B.TECH (COMPUTER SCIENCE)

EKTA.22698@stu.upes.ac.in

EKTA BHARDWAJ

BATCH :- 20

SAP ID :-

590022698

SUBMITTED TO

MR.RAHUL PRASAD SIR

INDEX

NAME Ukta Bhardwaj SUB. Programming in C

STD B. Tech (CSE) DIV. _____ ROLL NO. 590022698

S. No.	Date	Title	Pages No.	Remarks
1.	06-08-25	Prog to print "Hello world"	1-2	
2.	06-08-25	Prog. to print address in multiple lines	3-4	
3	06-08-25	Prog. to enter their name and age	5-6	
4	06-08-25	Prog. to add two numbers	7-8	
5	06-08-25	Prog. to calculate area & perimeter of rectangle	9-10	
6	06-08-25	Prog. to convert temp °C to °F	11-12	
7.	16-08-25	Prog to check and validate triangle type	13-15	
8.	16-08-25	Prog to compute BMI	16-18	
9	16-08-25	Prog. to check collinearity	19-20	
10	16-08-25	Prog to find out day	21-23	
11.	16-08-25	Prog to calculate highest perimeter	24-25	
12	16-08-25	Prog to apply bitwise operator	26-27	
13.	16-08-25	Prog. to apply left & right shift	28-29	
14	31-08-25	Prog to display the count of positive, negative & zero	30-31	
15	31-08-25	Prog. to print table	32-33	
16	31-08-25	Prog. to print pattern	34-36	
17	14-09-25	Determine population	37-38	

INDEX

NAME Ekta Bhardwaj SUB Programming in C

STD B. Tech (CSE) DIV ROLL NO 5900 22698

S. No.	Date	Title	Pages No.	Remarks
18	14-9-25	Declare a global variable	39-40	
19	14-9-25	Declare a local variable	41-42	
20	14-9-25	Declare within blocks	43	
21	14-9-25	Declare a static variable	44	
22	14-9-25	Develop recursive f^n for factorial	45-46	
23	14-9-25	Recursive f^n for GCD	47-48	
24	14-9-25	Recursive function for fibo series	49-50	
25	14-9-25	Recursive function for prime no	51-52	
26	30-9-25	Recursive function for string	53-54	
27	30-9-25	Second largest no in array	55-56	
28	30-9-25	Ramanujan no	57-58	
29	30-9-25	Count and display +ve, -ve no	60-62	
30	30-9-25	frequency of a number	63-64	
31	30-9-25	Product of matrix	65-68	
32	30-9-25	Print the value of pointer	69-70	
33	30-9-25	Pointer arithmetic	71-72	
34	30-9-25	Pass by reference in pts	73-74	
35	11-11-25	Read, write, add, sub of complex no	75-77	
36	11-11-25	Employee name & gross salary	78-79	
37	11-11-25	Print book details	80-81	
38	11-11-25	Create, open, read file	82-87	
39	12-11-25	Linked List	88-92	
40	13-11-25	Macros	93-94	
41	14-11-25	Static & Shared library	95-107	

Experiment No. 1

1.

Aim

Write a C program to print "Hello world".

Software used

Language : C
Compiler : GNU compiler collection
Editor : Programiz

Code

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

Teacher's Signature

Topic _____ Date __/__/____ Page No. : _____

Result

Clear

Output

```
Hello World

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

Teacher's Signature.....

Topic _____ Date __/__/____ Page No. : _____

2.

Aim

Write a program to print the address in multiple lines.

Software used

Language : C

Compiler : GNU compiler collection

Editor : Programiz

MR

Teacher's Signature _____

Topic _____ Date __/__/____ Page No. : _____

Code

```
#include <stdio.h>
int main()
{
    printf("F Block girls hostel\n");
    printf("Near dudha Devi Mandir\n");
    printf("Bidholi, Dehradun, 248007\n");

    return 0;
}
```

Result

Output Clear

```
F Block girls hostel
Near dudha Devi Mandir
Bidholi,Dehradun,248007

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

M/R

Teacher's Signature _____

Topic: _____ Date: __/__/____ Page No.: _____

3.

Aim

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

Software used

Language : C

Compiler : GNU compiler collection

Editor : Programiz.

M/R

Teacher's Signature: .....

Topic _____ Date __/__/____ Page No. : _____

Code

```
#include <stdio.h>
int main()
{
  char name[50];
  int age;
  printf("Enter your name :\n");
  scanf("%49s", &name);
  printf("Enter your age :\n");
  scanf("%d", &age);
  printf("Hello %s! You are %d years old.\n",
                    name, age);
  return 0;
}
```

Result

Input Clear

```
Enter your name :
 Ekta
Enter your age :
18
Hello Ekta! You are 18 years old.


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

Teacher's Signature _____

Topic _____ Date __/__/____ Page No. _____

4.

Aim

Write a program to add two numbers, take number from user.

Software used

Language : C

Compiler : GNU compiler Collection

Editor : Programiz

MJR

Teacher's Signature _____

Topic _____ Date __/__/____ Page No. : _____

Code

```
#include <stdio.h>
int main()
{
    int a, b, sum;
    printf("Enter first number :");
    scanf("%d", &a);
    printf("Enter second number :");
    scanf("%d", &b);
    sum = a + b;
    printf("Sum of two numbers = %d\n", sum);

    return 0;
}
```

Result

Output

Clear

```
Enter first number : 20
Enter second number :20
Sum of two numbers =40

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

M/R

Teacher's Signature _____

Topic: Experiment-2

AIM

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

SOFTWARE USED

Language : C

Compiler : GNU Compiler Collection

Editor : Programiz

Teacher's Signature.....

