

pointers1.c

```
Output:  
Values of variables:  
a = 10  
b = 20.50  
c = A  
  
Values using pointers:  
*ptr1 = 10  
*ptr2 = 20.50  
*ptr3 = A  
  
Addresses stored in pointers:  
ptr1 = [address]  
ptr2 = [address]  
ptr3 = [address]
```

pointers2.c

```
Output:  
Original addresses:  
ptr1 (int) = [address1]  
ptr2 (float)= [address2]  
ptr3 (char) = [address3]  
  
After increment and decrement:  
ptr1 + 1 = [address], ptr1 - 1 = [address]  
ptr2 + 1 = [address], ptr2 - 1 = [address]  
ptr3 + 1 = [address], ptr3 - 1 = [address]
```

pointers3.c

```
Input:  
x = 5, y = 8  
  
Output:  
Before modification: x = 5, y = 8  
After modification: x = 15, y = 16
```

variable1.c

```
Output:  
Count inside main: 10  
Count inside displayCount: 10  
Count inside modifyCount: 15  
Count after modifyCount: 15
```

variable2.c

```
Output:  
localNumber = 5  
number = 10  
Outside function, number = 10
```

EXPERIMENT → Installation of C on Windows

Objective → To install and setup C programming environment on a windows operating system using the GCC compiler via MinGW.

PROCEDURE →

Step 1 → Download MinGW (minimalist GNU for windows).

Step 2 → Install MinGW

① Run download mingw-get-setup.exe file

② Click install & choose a directory.

③ Install, the MinGW Installation Manager.

STEP - 3. Install GCC compiler

① In the MinGW Installation Manager, select:-

- mingw32-base
- mingw32-gcc-g++
- mingw32-gcc

② Click "Apply Changes" in the installation menu.

③ Wait for the packages to download and install.

STEP 4 : Add MinGW to System PATH.

- ① Open Control Panel > System and Security > System > Advanced system settings.
- ② Click "Environment Variables"
- ③ Under System variables, find "Path", click "Edit", then "New".
- ④ Add the MinGW bin directory (e.g. C\MinGW)
- ⑤ Click OK to save changes.

Step - 5: Verify Installation

- ① Open Command Prompt (Win + R, type cmd, press enter).
 - ② Type:
gcc --version
- Write a programme to print hello.
⇒ The programme executed successfully.

Observation :-

The GCC compiler successfully compiles the C program.

The executable (Hello.exe) runs without errors, displaying the output.

(B)

Rahul

13/8/15

Exp - 1

Write a C program to print "Hello world"

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

```
int main() {
    printf("HelloWorld!");
    return 0;
}
```

|| code executed successfully.

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

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

```
int main() {
    printf("Line 1\n");
    printf("Line 2\n");
    printf("Line 3\n");
    return 0;
}
```

Exp 3 → write a program that prompts the user to input their age and name.

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

```
int main() {
    int a;
    printf("enter your age name: ");
    scanf("%d", &a);
    return 0;
}
```

Exp 4 → C program to add two numbers.

2.1
Exp 4 → write a C program to calculate the area and perimeter of a rectangle based on its length & width.

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

```
int main() {
    int a, b, sum;
    printf("enter first num \n");
    scanf("%d", &a);
    printf("enter Second num \n");
    scanf("%d", &b);
    sum = a + b;
    printf("your sum is %d", sum);
```

→ #include <stdio.h>

int main() {

float l, w;

printf("Enter length & width: ");

Scan

scanf("%f %f", &l, &w);

printf("Area: %.2f \nPerimeter: %.2f", l * w, 2 * (l + w));
return 0;

}

Ex-2

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

#include <stdio.h>

int main() {

float c, f;

// input temp in celcius

printf("Enter temperature in celcius: ");

scanf("%f", &c);

