

Experiment No. 1 Name: Write a C program to print "HelloWorld"

PAGE NO.:

DATE: / /

- Aim:- Print "Hello World" Using C-Language

- Codes:-

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

```
int main()
{
    printf("Hello World");
    return 0;
}
```

- Terminal (Output):-

→ PS D:\Code Playground\Gojo Satoru> gcc 01-first.c

→ PS D:\Code Playground\Gojo Satoru> ./a.exe

Hello World

Teacher's Signature: _____

Experiment No. 2

Name: Write a c Program to print the address

PAGE NO.:

DATE: / /

- Aim :- To Print the address in Multiple lines (New line)

- Codes :-

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

```
int main() {  
    printf("Hello World \n");  
    return 0;  
}
```

- Terminal :-

Hello World

Teacher's Signature: _____

Experiment No. 3 Name: _____

PAGE NO.:

DATE: / /

- Aim:- To prompt the user to enter their name and age.

- Codes:-

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

```
int main() {  
    char name[19];  
    int age;
```

```
    printf("Ayush:");  
    scanf("%18s", name);
```

```
    printf("18:");  
    scanf("%d", &age);
```

```
    printf("\nHello %s! You are %d years old.\n", Ayush, 18);
```

```
    return 0;
```

```
}
```

- Terminal (Output):-

Hello Ayush! You are 18 years old.

Teacher's Signature: _____

Experiment No. 4

Name: _____

PAGE NO.:

DATE: / /

- Aim:- To add two Numbers, take number from user Using C-Language.

- Codes:-

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

```
int main () {
```

```
    int num1, num2, sum;
```

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

```
    scanf ("%d", &num1);
```

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

```
    scanf ("%d", &num2);
```

```
    sum = num1 + num2;
```

```
    printf ("\n The sum of %d and %d is : %d \n", num1, num2, sum);
```

```
    return 0;
```

```
}
```

- Terminal (Output):-

Example, first term is \Rightarrow 12

second term is \Rightarrow 8

\rightarrow The sum of 12 and 8 is : 20

Teacher's Signature: _____

Experiment No. 1

Name: _____

PAGE NO.:

DATE: / /

Experiment 2 :- Operator

- Aim:- To calculate the area and perimeter of a rectangle based on its length and width using C-Language

- Codes:-

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

```
int main() {
```

```
    float length, breadth, area, perimeter;
```

```
    printf("Enter the length of the rectangle: ");
```

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

```
    printf("Enter the breadth of the rectangle: ");
```

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

```
    area = length * breadth;
```

```
    perimeter = 2 * (length + breadth);
```

```
    printf("Area of the rectangle = %.2f \n", area);
```

```
    printf("Perimeter of the rectangle = %.2f \n", perimeter);
```

```
    return 0;
```

```
}
```

Teacher's Signature: _____

Terminal (Output) :-

- Area of rectangle = length \times breadth
- Perimeter of rectangle = $2 \times (l + b)$.

Eg:- $l = 10, b = 5$

Area of Rectangle = 50

Perimeter of the Rectangle = 30

{ (" : random width ") ; funcy
((1 min & " 10 ") ; funcy

{ (" : random height ") ; funcy
((5 min & " 10 ") ; funcy

length + brea = area

random("a/b") ; a/b
let brea be to one off a/b ; funcy

- Aim:- To convert Celsius to Fahrenheit using the formula:
 $F = (C * 9/5) + 32$.

- Codes:-

```
#include <stdio.h>
int main()
{
    float celsius, fahrenheit;
    printf("Enter temperature in Celsius:");
    scanf("%f", &celsius);
    fahrenheit = (celsius * 9/5) + 32;
    printf("%0.2f Celsius = %0.2f Fahrenheit\n", celsius,
           fahrenheit);
    return 0;
}
```

- Terminal (Output):-

Eg:- Enter temp in celsius : 37
37 Celsius = 98.6 Fahrenheit

Terminal :- gcc (file name . c)

Enter three sides of triangle : 5 5 5

The triangle is Valid.

It is an equilateral triangle

a tab

Enter three sides of triangle : 5 5 7

The triangle is Valid.

It is an Isosceles triangle

a tab

Enter three sides of triangle 3 4 5

The triangle is Valid

It is a Scalene triangle

a tab

Enter three sides of triangle 3 4 5

The triangle is Valid.

It is a Right angle triangle

a tab

Enter three sides of triangle 2 3 10

The triangle is not Valid.