Topic _____ Date __/__/____ Page No. _____

CODE

```
#include <stdio.h>
int main()
{
    float l, b, area, peri;
    printf("Enter the length of rectangle");
    scanf("%f", &l);
    printf("Enter the width of rectangle");
    scanf("%f", &b);
    area = l * b;
    peri = 2 * (l + b);
    printf("Area of rectangle = %.2f\n", area);
    printf("Perimeter of rectangle = %.2f\n", peri);

    return 0;
}
```

RESULT

Output Clear

```
Enter the length of rectangle 30.00
Enter the width of rectangle20.00
Area of rectangle =600.00
Perimeter of rectangle =100.00

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

M/R

Teacher's Signature.....

Topic _____ Date __/__/____ Page No. : _____

2.

Aim

Write a program to convert temperature from Celsius to Farenheit using the formula:
$F = (C * 9/5) + 32$.

Software used

Language : C

Compiler : GNU compiler Collection

Editor : Programiz

MJR

Teacher's Signature _____

Topic _____ Date __/__/____ Page No. :_____

Code

```
#include <stdio.h>
int main()
{
    float celsius, fahrenheit;
    printf("Enter temperature in celsius: ");
    scanf("%f", &celsius);
    fahrenheit = (celsius * 9/5) + 32;
    printf("%.2f Celsius = %.2f Fahrenheit\n",
                celsius, fahrenheit);
    return 0;
}
```

Result

Output

Clear

```
Enter temperature in celsius 24.00
24.00 Celsius =75.20 Fahrenheit

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

MJR

Teacher's Signature.....

Experiment - 3.1

Topic

Date / /

Page No.

1.

Aim

Write a program to take check if the triangle is valid or not. If the validity is established, do check if the triangle is isosceles, equilateral, right angle or scalene. Take sides of the triangle as input from a user.

It's Working

(1) Validity check → Uses triangle inequality Theorem (sum of two sides > third side).

(2) Type check :-

- Equilateral : all sides equal
- Isosceles : any two sides equal
- Right angle : uses Pythagoras theorem
- Scalene : all sides different

(3) Order of checking :
Equilateral must be checked before isosceles, because an equilateral triangle also satisfies the isosceles condition.

MJR

Teacher's Signature

Topic _____ Date __/__/____ Page No. :_____

Code

```
#include <stdio.h>
int main
{
   int a, b, c;
   printf ("Enter three sides of a triangle : ");
   scanf ("%d %d %d", &a, &b, &c);
   // Check triangle validity
   if (a+b>c && b+c>a && a+c>b)
   {
      printf ("Triangle is valid\n");
      if (a==b && b==c)
      {
         printf ("Equilateral triangle\n");
      }
      else if ((a==b && b!=c) || (b==c && c!=a) ||
               (a==c && c!=b))
      {
         printf ("Isosceles triangle\n");
      }
      else if ((a*a + b*b == c*c) || (b*b + c*c == a*a)
               || (c*c + a*a == b*b))
      {
         printf ("Right-angled triangle\n");
      }
      else if (a!=b && b!=c && c!=a)
      {
```

Teacher's Signature _____

Topic: Date: / / Page No. :

```
        printf ("Scalene triangle\n");
    }
}
else
{
    printf ("Triangle is not valid\n");
}
return 0;
}
```

Result

Output Clear

```
Enter three sides of a triangle: 3 4 5
Triangle is valid
Right-angled triangle

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

M/R

Teacher's Signature.....

Topic _____ Date __/__/____ Page No. : _____

2. # Aim

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

	BMI
Starvation	< 15
Anorexic	15.1 to 17.5
Underweight	17.6 to 18.5
Ideal	18.6 to 24.9
Overweight	25 to 25.9
Obese	30 to 39.9
Morbidity Obese	40.0 above

Code

```
# include <stdio.h>
int main ()
{
    float w, h, bmi ;
    printf ("Enter the weight in kgs and height
            in mtrs :");
    scanf ("%.2f,%.2f", &w, &h);
```

Teacher's Signature.....

Topic _____ Date __/__/____ Page No. : _____

```
bmi = w/(h*h);
if (bmi <15.0)
{
   printf (" starvation");
}
else if (bmi >=15.1 && bmi <=17.5)
{
   printf (" anorexic ");
}
else if (bmi >=17.6 && bmi <=18.5)
{
   printf (" Underweight");
}
else if (bmi >=18.6 && bmi <=24.9)
{
   printf (" Ideal");
}
else if (bmi >=25.0 && bmi <=25.9)
{
   printf (" Overweight.");
}
else if (bmi >=30.0 && bmi <=39.9)
{
   printf (" Obese");
}
else
{
   printf (" Morbidity Obese");
}
return 0;
}
```

Teacher's Signature.....

Topic

Date __/__/____

Page No. :

Result

Clear

Output

```
Enter the weight in kgs and heightin mtrs : 55.00,175.5
morbidity obese

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

Teacher's Signature.....

3. Topic _____ Date __/__/____ Page No.: _____

Aim

WAP to check if three points (x1, y1), (x2, y2) and (x3, y3) are collinear or not.

Formula of area

$$Area = \frac{1}{2}[x1(y2-y3) + x2(y3-y1) + x3(y1-y2)]$$

- If area = 0 → Points collinear
- Otherwise → Not collinear

Code

```
#include <stdio.h>
int main()
{
   int x1, y1, x2, y2, x3, y3, area;
   printf("Enter Points");
   scanf("%d,%d,%d,%d,%d,%d", &x1, &y1, &x2, &y2, &x3, &y3);
   area = (x1*(y2-y3) + (x2*(y3-y1) + x3*(y1-y2));
   if (area == 0)
   {
```

M/R

Teacher's Signature _____

Topic _____ Date __/__/____ Page No. _____

```
    printf ("Points are collinear");
}
else
{
    printf ("Points are not collinear");
}
return 0;
}
```

Result

Output

Clear

```
Enter points :1,1,1,4,1,5
Points are collinear

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

Topic _____ Date / / Page No. _____

4.

Aim

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

Rules

- Normal year = 365 days → remainder 365%7 = 1
- Leap year = 366 days → remainder 366%7 = 2
- Leap year condition -
  - Divisible by 400 → Leap
  - Divisible by 4 but not by 100 → Leap

Code

```
#include <stdio.h>
int main
{
    int year, i, day = 1;
    printf ("enter year :");
    scanf ("%d", &year);
```

MJR

Teacher's Signature _____