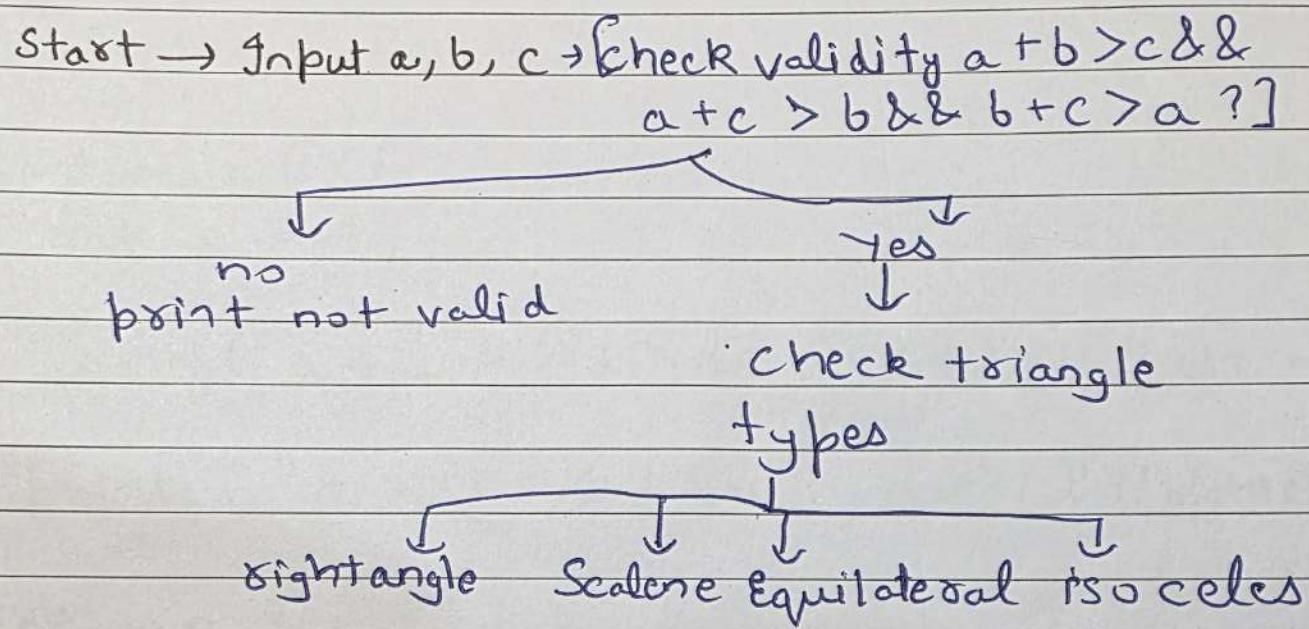
f = (c * 9/5) + 32;

printf("Temp in F: %.2f \n", f);

return 0;

}

Flow charts for 1st & 2nd questions.



2nd question.

Start → Input weight, height.

↓
Calculate BMI

↓

Check BMI

↓

give them category
using if, else if &
else.

↓

Output

② Write a program to apply left shift & right shift operators.

Ans #include <stdio.h>

```
int main () {  
    int a = 4;  
  
    printf ("a<<1 = %d\n", a<<1); //leftshift  
    printf ("a>>1 = %d\n", a>>1); //rightshift  
    return 0;
```

y

output

a << 1 = 8

a >> 1 = 2.

Output

$$x_1 = 1, y_1 = 2$$

$$x_2 = 2, y_2 = 4$$

$$x_3 = 3, y_3 = 6$$

The points are collinear.