2. sides of first and second
third side > sum of sides

- Aim:- To check if the triangle is valid or not. If the validity is established, to check if the triangle is isosceles, equilateral, right angled, or scalene. Take sides of the triangle as input from a user using C language.

- Codes:- (Program to check for establishment of triangle)

```
#include <stdio.h>
int main()
{
    int a, b, c;
    printf("Enter three sides of a triangle:");
    scanf("%d %d %d", &a, &b, &c);
    if ((a+c>b) && (a+c>b) && (b+c>a))
    {
        printf("The triangle is Valid.\n");
        if (a==b && b==c)
            printf("It is Equilateral triangle.\n");
    }
    else if (a==b || b==c || a==c)
        printf("It is an Isosceles triangle.\n");
    else if ((a*a == b*b + c*c) || (b*b == a*a + c*c) ||
              (c*c == a*a + b*b))
        printf("It is a Right angle triangle.\n");
    else
        printf("It is an Scalene triangle.\n");
}
else
    printf("The triangle is not Valid.\n");
return 0;
```

Teacher's Signature: _____

Terminal :- gcc (file name) Enter coordinates of first point (x_1, y_1): 1
Enter coordinates of second point (x_2, y_2): 2
Enter coordinates of third point (x_3, y_3): 3
The points are collinear.

(a tab) The point are collinear

Enter coordinates of first point (x_1, y_1): 0 0

Enter coordinates of second point (x_2 , y_2): 2 3

Enter coordinates of third point (x_3, y_3): 4 5

The points are not collinear.

al point to take with him? (part 2)

}((08344) 83 & (08378) 88 (08333))

"(A body is almost 97% water)

(2nd Bl. 3rd)

(at a point located at $t = 0$) from

{ (x:=a || x:=b || d := e) } b d

L. ("Lindbergia oblonga" sp. n.)

$$\|(\phi^* z + \phi^* w - d^* d)\| \|(\phi^* z + d^* d - w^* w)\| + \|d\|^2$$

$$j((d^2d+e^2e+z^2z))$$

L'admiral du Bosc (1754)

6. (Int. deposit orders on a flat) Point

Wann kann ich den ersten Tag?

- Aim:- WAP to check if three points (x_1, y_1) , (x_2, y_2) and (x_3, y_3) are collinear or not using C Language.

- Codes:-

```
#include <stdio.h>
int main() {
    int x1, y1, x2, y2, x3, y3;
    printf("Enter coordinates of first point (x1, y1) : ");
    scanf("%d %d", &x1, &y1);
    printf("Enter coordinates of second point (x2, y2) : ");
    scanf("%d %d", &x2, &y2);
    printf("Enter coordinates of third point (x3, y3) : ");
    scanf("%d %d", &x3, &y3);
    int area = (x1 * (y2 - y3) + x2 * (y3 - y1) + x3 * (y1 - y2));
    if (area == 0) {
        printf("The points are collinear.\n");
    } else {
        printf("The points are not collinear.\n");
    }
    return 0;
}
```

- Aim:- WAP to compute the BMI index of the person and print the BMI values as per the following ranges using C language.

- Codes:-

```
#include <stdio.h>
int main() {
    float weight, height; bmi;
    printf("Enter weight (in Kg): ");
    scanf("%f", &weight);
    printf("Enter height (in meters): ");
    scanf("%f", &height);
    bmi = weight / (height * height);
    printf("Your BMI is : %.2f\n", bmi);
    if (bmi < 15) {
        printf("Category: Starvation\n");
    }
    else if (bmi >= 15 & & bmi < 17.5) {
        printf("Category: Anorexic\n");
    }
    else if (bmi >= 17.5 & & bmi < 18.5) {
        printf("Category: Normal weight\n");
    }
    else if (bmi >= 25 & & bmi < 30) {
        printf("Category: Overweight\n");
    }
}
```

Teacher's Signature: _____

Experiment No. _____

Name: _____

PAGE NO.: _____

DATE: / /

```
else if (bmi >= 30 & & bmi < 40) {  
    printf ("category: obese \n");  
}  
else {  
    printf ("category: Morbidly obese. \n");  
}  
return 0;  
}
```

Teacher's Signature: _____

Terminal :- gcc (file name . c)

Enter weight (in Kg) : 40
Enter height (in m) : 1.6
Your BMI is : - 13.84
Category Starvation

a tab

Enter weight (in Kg) : - 50

Enter height (in m) : 1.70

Your BMI is 17.30

Category : Underweight ("Underweight")
(":(Custom a) weight ratio")

a tab

Enter weight (Kg) : -(Weight > 30)

Enter height (in m) : -(1.62) Height

Entered BMI is : 22.49 ; and over")

a tab

Enter ("weight", (in Kg) : 65")

}(2.5) > and & 2 21 = < and) { if else

{"caloriecount : (calories") } { else

}(2.8) > and & 2 21 = < and) { if else

{"caloriesview person : (proteins") } { else

}(0.8 > and & 2 21 = < and) { if else

{"caloriesview person : (proteins") } { else

Experiment No. 3.1 (4): Name:

PAGE NO.:

DATE: / /

- 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 using C-Programming.