```
for (i=1 ; i<year; i++)
{
 if ((i%400==0) || (i%4==0 && i%100!=0))
 {
   day = (day+2)%7;
 }
 else
 {
   day = (day+1)%7;
 }
printf ("On 01/01/%d it was", year);
switch (day)
{
  case 0: printf ("Sunday\n");
          break;
  case 1: printf ("Monday\n");
          break;
  case 2: printf ("Tuesday\n");
          break;
  case 3: printf ("Wednesday\n");
          break;
  case 4: printf ("Thursday\n");
          break;
  case 5: printf ("Friday\n");
          break;
  case 6: printf ("Saturday\n");
          break;
 }
return 0;
}
```

Teacher's Signature

Topic _____ Date __/__/____ Page No. : _____

Result

Output Clear

```
Enter year: 2025
On 01/01/2025 it was Wednesday


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

MJR

Teacher's Signature _____

Topic _____ Date __/__/____ Page No.: _____

5.

Aim

WAP 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 number of rectangles should be three.

Code

```
# include < stdio.h >
int main ( )
{
   int l1, b1, l2, b2, l3, b3, p1, p2, p3, max ;
   printf ("Enter the lengths and breadths of
            three rectangles : ");
   scanf ("%d %d %d %d %d %d", &l1, &b1, &l2,
          &b2, &l3, &b3);
   p1 = 2 * (l1+b1);
   p2 = 2 * (l2+b2);
   p3 = 2 * (l3+b3);
   max = (p1>p2) ?((p1>p3) ?p1:p3) :((p2>p3)?
                                    p2:p3);
   if (max==p1)
     printf (" Highest perimeter of rectangle 1:");
```

Teacher's Signature _____

Topic _____ Date __/__/____ Page No. :_____

```
else if (max == p2)
{
  printf ("Highest perimeter %d of rectangle 2", max);
}
else
{
  printf ("Highest perimeter %d of rectangle 3",
                                        max);
}
return 0;
}
```

result

Output

Clear

```
Enter the lengths and breadths of three rectangles24 25 45 46 78 79
Highest perimeter 314 of rectangle 3

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

MR

Teacher's Signature.....

Experiment - 11

Topic: Date: / / Page No.:

1.

Aim

WAP to apply bitwise OR, AND and NOT operators on bit level.

Code

```
#include <stdio.h>
int main()
{
  int a, b;
  printf("Enter two integers:");
  scanf("%d %d", &a, &b);
  printf("a = %d , b = %d \n", a, b);
  printf("Bitwise AND (a&b) = %d \n", a&b);
  printf("Bitwise OR (a|b) = %d \n", a|b);
  printf("Bitwise NOT (~a) = %d \n", ~a);
  printf("Bitwise NOT (~b) = %d \n", ~b);

  return 0;
}
```

MJR

Teacher's Signature.....

Topic: Date: __/__/____ Page No. :

Result

Output Clear

```
Enter two integers: 2 4
a = 2, b = 4
Bitwise AND (a & b) = 0
Bitwise OR  (a | b) = 6
Bitwise NOT (~a)    = -3
Bitwise NOT (~b)    = -5

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

Teacher's Signature.....

Topic _____ Date __/__/____ Page No. : _____

2.

Aim

WAP to apply left shift and right shift operator

Explanation

- Left shift $x << n$ → multiply by 2^n
- Right shift $x >> n$ → divide by 2^n

Code

```
#Std include <stdio.h>
int main()
{
    int x, n;
    printf("Enter an integer : ");
    scanf("%d", &x);
    printf("Enter shift amount : ");
    scanf("%d", &n);
    printf("Original number : %d\n", x);
```

MR

Teacher's Signature _____

Topic _____ Date __/__/____ Page No. : _____

```
printf("Left shift (x<<%d) = %d\n", n, x<<n);
printf("Right shift (x>>%d) = %d\n", n, x>>n);

return 0;
}
```

result

Output Clear

```
Enter an integer: 8
Enter shift amount: 6
Original number: 8
Left Shift (x << 6)  = 512
Right Shift (x >> 6) = 0


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

M/R

Teacher's Signature _____

Experiment -3.2

Topic _____ Date __/__/____ Page No. _____

1

Aim

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

Working

- This program takes input number from the user inside a do-while loop.
- For each number, it checks whether it is positive, negative or zero.
- Then continue and at the end display all

Code

```
# include <stdio.h>
int main ()
{
    int num, ch;
    int p, n, z;
    do
    {
        printf (" Enter a number :");
        scanf (" %d", &num);
        if (num > 0)
```

MJR

Teacher's Signature _____

Topic _____ Date __/__/____ Page No. :_____

```
        p++;
    else if (num<0)
        n++;
    else
        z++;
    printf ("Want to enter another number ?
                (1 = Yes / 2 = No) :");
    scanf ("%d", &ch);
}
while (ch==1)
printf ("Positive number :%d\n", p);
printf ("Negative number :%d\n", n);
printf ("Zeros number :%d\n", z);
return 0;
}
```

Output

Output Clear

```
Enter a number: 87
Do you want to enter another number? (1 = Yes / 0 = No): 1
Enter a number: 67
Do you want to enter another number? (1 = Yes / 0 = No): 0
Positive numbers: 2
Negative numbers: 0
Zeroes: 0


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

MR

Teacher's Signature _____

Topic _____ Date __/__/____ Page No. : _____

2. Aim

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

Code

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

  return 0;
}
```

MJR

Teacher's Signature.....

Topic_____ Date ___/___/_____ Page No. : _____

Output

Output Clear

```
Enter a number : 12
12 * 1 = 12
12 * 2 = 24
12 * 3 = 36
12 * 4 = 48
12 * 5 = 60
12 * 6 = 72
12 * 7 = 84
12 * 8 = 96
12 * 9 = 108
12 * 10 = 120


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

M/R

Teacher's Signature.....

Topic: Date: / / Page No:

3.(a)

HIM

WAP to generate the following set of output.

a.