Qn 11: Bitwise Operators

Q. Write a program to apply OR, AND & NOT operators on bit level.

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

```
int main()
```

```
    int a = 12;
```

```
    int b = 10;
```

```
    printf("a & b = %d\n", a & b);
```

```
    printf("a | b = %d\n", a | b);
```

```
    printf("~a = %d\n", ~a);
```

```
return 0;
```

Output

$$a \& b = 8$$

$$a | b = 14$$

$$\sim a = -13$$

(3)

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

Ans

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

```
int main () {
```

```
float x1, y1, y2, x2, x3, y3;
```

```
printf ("Enter x. & y.: ");
```

```
scanf ("%f %f", &x1, &y1);
```

```
printf ("Enter x. & y.: ");
```

```
scanf ("%f %f", &x2, &y2);
```

```
printf ("Enter x. & y.: ");
```

```
scanf ("%f %f", &x3, &y3);
```

|| check for collinearity using slope method.

```
if ((y2 - y1) * (x3 - x2) == (y3 - y2) * (x2 - x1))
```

{

```
    printf ("The points are collinear ");
```

}

```
else {
```

```
    printf ("points are not collinear ");
```

}

```
return 0;
```

}

else if (isRight)

else if ((a * a) == (b * b) + (c * c) || (b * b) == (a * a) + (c * c))
|| (c * c) == (a * a) + (b * b))

{
 point} ("Triangle is right angled");

else {
 point} ("Triangle is scalene");

}

 return 0;
}

OUTPUT

a = 3
b = 4
c = 5

Triangle is right angled.

Exp - 3.1

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

Ans

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

```
int main () {
```

```
float a, b, c ;
```

```
printf ("Enter side a: ");
```

```
scanf ("%f", &a);
```

```
printf ("Enter side b: ");
```

```
scanf ("%f", &b);
```

```
printf ("Enter side c: ");
```

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

```
if (a == b && b == c) {
```

```
    printf ("It is an equilateral triangle.\n");
```

```
}
```

```
else if (a == b || b == c || a == c) {
```

```
    printf ("It is an isosceles triangle.\n");
```

```
}
```

Q 2.) WAP to compute the BMI index of the person and print the BMI values as per the following ranges.

Ans → # include <stdio.h>

```

int main() {
    float w, h, b;
    printf("Enter height in meters: ");
    scanf("%f", &h);
    printf("Enter weight in Kg: ");
    scanf("%f", &w);
    b = w / (h * h);
    printf("Your bmi is: %.2f\n", b);
    if (bmi < 15) {
        printf("category: starvation\n");
    } 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 >= 30 && bmi <= 39.9) {  
    printf("category : obese "); }  
  
else if( bmi >= 40.0 ) {  
    printf("category : Morbidly obese "); }  
  
else {  
    printf("category : unspecified "); }  
  
return 0;  
}
```

Output.

Enter weight in kg : 70

Enter height in meters : 1.75

your BMI is : 22.86

category : Ideal

Q) According to the gregorian calendar, it was monday on the date 01/01/01 If any year is input through the WAP find out day on that year.

Ans

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

```
int main () {
```

```
    int year, i, day = 1; // Monday = 1
```

```
    printf ("Enter year: ");
```

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

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

```
        if ((i % 400 == 0) || (i % 4 == 0 && i % 100 != 0))
```

```
            day += 2; // leap year
```

```
        else {
```

```
            day += 1; // normal year
```

```
}
```

```
    day = day % 7;
```

```
    switch (day) {
```

```
        Case 0: printf ("Day on 1st jan %d: Sunday\n", year);
```

```
        Case I: printf ("Day on 1st jan %d: Monday\n", year);
```

```
        break;
```

```
        Case II: printf ("Day on 1st jan %d: Tuesday\n", year);
```

```
        break;
```

```
        Case 3: printf ("Day on 1st jan %d: Wednesday\n", year);
```

```
        break;
```

```
        Case 4: printf ("Day on 1st jan %d: Thursday\n", year);
```

```
        break;
```

Date _____

```
Case 5; printf("Day on 1st jan Y.d : Thursday\n",year)
break;
Case 6; printf("Day on 1st jan Y.d : saturday\n",year);
break;
}
return 0;
}
```

Output

Enter year: 2014

Day on 1st jan Tuesday.

Exap 3.2 Loops.