- Codes:-

```
#include <stdio.h>
int main() {
    int l1, b1, l2, b2, l3, b3;
```

```
    int p1, p2, p3, max;
```

```
    printf("Enter length and breadth of rectangle 1: ");
    scanf("%d %d", &l1, &b1);
```

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

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

$$p_1 = 2 * (l_1 + b_1);$$

$$p_2 = 2 * (l_2 + b_2);$$

$$p_3 = 2 * (l_3 + b_3);$$

$$\max = (p_1 > p_2) ? (p_1 > p_3) ? p_1 : p_3 : ((p_2 > p_3) ? p_2 : p_3);$$

Teacher's Signature:

Terminal (Output) :- gcc [file name]
a - tab / sizeof : (int) *)

Enter length and breadth of rectangle 1 : 12 10

Enter length and breadth of rectangle 2 : 7 9

Enter length and breadth of rectangle 3 : 15 19

Perimeter of Rectangle 1 = 44

Perimeter of Rectangle 2 = 32

Perimeter of Rectangle 3 = 68

The rectangle with the highest perimeter has perim
= 68.

Experiment No. _____

Name: _____

PAGE NO.: _____

DATE: / /

```
printf("\n Perimeter of Rectangle 1 = %d", p1);  
printf("\n Perimeter of Rectangle 2 = %d", p2);  
printf("\n Perimeter of Rectangle 3 = %d", p3);  
printf("\n The rectangle with the highest perimeter  
has perimeter = %d \n", max);
```

```
return 0;
```

```
}
```

Teacher's Signature: _____

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

- Code:-

```
#include <stdio.h>
int main() {
    int year;
    printf("Enter a year:");
    scanf("%d", &year);
    int base_day = 1;
    int total_days = 0;
    for (int i=1; i<year; i++) {
        if ((i%400 == 0) || (i%4 == 0 && i%100 != 0))
            total_days += 366;
        else
            total_days += 365;
    }
}
```

printf("1st January of year %d is", 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;

Terminal (Output) :- $b = 1$ is passed to statement 3.

RJ> Enter a year : 12
1st January of 12 is Sunday

Enter a year : 2
1st January of 2 is Tuesday

Enter a year : 6

1st January of 6 is Sunday

Experiment No. _____

Name: _____

PAGE NO.: _____

DATE: / /

Case 4: pointf ("Thursday \n"); break;

Case 5: pointf (" Friday \n"); break;

Case 6: pointf (" Saturday \n"); break;

}

return 0;

y

Teacher's Signature: _____

Exp-1

- Aim:- WAP to enter numbers till the user wants. At the end, it should display the count of positive, negative and zeroes entered. Using c- Programming

- Code:-

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

```
int main () {
```

```
    int num;
```

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

```
    char choice;
```

```
    do {
```

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

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

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

```
            positive++;
```

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

```
            negative++;
```

```
        else
```

```
            zero++;
```

```
    Point ("Enter another number(y/n): ");
```

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

```
} while (choice == 'y' || choice == 'Y');
```

```
printf ("\n Count of positive numbers: %d \n", positive);
```

```
printf ("\n Count of negative numbers: %d \n", negative);
```

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

Teacher's Signature: _____

Terminal :- (Output)

Enter a number: 6

Enter another number (y/n): y

Enter a number: -3

Enter another number (y/n): y

Enter a number: 0

Enter another number (y/n): y

Enter a number: 12

Enter another number (y/n): n

Count of positive numbers: 2

Count of negative numbers: 1

Count of zeroes: 1

Experiment No. _____

Name: _____

PAGE NO.: _____

DATE: / /

return 0;

}

Teacher's Signature: _____

- Aims :- WAP to generate the following set of output
- ⑥ 1
2 3
4 5 6
- using C Programming.