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

Code

```
# include <stdio.h>
int main()
{
  int n=1 , r=3 ;
  for (int i=1 ; i<=r ; i++)
  {
    for (int s=1 ; s<=r-i ; s++)
      printf (" ");
  }
  for (int j=1 ; j<=i ; j++)
  {
      printf ("%d" , n);
        n++;
  }
  printf ("\n");
```


Teacher's Signature.....

Topic _____ Date __/__/____ Page No. :____

```
}
    return 0;
}
```

Output

Output Clear

```
1
2 3
4 5 6

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

(b)

b.

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

```
# include <stdio.h>
int main()
{
    int rows=5;
```


Teacher's Signature _____

Topic _____ Date __/__/____ Page No. : _____

```
for (int i=0; i< rows; i++)
{
  for (int s=0; s< rows-i; s++)
  {
    printf (" ");
  }
  int val=1;
  for (int j=0; j<=i; j++)
  {
    printf ("%d", val);
    val = val * (i-j)/(j+1);
  }
  printf ("\n");
}
return 0;
}
```

Output

Output

Clear

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

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

MJR

Teacher's Signature.....

Topic _____ Date __/__/____ Page No. :_____

A. AIM

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

Code

```
#include <stdio.h>
int main()
{
    float p = 100000.00;
    float a;
    for (int i = 1; i <= 10; i++)
    {
        a = p + p * 0.1;
    }
    printf ("%.2f", a);
    return 0;
}
```

MJR

Teacher's Signature.....

Output

Output Clear

```
110000.00

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

Rishi

PAGE No.:
DATE

5.

Aim

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.

Ex - 1729

$12^3 + 1^3$ and $10^3 + 9^3$ for a no $L = 20$ (that is limit)

Explanation

- we loop through all pairs (a, b) and (c, d) up to limit L
- Compute their sums of cubes.
- If two different pairs give the same sum, that sum is a Ramanujan number.
- Print the result in the form
  $1729 = 12^3 + 1^3 = 10^3 + 9^3$

Code

```
# include <stdio.h>
int main ()
{
    int L, a, b, c, d;
    int sum1, sum2;
```

Teacher's Signature :

Rysh

PAGE No.:
DATE

```
printf ("Enter the limit (L) : ");
scanf ("%d", &L);

printf ("Ramanujan numbers up to limit
        %d are : \n", L);

for (a=1 ; a<=L; a++)
{
  for (b=a+1 ; b<=L; b++)
  {
      sum1 = a*a*a + b*b*b;

  for (c=a+1 ; c<=L; c++)
{
  for (d=c+1 ; d<=L; d++)
  {
    sum2 = c*c*c + d*d*d;

  if ( sum1 == sum2)
{
  printf ("%d = %d^3+%d^3 = %d^3 + %d^3\n",
  sum1, a, b, c, d);
  }
  }
  }
  }
}
return 0;
}
```

Teacher's Signature :

Rush

PAGE No.

DATE

Output

Output Clear

```
Enter the limit (L): 20
Ramanujan numbers up to limit 20 are:
1729 = 1^3 + 12^3 = 9^3 + 10^3
4104 = 2^3 + 16^3 = 9^3 + 15^3


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

Teacher's Signature :

Experiment-4

1.

Aim

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

Code

```
# include <stdio.h>
  void gruet ();
  int a = 20;
  int main ()
{
    printf ("%d \n", a);
    gruet ();
    printf ("%d \n", a);
   return 0;
}
  void gruet()
{
    a = 50;
    printf ("%d \n", a);
}
```

Output

Output

Clear

```
20
50
20


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

2. Aim

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.

code

```
#include <stdio.h>
void greet ();
int a=5000;
int main ()
{
    int a=20;
    printf ("%d\n", a);
  greet ();
    printf ("%d\n", a);
    return 0;
}
void greet()
{
    int x=50;
    printf ("%d\n", a);
}
```

output

Output

Clear

```
20
50
50


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

3.

Aim

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

Code

```
# include <stdio.h>
  void greet();
  int main()
{
   int a=20;
   {
      int b=150;
      printf("%d\n", b);
   }
   printf("%d\n", a);
   return 0;
}
```

OUTPUT

Output Clear

```
150
20

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

4. # Aim

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

Code

```
#include <stdio.h>
void greet();
int main();
{
    greet();
    greet();
    greet();
    return 0;
}
void greet()
{
    static int count = 0;
    count++;
    printf("%d\n", count);
}
```

OUTPUT

Output

Clear

```
1
2
3

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

Experiment -6

1. **AIM**

Develop a recursive and non-recursive function FACT (num) to find the factorial of a number, n! defined by FACT(n) = 1, if n=0.0 Otherwise, FACT (n) = n * FACT (n-1). Using this function, write a C program to compute the binomial coefficient. Tabulate the results for different values of n and r with suitable messages.

Code

```
#include <stdio.h>
int FACT (int);
int bincoeff (int n, int r);
int main()
{
  int n, r, num;
  printf ("Enter a number : \n");
  scanf ("%d", &num);
  int a = FACT (num);
  printf ("Factorial of a number : %d \n", a);
  printf ("Enter the value of n and r : \n");
  scanf ("%d %d", &n, &r);
  float z = bincoeff(n, r);
  printf ("Binary coefficient of a number : %.1f", z);
  return 0;
}
```

```
int FACT ( int num)
{
  if (num == 0)
   return 1;
  else
    return num * FACT (num - 1);
}
int bincoeff (int n, int r)
{
  float C = (FACT (n)) / (FACT (r) x (FACT (n-r)));
   return C;
}
```

Output

Output

Clear

```
Enter a number:
5
Factorial of a number:120
Enter the value of n and r:
4 2
Binary coefficient of a number:6.0

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

2. # AIM

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

Code

```
#include <stdio.h>
int GCD (int num1, int num2);
int main ()
{
   int num1, num2;
   printf ("Enter two numbers:\n");
   scanf ("%d %d", &num1, &num2);
   int a = GCD (num1, num2);
   printf ("Greatest common divisor of two
          given integers : %d\n", a);
   return 0;
}
   int GCD (int num1, int num2)
   {
      if (num2 == 0)
         return num1;
      else
         return GCD (num2, num1 % num2);
   }
```

output

Output

Clear

```
Enter two numbers:
12 18
Greatest common divisor of two given integers:6


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

3. # Aim

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.

Code

```
# include <stdio.h>
int FIBO (int num);
int main ()
{
  int i, a, num;
  printf ("Enter the limit upto which fibo
          series print :");
  scanf ("%d", &num);
  printf ("Fibonacci series :");
  for (i=0; i<num ; i++)
  {
    int a = FIBO (i);
    printf ("%d", a);
  }
  printf ("\n");
  return 0;
}
int FIBO (int num)
{
  if (num == 0)
```

```
return 0;
if (num == 1)
{ return 1;
else
    return FIBO(num-2) + FIBO(num-1);
}
}
```

output

Output

Clear

```
Enter the limit upto which fibo series print:6
Fibonacci Series: 0 1 1 2 3 5

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

4.