(1)

Ans → # include < stdio.h >

```
int main() {  
    int num;  
    int positive = 0, neg = 0, zero = 0;  
    printf("Enter a num: ");  
    scanf("%d", &num);  
    char ch = 'y';  
    while (ch == "y" || ch == 'y') {  
        if (num > 0)  
            positive++;  
        else if (num < 0)  
            negative++;  
        else  
            zero++;  
    }  
}
```

Topic _____

```
printf("Do you want to enter another  
number? (y/n): ");
```

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

```
printf("\n count of post num: %d",  
      positive);
```

```
printf("\n count of neg num: %d",  
      neg);
```

```
printf("\n count of zeroes: %d\n", zero);
```

```
return 0;
```

```
}
```

Expt 3.2 : Ques -②

Ans → `#include <stdio.h>`

```
int main() {  
    int num, i;  
    printf("Enter a num");  
    scanf("%d", &num);  
    printf("Multiplication table %d:\n", num);  
    for(i = 1; i <= 10; i++) {  
        printf("%d * %d = %d\n", num, i,  
               num * i);  
    }  
    return 0;  
}
```

Topic _____

Q

En 3.2

Q 3 (a)

Ans #include <stdio.h>

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

Topic _____

(b)

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

```
int main() {
```

```
    int i, j, num;
```

```
    for (i = 0; i < 5; i++) {
```

// print leading spaces to centre the triangle

```
        for (j = 0; j < 5 - i; j++) {
```

```
            printf(" " );
```

```
}
```

```
        num = 1;
```

```
        for (j = 0; j <= i; j++) {
```

```
            if (num < 10)
```

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

```
            else
```

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

```
            num = num * (i - j) / (j + 1);
```

```
}
```

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

```
}
```

```
return 0;
```

```
}
```

Teacher's Signature _____

Topic _____

Qub → 3.2

Q 4

Ans → #include <stdio.h>

int main () {

int year ;

int population = 100000 ;

printf ("year \t population \n ");

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

{

 population = population + (population * ~~10/00~~);

printf ("%d \t %d \n ", year, population);

}

}

Ques ⑤ Eu \rightarrow 3:2 loops

Q5 \rightarrow Ramanujan number

Ans \rightarrow # include < stdio.h >

```
int main () {  
    int a, b, c, d;  
    int L;  
    printf (" Enter the Limit : ");  
    scanf ("%d", &L);  
    printf (" In Ramanujan num Limit %d ", L);  
  
    for (a = 1; a <= L; a++) {  
        for (b = a; b <= L; b++) {  
            int sum1 = a * a * a + b * b * b;  
  
            for (c = a + 1; c <= L; c++) {  
                for (d = c; d <= L; d++) {  
                    int sum2 = c * c * c + d * d * d;  
  
                    if (sum1 == sum2) {  
                        printf (" %d = %d^3 + %d^3 = %d^3 + %d^3  
                                sum1, a, b, c, d );  
                    }  
                }  
            }  
        }  
    }  
}
```

Topic _____

Exp. PG. # _____

Date ____ / ____ / ____

return 0;
}

Output = Enter limit : 20

$$1729 = 1^3 + 12^3 = 9^3 + 10^3$$

$$4104 = 2^3 + 16^3 = 9^3 + 15^3$$

Expt - 4 : Variable & Scope of Variable

Q ① Global Variable .

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

```
int count = 10;
```

```
void displayCount() {
```

```
    printf("count inside funct: %d\n", count);
```

```
void modifiedCount() {
```

```
    count += 5;
```

```
    printf("count inside modifiedCount: %d\n", count);
```

```
}
```

```
int main() {
```

```
    printf("count inside main: %d\n", count);
```

```
    displayCount();
```

```
    modifiedCount();
```

```
    printf("count after modifiedCount: %d\n", count);
```

```
    return 0;
```

```
}
```

Exap - 4

Q(2) Local function.