Codes:-

```
#include <stdio.h>
int main() {
    int num = 1;
    int rows = 3;
    for (int i = 1; i <= rows; i++) {
        for (int s = 1; s <= rows - i; s++) {
            printf(" ");
        }
        for (int j = 1; j <= i; j++) {
            printf("%d", num);
            num++;
        }
        printf("\n");
    }
    return 0;
}
```

Terminal :- (Output)

Enter a number: 5

multiplication table of 5:

$$5 * 1 = 5$$

$$5 * 2 = 10$$

$$5 * 3 = 15$$

$$5 * 4 = 20$$

$$5 * 5 = 25$$

$$5 * 6 = 30$$

$$5 * 7 = 35$$

$$5 * 8 = 40$$

$$5 * 9 = 45$$

$$5 * 10 = 50$$

Aim:- (b) 1

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

C- Programming

Codes:-

```
#include <stdio.h>
int main() {
    int n=5, space;
    for (int i=0; i<n; i++) {
        for (space=1; space<=n-i; space++)
            {
                printf(" ");
            }
        int num=1;
        for (int j=0; j<=i; j++) {
            printf("%d", num);
            num = num*(i-j)/(j+1);
        }
        printf("\n");
    }
    return 0;
}
```

Terminal :- Output

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

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.

Code:-

```
#include <stdio.h>
int main() {
    int limit = 10000;
    for (int num = 1; num <= limit; num++) {
        int count = 0;
        for (int a = 1; a*a*a < num; a++) {
            for (int b = a; a*a*a + b*b*b <= num; b++) {
                if (a*a*a + b*b*b == num) {
                    count++;
                }
            }
        }
        if (count >= 2) {
            printf("%d\n", num);
        }
    }
    return 0;
}
```

Terminal (Output):-

1729

4104

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.

Codes :- #include <Stdio.h>

```
int main() {  
    double population = 100000;  
    double rate = 0.1;  
    for (int year = 1; year <= 10; year++) {  
        population = population + (population * rate);  
        printf ("Year %d : %.0f \n", year, population);  
    }  
    return 0;  
}
```

Terminal (Output) :-

Year 1 : 110000

Year 2 : 121000

Year 3 : 133100

Year 4 : 146410

Year 5 : 161051

Year 6 : 177156

Year 7 : 194872

Year 8 : 214359

Year 9 : 239374

Year 10 : 289374

Exp - 1

- Aim:- Declare a Global Variable outside all functions and use it inside various functions to understand its accessibility. Using C Programming.

Codes:- # include <stdio.h>

```
int count = 0;
```

```
void increment () {
```

```
    count++;
```

```
    printf ("Inside increment (): count = %d \n", count);
```

```
}
```

```
void display () {
```

```
    printf ("Inside display (): count = %d \n", count);
```

```
}
```

```
int main () {
```

```
    printf ("Initially: count = %d \n", count);
```

```
    increment ();
```

```
    increment ();
```

```
    display ();
```

```
    return 0;
```

```
}
```

- Terminal (Output) :-

Initially : count = 0.

Inside increment(): count = 1

Inside increment(): count = 2

Inside display(): count = 2

Exp-2

- Aim:- Declare a local variable outside all functions and use it inside various functions. Using C-Programming.

Codes:- #include <stdio.h>

```
int globalVar = 100;
```

```
Void showLocal () {
```

```
    int localVar = 50;
```

```
    printf ("Inside showLocal (): localVar = %d \n", localVar);
```

```
}
```

```
int main () {
```

```
    showLocal ();
```

```
    printf ("In main (): globalVar = %d \n", globalVar);
```

```
    return 0;
```

```
}
```

Exp-3

- Aim:- Declare variables within different code blocks and test their accessibility within the outside those blocks.

Codes:- #include <stdio.h>

```
int main () {
```

```
    int a = 20;
```

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

```
}
```

```
    int b = 10;
```

Teacher's Signature:

Terminal (Output) :-

Inside showlocal (): localVar = 50

Inside showlocal (): globalVar = 100

In main (): globalVar = 100

Terminal (Output) :-

a inside main: 20

a inside first block: 20

b inside first block: 10

a inside second block: 20

c inside second block: 40

```
printf("a inside first block: %d \n", a);
printf ("b inside first block : %d \n", b);
}
{
int c = 40;
printf("a inside second block: %d \n", a);
printf ("c inside second block : %d \n", c);
}
return 0;
}
```

Exp - 4

- Aim:- Declare a static local Variable inside a function.
Observe how its value persists across function calls.