Aim

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

Code

```
#include <stdio.h>
int ISPRIME (int num);
int main()
{
    int lower, upper, i;
    printf ("enter lower limit :");
    scanf ("%d", &lower);
    printf ("enter upper limit :");
    scanf ("%d", &upper);
    printf ("Prime numbers between %d and
            %d are :\n", lower, upper);
    for (i = lower ; i <= upper ; i++)
    {
        if (ISPRIME (i))
        {
            printf ("%d ", i);
```

```
    }
}
    printf("\n");
    return 0;
}
int ISPRIME(int num)
{
    int i;
    if (num <=1)
    return 0;

    for (i=2; i*i<=num; i++)
    {
        if (num%i==0)
        {
            return 0;
        }
    }
    return 1;
}
```

Output

Output Clear

```
Enter lower limit: 10
Enter upper limit: 30
Prime numbers between 10 and 30 are:
11 13 17 19 23 29


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

Rush

PAGE No.:
DATE

5.

Aim

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.

Code

```
# include <stdio.h>
    REV
# include <string.h>
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:");
```

Teacher's Signature :

Rush

PAGE No.:

DATE

```
scanf ("%.[^\n]", str);
                    //Read string including spaces

REVERSE (str);
printf ("Reversed string : %s\n", str);
return 0;
}
```

Output

Output Clear

```
Enter a string: Hello World
Reversed string: dlroW olleH


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

Experiment -5

1. Aim

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

Code

```
# include < stdio.h >
 int main ()
{
  int i, a[50], n, j, temp;
  printf (" Enter the size of the array\n");
  scanf ("%d", &n);
  printf ("Enter the elements in array :\n");
  for (i=0; i<n; i++)
  {
      printf ("a[%d] ", i);
      scanf ("%d", &a[i]);
  }
  for (i=0; i<n-1; i++)
  {
     for (j=0; j<n-1-i; j++)
     {
        if (a[j] > a[j+1])
        {
           temp = a[j];
```

```
            a[j] = a[j+1];
            a[j+1] = temp;
        }
    }
}
printf("second largest integer is :\n");
printf("%d", a[n-2]);
return 0;
}
```

output

Output Clear

```
Enter the size of the array:
5
Enter the elements in array:
a[0] 10
a[1] 46
a[2] 34
a[3] 78
a[4] 90
Second largest element is:
78

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

2. Aim

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

Code

```
#include <stdio.h>
int main()
{
    int i, a[50], even, odd, pos, neg, n;
    printf("Enter the size of the array:\n");
    scanf("%d", &n);
    printf("Enter the elements in array:\n");
    for (i=0; i<n; i++)
    {
        printf("a[%d]", i);
        scanf("%d", &a[i]);
    }
    even = 0;
    odd = 0;
    pos = 0;
    neg = 0;
    for (i=0; i<n; i++)
    {
```

```
    if (a[i] > 0)
{
    pos++;
}
else
{
    neg++;
}
}
for (i=0; i<n; i++)
{
    if (a[i]%2 == 0)
    {
        even++;
    }
else
{
    odd++;
}
}
printf("Positive numbers present in array is : %d", pos);
printf("Negative numbers present in array is : %d", neg);
printf("even numbers present in array is : %d", even);
printf("odd numbers present in array is : %d", odd);
return 0;
}
```

classmate
Date :
Page :

Output

Output Clear

```
Enter the size of the array:
7
Enter the elements in array:
a[0] 45
a[1] 67
a[2] -90
a[3] 87
a[4] 22
a[5] 65
a[6] -8
Positive numbers present in array is:5
Negative numbers present in array is:2
Even numbers present in array is:3
Odd numbers present in array is:4


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

3.

Aim

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

Code

```
# include <stdio.h>
int main()
{
   int i, a[50], n, k;
   printf ("Enter the size of the array :\n");
   scanf ("%d", &n);
   printf ("Enter the elements in array :\n");
   for (i=0; i<n; i++)
   {
      printf ("a[%d]", i);
      scanf ("%d", &a[i]);
   }
   printf ("Enter the number whose frequency is
                    checked :\n");
   scanf ("%d", &k);
   for (i=0; i<n; i++)
   {
```

```
        if (a[i] == k)
            f += 1;
            break;
    }
    printf ("Frequency of %d in array is : %d", k, f);
    return 0;
}
```

Output

```
Output                                                    Clear

Enter the size of the array:
5
Enter the elements in array:
a[0] 1
a[1] 2
a[2] 3
a[3] 4
a[4] 2
Enter the number whose frequency is checked:
2
Frequency of 2 in array is: 2


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

Rush

PAGE No.:

DATE

4. # Aim

WAP that reads two matrices A(m x n) and B(p x q) and computes the product A and B. Read matrix A and B in row major order respectively. Print both the input matrices and resultant matrix with suitable headings and output should be in matrix format only. Program must check the compatibility of orders of the matrices for multiplication. Report appropriate message in case of incompatibility

Code

```
# include <stdio.h>
 int main ()
{
  int m, n, p, q, i, j, k;
  printf ("Enter the order of matrix A(m n):\n");
  scanf ("%d %d", &m, &n);
  int a [m] [n];
  printf ("Enter elements in matrix A:\n");
  for (i=0 ; i< m ; i++)
  {
      for (j=0; j<n ; j++)
      {
          scanf ("%d", &a [i] [j]);
```

Teacher's Signature :

Rysh

PAGE No.:
DATE