Ans → #include <stdio.h>

int main() {

 int num = 10;

 void myfunction() {

 int locnum = 5;

 printf(" local num = %d \n", locnum);

 printf(" number = %d \n", number);

}

 int main() {

 myfunction();

 printf(" outside function) number = %d \n",

 number);

 return 0;

}

Output :

local number = 5

number = 10

outside number = 10.

Exp - 4 Variable .

Q ③ Blocks.

Ans → #include <stdio.h>

int main()

int z = 5;

{

int a = 10;

printf("y.d\n", a);

printf("y.d", z);

}

{

int b = 15;

printf("\ny.d", b);

}

return 0;

}

Topic _____

En - 4 Variable

Q 9 static local variable

#include <stdio.h>

```
void count() {  
    static int count = 0;  
    count++;  
    printf("count = %d\n", count);  
}  
int main() {  
    count();  
    count();  
    count();  
    return 0;  
}
```

Output :

1

2

3

Experiment - 5? Array -

Q1 Second largest .

Ans → # include <stdio.h>

int main() {

 int n, i, j;

 int arr[100];

 int first, second;

 printf("Enter how many numbers: ");

 scanf("%d", &n);

 printf("Enter %d numbers\n", n);

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

 scanf("%d", &arr[i]);

}

 first = second = arr[0];

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

 if (arr[i] > first) {

 second = first;

 first = arr[i];

}

 else if (arr[i] > second && arr[i] < first)

{

Topic _____

Second = arr[i]);

}

}

printf ("second largest number = %d\n",

second);

return 0;

}

Output

2 Enter how many numbers: 5

Enter 5 numbers

10 25 8 30 15

Second largest = 25.

Experiment - 5 arrays

Q2 pos, neg, odd, even.

Ans # include <stdio.h>

```
int main () {
```

```
    int n, i;
```

```
    int arr[100];
```

```
    int pos = 0, neg = 0, even = 0, odd = 0;
```

```
    printf("Enter how many numbers: ");
```

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

```
        scanf("%d", &arr[i]);
```

```
}
```

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

```
        if (arr[i] > 0)
```

```
            pos++;
```

```
        else if (arr[i] < 0)
```

```
            neg++;
```

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

```
            even++;
```

```
        else
```

```
            odd++;
```

```
}
```

Topic _____

Date ___/___/___

```
pointf(" count of positive %d ", poss);  
printf(" \n count of negative %d ", neg);  
printf(" \n count of odd %d ", odd);  
printf(" \n count of even %d ", even);  
  
return 0;
```

y

Experiment -5 arrays

Q 3

frequency

Ans # include < stdio.h >

int main() {

int n, i, num, count = 0;
int arr[100];

printf("Enter how many numbers: ");

scanf("%d", &n);

printf("Enter %d integers: ", n);

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

scanf("%d", &arr[i]);

}

printf("Enter the num to find its frequency ");

scanf("%d", &num);

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

if (arr[i] == num)

count++;

printf("the num %d appears %d times ", num,
count);

return 0;

}

Ex-5 array.

Q 4 Multiplication

Ans → #include <stdio.h>

int main () {

int a[10][10], b[10][10], c[10][10];

int m, n, p, q, i, j, k;

printf ("Enter rows & column of A: ");

scanf ("%d %d", &m, &n);

printf ("Enter rows & column of B: ");

scanf ("%d %d", &p, &q);

if (n != p) {

printf ("Matrix multiplication not possible");

return 0;

}

printf ("Enter elements of A : ");

for (i=0; i<m; i++)

for (j=0; j<n; j++)

scanf ("%d", &a[i][j]);

printf ("Enter elements of B : ");

for (i=0; i<p; i++)

for (j=0; j<q; j++)

scanf ("%d", &b[i][j]);

Topic _____

```
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("Result:\n");
```

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

```
        printf("%d\t", c[i][j]);  
    }  
    printf("\n");
```

```
return 0;
```

```
}
```