Codes:-

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

```
int main() {
    demo Function();
    demo Function();
}
```

Terminal output) :-

count = 1

count = 2

count = 3

count = 4

Experiment No.

Name :

PAGE NO.:

DATE : / /

```
demo function();  
demo function();  
return 0;  
}
```

Teacher's Signature:

Exp - 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 n, i;
    printf("Enter the number of elements: ");
    scanf("%d", &n);
    int arr[n];
    printf("Enter %d integers: \n", n);
    for (i=0; i<n; i++) {
        scanf("%d", &arr[i]);
    }
    int largest, second_largest;
    if (n<2) {
        printf("At least two numbers are required.\n");
    } else {
        if (arr[0] > arr[1]) {
            largest = arr[0];
            second_largest = arr[1];
        } else {
            largest = arr[1];
            second_largest = arr[0];
        }
        for (i=2; i<n; i++) {
            if (arr[i] > largest) {
                second_largest = largest;
                largest = arr[i];
            } else if (arr[i] > second_largest) {
                second_largest = arr[i];
            }
        }
    }
}
```

Teacher's Signature:

```
for (i=2; i<n; i++) {  
    if (arr[i] > largest) {  
        second - largest = largest;  
        largest = arr[i];  
    } else if (arr[i] > second - largest && arr[i] != largest)  
    {  
        second - largest = arr[i];  
    }  
}  
  
printf ("The second largest number is: %d \n", second -  
largest);  
return 0;  
}
```

• Terminal :-

→ Enter the number of element: 5

Enter 5 integers:

10 25 30 15 20

The second largest number is 25

Teacher's Signature:

Aim:- WAP to read a list of integers and store it in a single dimensional array.

```
. code:- #include < stdio.h >
int main () {
    int n, i;
    printf (" Enter the number of elements: ");
    scanf ("%d", &n);
    int arr[n];
    printf (" Enter %d integers: \n ", n);
    for (i=0, i<n, i++) {
        scanf ("%d", &arr[i]);
    }
    int positive=0, negative=0, even=0, odd=0;
    for (i=0, i<n, i++) {
        if (arr[i]>0)
            positive++;
        else if (arr[i]<0)
            negative++;
        if (arr[i] % 2 == 0)
            even++;
        else
            odd++;
    }
}
```

Teacher's Signature:

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

return 0;

3

Exp-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 in a list of integers.

Codes:- #include <stdio.h>

```
int main()
```

```
int n, i, num, count = 0;
```

```
printf("Enter the number of elements:");
```

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

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

```
printf("Enter %d integer: \n", n);
```

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

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

```
printf("Enter the number to find its frequency:");
```

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

Teacher's Signature:

• Terminal:-

Enter the number of elements: 6

Enter 6 integers:

10 -5 0 7 -2 4

Positive numbers: 3

Negative numbers: 2

Even numbers: 3

Odd numbers: 3

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

}

```
printf ("The frequency of %d is: %d\n", num,  
       count);
```

```
return 0;
```

}

Experiments

Teacher's Signature:

• Terminal:-

Enter the number of elements : 6

Enter 6 integers :-

2 4 2 5 6 2

Enter the number to find its frequency :-

The frequency of 2 is : 3

Ewp - 1

Aim:- Declare different types of pointer (int, float, char) and initialize them with the address of variables. Point the value of both the pointers and the variables they point to.

Codes:- #include <stdio.h>

```
int main () {
    int a = 10;
    float b = 20.5;
    char c = 'A';
    int *p1 = &a;
    float *p2 = &b;
    char *p3 = &c;
    printf ("Value of a = %d \n", a);
    printf ("Value of b = %.2f \n", b);
    printf ("Value of c = %c \n", c);
    printf ("\n Address stored in p1 = %p \n", p1);
    printf ("Address stored in p2 = %p \n", p2);
    printf ("Address stored in p3 = %p \n", p3);
    printf ("\n Value pointed by p1 = %d \n", *p1);
    printf ("Value pointed by p2 = %.2f \n", *p2);
    printf ("Value pointed by p3 = %c \n", *p3);
    return 0 ; }
```

Teacher's Signature:

Terminal :- Value of $a = 10$
Value of $b = 20.5$
Value of $c = A$
Address stored in $p_1 = 0061\text{FF}10$
Address stored in $p_2 = 0061\text{FF}0C$
Address stored in $p_3 = 0061\text{FF}0B$
Value pointed by $p_1 = 10$
Value pointed by $p_2 = 20.50$
Value pointed by $p_3 = A$

Exp-2

Aim:- Perform pointer arithmetic on pointers of different data types. Observe how the memory address changes and the effect on data access.

Codes:- #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 ("Initial pointer value and data : \n");
```

```
    printf ("p1 = %p, *p1 = %d \n", p1, *p1);
```

```
    printf ("p2 = %p, *p2 = %.1f \n", p2, *p2);
```

```
    printf ("p3 = %p, *p3 = %c \n", p3, *p3);
```

```
    p1 ++;
```

```
    p2 ++;
```

```
    p3 ++;
```

```
    printf ("\n After Increment (p1++, p2++, p3++) : \n");
```

```
    printf ("p1 = %p, *p1 = %d \n", p1, *p1);
```

```
    printf ("p2 = %p, *p2 = %.1f \n", p2, *p2);
```

```
    printf ("p3 = %p, *p3 = %c \n", p3, *p3);
```