```
    }
}
printf("Enter the order of matrix B(p q):\n");
scanf("%d %d", &p, &q);

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

if (n!=p)
{
printf("Matrix multiplication is not possible.\n");
printf("Reason: No. of columns of A (%d) != No. of
        rows of B (%d)\n");
return 0;
}

int result[m][q];

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

Teacher's Signature :

```
for (k=0 ; k< n ; k++)
{
  result[i][j]= result[i][j]+a[i][k]*
                b[k][j];
}
}
}

printf ("\n Matrix A: \n");
for (i=0; i<m ; i++)
{
  for (j=0; j<n ; j++)
  {
    printf ("%d ", a[i][j]);
  }
}

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

```
            printf("%d", result[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

Output

```
Output                                                    Clear

Enter rows and columns of matrix A (m n): 2 2
Enter rows and columns of matrix B (p q): 2 2
Enter elements of matrix A (2x2) in row-major order:
1
2
3
4
Enter elements of matrix B (2x2) in row-major order:
5
6
7
8
Matrix A:
1   2
3   4
Matrix B:
5   6
7   8
Product Matrix (A x B):
19  22
43  50
```

Experiment - 8

Rysh

PAGE No.:

DATE

1. Aim

Declare different 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.

Code

```
#include <stdio.h>
int main()
{
    int a=10;
    float b=3.14;
    char c='A';
    int *p = &a;
float *p1 = &b;
    char *p2 = &c;
    printf ("Variable values :\n");
    printf ("a = %d\n", a);
    printf ("b = %.2f\n", b);
    printf ("c = %c\n", c);
        printf ("Pointer values :\n");
        printf (" p = %p\n", p);
        printf (" p1 = %p\n", p1);
        printf (" p2 = %p\n", p2);
```

Teacher's Signature :

```
printf("Values using dereferencing:\n");
printf("*p = %d\n", *p);
printf("*p1 = %.2f\n", *p1);
printf("*p2 = %c\n", *p2);

return 0;
}
```

Output

Output Clear

```
Variable values:
a = 10
b = 3.14
c = X
Pointer values:
p1 = 0x7ffe487e56f4
p2 = 0x7ffe487e56f0
p3 = 0x7ffe487e56ef
Values using dereferencing:
*p1 = 10
*p2 = 3.14
*p3 = X


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

Teacher's Signature :

Rush

PAGE No.:

DATE

2.

Aim

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

Code

```
#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);
```

Teacher's Signature :

Rysh

PAGE No.:

DATE

```
printf ("p3 = %p , %c \n", p3 , *p3);
p1--;
p2--;
p3--;
printf ("\n After Decrement \n");
printf (" p1 = %p , %d \n", p1, *p1);
printf (" p2 = %p , %.1f \n", p2 , *p2);
printf (" p3 = %p , %c \n", p3 , *p3);
return 0;
}
```

Output

Output Clear

```
Original Addresses:
p1 = 0x7fff90e81bec
p2 = 0x7fff90e81be0
p3 = 0x7fff90e81bdd

After Increment:
p1 = 0x7fff90e81bf0 , 20)
p2 = 0x7fff90e81be4 , 2.2)
p3 = 0x7fff90e81bde , B)

After Decrement (back to original):
p1 = 0x7fff90e81bec , 10)
p2 = 0x7fff90e81be0 , 1.1)
p3 = 0x7fff90e81bdd , A)


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

Teacher's Signature :

Page
PAGE No.
DATE

3.

Aim

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

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

Code

```
#include <stdio.h>
int * 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);
```

Teacher's Signature :

Rishi

PAGE No.

DATE

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

Output

Output Clear

```
Before swap:
a = 5, b = 10

After swap:
a = 10, b = 5


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

Teacher's Signature :

Rysh

PAGE No.:
DATE

Experiment - 7

1. ## AIM

Write a C program 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 numbers

Code

```
#include <stdio.h>
struct complex
{
    float real;
    float imag;
};

struct complex readComplex()
{
    struct complex c;
    printf("Enter real part:\n");
    scanf("%f", &c.real);
    printf("Enter imaginary part:");
    scanf("%f", &c.imag);
    return c;
}

void writecomplex(struct complex c)
{
```

Teacher's Signature :

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

struct complex addcomplex (struct complex a,
                           struct complex b)
{
    struct complex result;
    result.real = a.real + b.real;
    result.imag = a.imag + b.imag;
    return result;
}

struct complex subtractcomplex (struct
                    complex a, struct complex b)
{
    struct complex result;
    result.real = a.real - b.real;
    result.imag = a.imag - b.imag;
    return result;
}

int main ()
{
    struct complex c1, c2, sum, diff;
    printf ("Enter first complex number:\n");
```

Rush

PAGE No.:

DATE

```
c1 = readComplex();
printf("Enter second complex number:\n");
c2 = readComplex();
sum = addComplex(c1, c2);
diff = subtractComplex(c1, c2);

printf("\nFirst complex number: ");
writeComplex(c1);
printf("\nSecond complex number: ");
writeComplex(c2);
printf("\nAddition: ");
writeComplex(sum);
printf("Subtraction: ");
writeComplex(diff);

return 0;
}
```

Output

Output Clear

```
Enter first complex number:
Enter real part: 6.00
Enter imaginary part: 14.00
Enter second complex number:
Enter real part: 4.00
Enter imaginary part: 12.00

First Complex Number: 6.00 + 14.00i
Second Complex Number: 4.00 + 12.00i

Addition: 10.00 + 26.00i
Subtraction: 2.00 + 2.00i


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

Rush

PAGE No.:

DATE

2. # Aim

Write a C program 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. Gross salary (basic pay + DA). Print the employees name and gross salary.

Code