Exp - 6: Functions

Q1 Factorial recursive & non recursive

Ans →

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

```
int factx(int n) {
    if(n == 0)
        return 1;
} else {
    return n * factx(n-1);
}
```

```
int factn(int n) {
```

```
    int i, f = 1;
    for(i = 1; i <= n; i++)
        f = f * i;
    return f;
}
```

```
int main() {
```

```
    int n, x, ans1, ans2;
    printf("Enter n & x: ");
    scanf("%d%d", &n, &x);
```

```
    if(x > n) {
```

```
        printf("invalid input! x cannot be greater than  
n.\n");
```

```
        return 0;
    }
```

Topic

```
ans1 = fact(x(n)) / (fact(x)*fact(x(i-x)));
```

```
ans2 = factn(n) / (factn(x) * factn(n-x));
```

printf("In using Recursive function : nCr = %d", ans);

pointf("In Using Non-Recursive function: $nCr = \frac{n!}{r!(n-r)!}$, ans");

return 0;

۲

Output

Enter n and s: 6 2

using recursive function : $nC_8 = 15$.

Using Non-recursive function: $nC_8 = 15$.

Exp - 6 Functions .

Q ② GCD of two numbers.

#include <stdio.h>

```
int GCD(int a, int b) {
    if (b == 0)
        return a;
    else
        return GCD(b, a % b);
}
```

int main() {

int num1, num2, result;

```
printf("Enter two numbers: ");
scanf("%d%d", &num1, &num2);
```

```
result = GCD(num1, num2);
```

```
printf("GCD of %d and %d is %d\n", num1, num2,
      result);
```

```
return 0;
```

}

Output :-

Enter two numbers: 20 12
GCD of 20 and 12 is 4

Topic _____

Chap - 6 Function

Q3 Fibonacci series

⇒ #include <stdio.h>

```
int FIBO(int n) {
    if (n == 0)
        return 0;
    else if (n == 1)
        return 1;
    else
        return FIBO(n-1) + FIBO(n-2);
```

```
int main() {
    int num, i;
    printf("Enter how many terms: ");
    scanf("%d", &num);
    printf("Fibonacci series: ");
    for (i = 0; i < num; i++) {
        printf("%d ", FIBO(i));
    }
    printf("\n");
    return 0;
}
```

Output :-

Enter how many terms: 7.

Fibonacci Series: 0 1 1 2 3 5 8.

Exp - 6 Functions

Q4 → ISPRIME () .

Ans → #include <stdio.h>

```
int i;
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 start & end range: ");
```

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

```
    printf("Prime numbers between %d & %d are:  
          start, end);
```

```
    for (i = start; i <= end; i++) {
        if (ISPRIME(i))
            printf("%d", i));
```

Topic _____

```
y  
baintf("In");
```

```
return 0;
```

```
3
```

Output : -

Enter start & end range : 10 30

Prime numbers between 10 & 30 .

11 13 17 19 23 29.

Functions.

Q5 Reverse of a string.

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

// Function to reverse a string
void REVERSE(char str[]) {
    int i, len;
    char temp;
    len = strlen(str);
    for (i = 0; i < len/2; i++) {
        temp = str[i];
        str[i] = str[len - i - 1];
        str[len - i - 1] = temp;
    }
}

int main() {
    char str[100];
    printf("Enter a string: ");
    gets(str);
    REVERSE(str);
    printf("Reversed string: %s\n", str);
    return 0;
}
```

Ex- Pointers

Q → Addressing

Ans → #include <stdio.h>