```
#include <stdio.h>
struct Employee
{
    char name[50];
    float basic_pay;
    float gross_salary;
};

int main()
{
    struct Employee emp[3];
    int i;
    printf("Enter details of 3 employees:\n");
    for(i=0; i<3; i++)
    {
        printf("\nEmployee %d Name: ", i+1);
        scanf("%s", emp[i].name);
        printf("Basic pay: ");
        scanf("%f", &emp[i].basic_pay);
```

Teacher's Signature :

Rajsh

PAGE No.

DATE

```
float da = 0.52 * emp[i].basic_pay;
emp[i].gross_salary = emp[i].basic_pay +
                      da;
printf("\n---Employee Gross Salary
        List---\n");
for (i=0 ; i<3; i++)
{
    printf("Name : %s \t Gross Salary:%.2f \n", emp[i].name,
           emp[i].gross_salary);
}
return 0;
}
```

Output

Output Clear

```
Enter details of 3 employees:

Employee 1 Name: RAJEEV
Basic Pay: 30000

Employee 2 Name: SANJEEV
Basic Pay: 45000

Employee 3 Name: AMIT
Basic Pay: 50000

--- Employee Gross Salary List ---
Name: RAJEEV     Gross Salary: 45600.00
Name: SANJEEV    Gross Salary: 68400.00
Name: AMIT   Gross Salary: 76000.00


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

Rysh

PAGE No.:

DATE

3. Aim

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

Code

```
#include <stdio.h>
struct Book {
    int book_id;
    char title[50];
    char author[50];
    float price;
};

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

int main()
{
    struct Book b1;
```

Teacher's Signature :

Rush

PAGE No.:

DATE

```
printf ("Enter Book ID: ");
scanf ("%d", &b1.book_id);
printf ("Enter Title: ");
scanf ("%s", &b1.title);
printf ("Enter Author: ");
scanf ("%s", &b1.author);
printf ("Enter Price: ");
scanf ("%f", &b1.price);

displayBook (b1);
return 0;
}
```

Output

Output Clear

```
Enter Book ID: 21
Enter Title: POWERLESS
Enter Author: LAUREN
Enter Price: 509

Book Details:
ID: 21
Title: POWERLESS
Author: LAUREN
Price: 509.00


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

Teacher's Signature :

Rush

PAGE No.:

DATE

4.

Aim

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

Code

```
#include <stdio.h>
union address
{
    char name [50];
    char home-address [100];
    char hostel address [100];
    char city [50];
    char state [50];
    char zip [50];
};

int main ()
{
    union Address addr;
    printf ("Enter Name: ");
    scanf ("%s", addr.name);
    printf ("Name: %s\n", addr.name);

    printf ("Enter Home Address: ");
    scanf ("%s", addr.home-address);
```

Teacher's Signature :

Rush

PAGE No.:

DATE

```
printf ("Enter Hostel Address: ");
scanf ("%s", addr.hostel_address);
printf ("Hostel Address: %s\n", addr.hostel_
                address);

printf ("Enter City: ");
scanf ("%s", addr.city);
printf ("City: %s\n", addr.city);

printf ("Enter State: ");
scanf ("%s", addr.state);
printf ("State: %s\n", addr.state);

printf ("Enter ZIP code: ");
scanf ("%s", addr.zip);
printf ("ZIP code: %s\n", addr.zip);
return 0;
}
```

Output

Output

Clear

```
Enter Name: EKTA BHARDWAJ
Name: EKTA BHARDWAJ
Enter Home Address: ADARSH NAGAR
Home Address: ADARSH NAGAR
Enter Hostel Address: F BLOCK GIRLS HOSTEL
Hostel Address: F BLOCK GIRLS HOSTEL
Enter City: DEHRADUN
City: DEHRADUN
Enter State: UTTARAKHAND
State: UTTARAKHAND
Enter ZIP Code: 248007
ZIP Code: 248007


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

Experiment - 9

Rush
PAGE No.:
DATE

1.

AIM

Write a program to create a new file and write text into it

Code

```
#include <stdio.h>
int main()
{
    File *fp;
    char text[200];
    fp = fopen("myfile.txt", "w");
    if (fp == NULL)
    {
        printf("Error creating file!");
        return 1;
    }
    printf("Enter text to write into the file:\n");
    fgets(text, sizeof(text), stdin);
    fputs(text, fp);
    fclose(fp);
    printf("File created and written
                successfully!\n");
```

Teacher's Signature :

Rishi

PAGE No.:

DATE

return 0;
}

Output

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\HP\OneDrive\Documents\100daysc> cd "c:\Users\HP\OneDrive\Documents\100days
 if ($?) { .\d77 }
Enter text to write into the file:
hello upes
File created and text written successfully!
PS C:\Users\HP\OneDrive\Documents\100daysc>
```

Teacher's Signature :

Rysh

PAGE No :

DATE

2. **Aim**

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

Code

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

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

Teacher's Signature :

Rush

PAGE No.:

DATE

Output

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\HP\OneDrive\Documents\100days\c> cd "c:\Users\HP\OneDrive\Documents\100days\c\" ; if ($?) { gcc d77.c -o d77 } ;
 if ($?) { .\d77 }
File content:
hello upes
PS C:\Users\HP\OneDrive\Documents\100days\c>
```

Teacher's Signature :

Rushi

PAGE No.:

DATE

3.