```
int main() {
    int a = 10;
    float b = 20.5;
    char c = 'A';

    int *ptr1 = &a;
    float *ptr2 = &b;
    char *ptr3 = &c;

    printf("a = %d\n", a);
    printf("b = %f\n", b);
    printf("c = %c\n", c);

    printf("*ptr1 = %d\n", *ptr1);
    printf("*ptr2 = %f\n", *ptr2);
    printf("*ptr3 = %c\n", *ptr3);

    printf("ptr1 = %p\n", ptr1);
    printf("ptr2 = %p\n", ptr2);
    printf("ptr3 = %p\n", ptr3);

    return 0;
}
```

@ Em - Pointers

Increment & Decrement .

#include <stdio.h>

```
int main() {  
    int a = 10;  
    float b = 20.5;  
    char c = 'A';
```

```
    int *ptr1 = &a;  
    float *ptr2 = &b;  
    char *ptr3 = &c;
```

```
printf("Original address - ");  
printf("ptr1 = %.p", ptr1);  
printf("ptr2 = %.p", ptr2);  
printf("ptr3 = %.p", ptr3);
```

// After arithmetic.

```
printf("ptr1 + 1 = %.p, ptr1 - 1 = %.p\n", ptr1 + 1, ptr1 - 1);  
printf("ptr2 + 1 = %.p, ptr2 - 1 = %.p\n", ptr2 + 1, ptr2 - 1);  
printf("ptr3 + 1 = %.p, ptr3 - 1 = %.p\n", ptr3 + 1, ptr3 - 1);
```

return 0;

}

Topic _____

Ex- Pointers

Q accept pointer & modify the value.

Ans → #include <stdio.h>

```
void modifyValue(int *a, int *b) {
    *a = *a + 10;
    *b = *b * 2;
}
```

```
int main() {
    int n = 5, y = 8;
```

printf("Before modification : n=%d, y=%d\n", n, y);

modifyValue(&n, &y);

printf("After modification : n=%d, y=%d\n", n, y);

return 0;

}

Lab - 9 File Handling in C.

Q ① Create a new file.

→ #include <stdio.h>

```
int main() {
    FILE * fp = fopen("myfile.txt", "w");
    if (fp == NULL) {
        printf("Error creating file!");
        return 1;
    }
```

```
    fprintf(fp, "Hello, this is a test file.\n");
    fclose(fp);
```

```
    printf("File created and text written.\n");
    return 0;
}
```

Q ②

Ans → #include <stdio.h>

```
int main() {
    FILE * fp = fopen("myfile.txt", "r");
    char ch;
    if (fp == NULL) {
        printf("File not found!");
        return 1;
    }
```

```
    while ((ch = fgetc(fp)) != EOF)
        putchar(ch);
    fclose(fp);
    return 0;
}
```



Q3 →

Ans → #include <stdio.h>

```
int main() {
    FILE *fp = fopen("myfile.txt", "r");
    char line[200];
    if (fp == NULL) {
        printf("File not found!");
        return 1;
    }
    while (fgets(line, sizeof(line), fp)) {
        printf("%s", line);
    }
    fclose(fp);
    return 0;
}
```

Q4 → MACROS.

Ans → #include <stdio.h>

#define ADD(a,b) ((a)+(b))

#define SUB(a,b) ((a)-(b))

int main()

int n = 10, y = 5;

printf("Add = %d\n", ADD(n,y));

printf("SUB = %d\n", SUB(n,y));

return 0;

}

A

Teacher's Signature _____

Experiment - 10 - Dynamic memory allocation

① Write - - - - - & structure.

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

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

int main() {
    struct Node* head = NULL;
    struct Node* second = NULL;
    struct Node* 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;

struct Node* temp = head;
printf("Linked List: ");
while (temp != NULL) {
    printf("%d ", temp->data);
    temp = temp->next}
```

}

```
return 0;
}
```

Q → 2

Ans →

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

```
struct N{ int d; struct N* n; };
```

```
int main () {
```

```
    struct N *a = malloc (sizeof (struct N));
    * b = malloc (sizeof (struct N));
    * c = malloc (sizeof (struct N));
    * m;
```

```
a -> d = 10; a -> n = b;
```

```
b -> d = 10; b -> n = c;
```

```
c -> d = 30; c -> n = NULL;
```

```
m = malloc (sizeof (struct N));
```

```
m -> d = 15;
```

```
m -> n = b;
```

```
a -> n = m;
```

```
for (struct N *t = a; t != t -> n; print ("x d =>",
      t -> d));

```

```
return 0;
```

}

Structure in C.