Aim

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

Code

```
#include <stdio.h>
int main()
{
    FILE *fp;
    char line[200];
    fp = fopen("myfile.txt", "r");
    if (fp == NULL)
    {
        printf("File not found!");
        return 1;
    }

    printf("Reading file line by line:\n");
    while (fgets(line, sizeof(line), fp))
    {
        printf("%s", line);
    }
    fclose(fp);
    return 0;
}
```

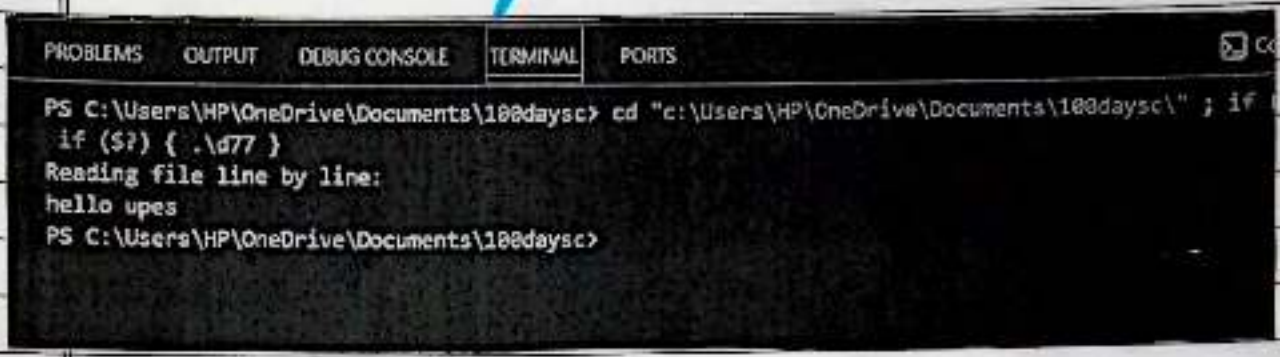
Teacher's Signature :

Rush

PAGE No.:

DATE

Output



Teacher's Signature :

Rysh

PAGE No.

DATE

Experiment - 10

1. **Aim**

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

Code

```
#include <stdio.h>
#include <stdlib.h>
struct Node
{
    int data;
    struct Node *next;
};

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

    head = (struct Node*)malloc(sizeof(struct Node));
    second = (struct Node*)malloc(sizeof(struct Node));
    third = (struct Node*)malloc(sizeof(struct node));
    head -> data = 10;
    head -> next = second;
    second -> data = 20;
    second -> next = third;
    third -> data = 30;
    third -> next = NULL;
```

Teacher's Signature :

Rysh

PAGE No.:

DATE

```
printf ("Linked List : ");
Traverse() Node *temp = head;
while (temp != NULL)
{
    printf ("%d -> ", temp -> data);
    temp = temp -> next;
}
printf ("NULL\n");
return 0;
}
```

Output

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\HP\OneDrive\Documents\100daysc> cd "c:\Users\HP\OneDrive\Do
 if ($?) { .\d77 }
Linked List: 10 -> 20 -> 30 -> NULL
PS C:\Users\HP\OneDrive\Documents\100daysc>
```

Teacher's Signature :

Rysh

PAGE No.:

DATE

2.

Aim

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

Code

```
#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(sizeof
                  (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!\n");
```

Teacher's Signature :

```
    return ;
}
    temp = temp -> next;
}
    newNode -> next = temp -> next;
    temp -> next = newNode;
}
int main()
{
struct Node *head = NULL, *second = NULL,
    *third = NULL;
head = (struct Node*) malloc (sizeof (struct Node));
second = (struct Node*) malloc (sizeof (struct Node));
third = (struct Node*) malloc (sizeof (struct Node));

head -> data = 10;
head -> next = second;
second -> data = 20;
second -> next = third;
third -> data = 30;
third -> next = NULL;
printf ("Original List : 10 -> 20 -> 30 -> NULL \n");
insertMiddle (head, 15, 1);
printf ("Updated List : ");
struct Node *temp = head;
while (temp != NULL)
{
    printf ("%d -> ", temp -> data);
    temp = temp -> next;
}
```

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

Output

```
Output                                                    Clear

Original List: 10 -> 20 -> 30 -> NULL
Updated List: 10 -> 15 -> 20 -> 30 -> NULL


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

Experiment - 12

Rysb

PAGE No.:
DATE

1. **Aim**

Write a program to define some constant variable in C preprocessor.

Code

```
#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;
}
```

Output

```
Output                                                    Clear

PI = 3.14
MAX = 100
MIN = 1


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

Teacher's Signature :

Rush

PAGE No.:

DATE

2. Aim

Write a program to define a function in directives.

Code

```
#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;
}
```

Output

Output Clear

```
Square of 5 is 25

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

Teacher's Signature :

Rysh

PAGE No.:

DATE

Experiment - 13

1. Aim

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

Code

```
#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("Multiplication = %d\n", MUL(x,y));
    printf("Division = %d\n", DIV(x,y));
    return 0;
}
```

Output

```
Output                                                    Clear

Addition = 24
Subtraction = 16
Multiplication = 80
Division = 5


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

Experiment - 14

Rysh

PAGE No.:

DATE

1. AIM

Write a program to create a static library for performing arithmetic functions

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;
}
```

Teacher's Signature :

Rysh

PAGE No.

DATE

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

Steps to create static library :

1. Compile the source file

   gcc -c arith.c

2. Create the static library

   ar rcs libarith.a arith.o

3. static library created

   libarith.a

Teacher's Signature :

Rysh

PAGE No.

DATE

2. **Aim**

Write a program to use static library in other program.

FILE : main_static.c

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

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

Compilation

```
gcc main_static.c -L. -larith -o static_app
```

Run

```
./static_app
```

Output

```
Add = 15
Sub = 5
Mul = 50
Div = 2
```

Teacher's Signature :

Rishi

PAGE No.:

DATE

Experiment - 15

1. **Aim**

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

FILE 1 : arith_shared.h

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

FILE 2 : arith_shared.c

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

Teacher's Signature :

Rush

PAGE No.:

DATE

Steps to create shared library

1. Compile with position - independent code

```
gcc -fPIC -c arith_shared.c
```

2. Create shared library

```
gcc -shared -o libarith.so arith_shared.o
```

shared library created.

libarith.so

Teacher's Signature :

Rish

PAGE No.

DATE

2. **Aim**

Write a program to use shared library in other program

FILE : main_shared.c

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

int main()
{
    int a = 10 , b = 5;
    printf("Add = %d\n", add(a,b));
    printf("Sub = %d\n", sub(a,b));
    printf("Mul = %d\n", mul(a,b));
    printf("Div = %d\n", divide(a,b));

    return 0;
}
```

Compilation

```
gcc main_shared.c -L. -larith -o shared_app
```

Run

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

Teacher's Signature :

Rushi

PAGE No.:

DATE

Output

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

Teacher's Signature :