Q① Create a union containing 6 strings : name , home_address.

Ans → #include <stdio.h>
#include <string.h>

struct Address {

char name [50];

char home_address [100];

char hostel_address [100];

char city [50];

char state [50];

char zip [10];

}

int main()

struct Address a = {

" Devyansh Deshpande ",

" 123, Green Valley, Dehradun ",

" Hostel Block A, Room 12 ",

" Dehradun ",

" Uttarakhand ",

" 248001 "

}

printf (" Present Address : \n %.s , %.s - %.s \n ",

a.home_address, a.city, a.state, a.zip);

return 0;

}

Q ③

Create a Book structure.

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

```
struct Book {  
    int book_id;  
    char title [50];  
    char author [50];  
    float price;  
};
```

```
void printBook(struct Book b){  
    printf("Book Details: \n");  
    printf("ID : %d\n", b.book_id);  
    printf("Title : %.50s", b.title);  
    printf("Author : %.50s", b.author);  
    printf("Price %.2f\n", b.price);  
}
```

```
int main(){
```

```
    struct Book bl = {101, "C Programming",  
        Dennis Ritchie, 299.50};
```

```
    printBook(bl);
```

```
    return 0;
```

```
}
```

Q2 →

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

```
struct Employee {  
    char name [50],  
    float basic;  
};
```

```

int main() {
    struct Employee emp[100];
    int i;
    float da, gross;
    for (i = 0; i < 100; i++) {
        printf("Enter name & basic pay of emp%d: ", i + 1);
        scanf("%s %f", emp[i].name, &emp[i].basic);
    }
    printf("\n Employee Name Gross Salary \n");
    printf("-----\n");
    for (i = 0; i < 100; i++) {
        da = 0.52 * emp[i].basic;
        gross = emp[i].basic + da;
        printf("%s %.2f\n", emp[i].name, gross);
    }
    return 0;
}

```

Experiment - 14 : static library in C.

① Static Library for arithmetic functions.
 Create header file - arith.h

```

int add(int a, int b);
int sub(int a, int b);
int mul(int a, int b);
int divi(int a, int b);

```

Create C File - 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 divi (int a, int b) { return a / b; }
```

gcc -c arith.c .

arith.o

arx rcs libarith.a arith.o.

libarith.a

Q② Use it in other program .

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

```
int main () {
    printf ("Add = %d\n", add (10, 5));
    printf ("Sub = %d\n", sub (10, 5));
    printf ("Mul = %d\n", mul (10, 5));
    printf ("Div = %d\n", divi (10, 5));
    return 0;
}
```

gcc main.c -L. -larith -o output .

.output

Q1 - Shared Library.

Q1)

Ans → The code can be same as static library.

during compile.

gcc -fPIC -c arith.c

arith.o

gcc -shared -o libarith.so arith.o

libarith.so

Q2) Using it in different program.

Ans → Code same as static

gcc main.c -L. -larith -o output

export LD_LIBRARY_PATH=.

• Output.

Expt 12 → Preprocessors and Directives in C.

a①

define some constant variable in pre processor.

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

```
#define PI 3.14159
```

```
#define COURSE "computer science"
```

```
int main () {
```

```
    printf ("PI = %.5f\n", PI);
```

```
    printf ("Course = %s\n", COURSE);
```

```
    return 0;
```

```
}
```

a②

Preprocessor Directive.

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

```
#define square (n) ((n) * (n))
```

```
#define MAX (a,b) ((a) > (b) ? (a) : (b))
```

```
int main () {
```

```
    int a = 5, b = 9;
```

```
    printf ("Square of %d = %d\n", a, square(a));
```

```
    printf ("Max of %d = %d\n", a, b, MAX(a, b));
```

```
    return 0;
```

```
}
```