Teacher's Signature:

Terminal:- Initially pointer value and Data:

$p_1 = 0061\text{FF08}$, $*p_1 = 10$

$p_2 = 0061\text{FFFC}$, $*p_2 = 1.1$

$p_3 = 0061\text{FEF9}$, $*p_3 = A$

After Increment (p_1++ , p_2++ , p_3++):

$p_1 = 0061\text{FF0C}$, $*p_1 = 20$

$p_2 = 0061\text{FF00}$, $*p_2 = 2.2$

$p_3 = 0061\text{FEFA}$, $*p_3 = A$

After Decrement (p_1-- , p_2-- , p_3--):

$p_1 = 0061\text{FF08}$, $*p_1 = 10$

$p_2 = 0061\text{FFFC}$, $*p_2 = 1.1$

$p_3 = 0061\text{FEF9}$, $*p_3 = A$

Experiment No. 8

Name: Pointers

PAGE NO.

DATE

Ans

p1 = -;

p2 = -;

p3 = -;

printf ("\\n After Decrement (p1--, p2--, p3--): \\n"),
printf ("p1 = %p, *p1 = %d \\n", p1, *p1);
printf ("p2 = %p, *p2 = %d f \\n", p2, *p2);
printf ("p3 = %p, *p3 = %c \\n", p3, *p3);

return 0;

}

Exp - 3

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

Codes:- #include <stdio.h>

```
Void modify Values (int *x, float *y, char *z);  
int main () {  
    int a = 10;  
    float b = 5.5;  
    char c = 'A';  
    printf ("Before Modification : \n");  
    printf ("a = %d, b = %.1f, c = %c\n", a, b, c);  
    modify Values (&a, &b, &c);  
    printf ("\n After Modification : \n");  
    printf ("a = %d b = %.1f, c = %c\n", a, b, c);  
    return 0;  
}
```

```
Void modify Values (int *x, float *y, char *z) {
```

```
    *x = *x + 5;
```

```
    *y = *y * 2;
```

```
    *z = 'Z';
```

```
}
```

Teacher's Signature:

Terminal :- Before Modification:

$a = 10, b = 5.5, c = A$

After Modification:

$a = 15, b = 51.0, c = z$

Experiment No. 9

Name: File Handling in C

PAGE NO.:

DATE: / /

Aim:- write a program to create a new file and write text into it.

Exp-1

Codes:- #include <stdio.h>

int main () {

FILE *fp;

fp = fopen("Output.txt", "w");

if (fp == NULL) {

printf("Error! Could not create file.\n");

return 1;

}

fputs ("This is a simple file handling example in C.\n",

fp);

fclose(fp);

printf("file created and text written successfully.\n");

return 0;

}

Exp-2

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

Codes:- #include <stdio.h>

int main () {

FILE *fp;

char ch;

fp = fopen("Output.txt", "r");

if (fp == NULL) {

Teacher's Signature:

Terminal Output :-

file created and text written successfully.

output.txt => This is a simple file handling example in

```
pointf ("Error! File does not exist.\n");
return 1;
```

}

```
while ((ch = fgetc (fp)) != EOF) {
    putchar(ch);
```

}

```
fclose(fp);
```

```
return 0;
```

}

Exp-3

Aim:

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

```
int main () {
```

```
FILE *fp;
```

```
char line[200];
```

```
fp = fopen ("output.txt", "w");
```

```
if (fp == NULL) {
```

```
pointf ("Error! File not found.\n");
```

```
return 1;
```

}

```
while (fgets (line, sizeof(line), fp) != NULL) {
```

```
pointf ("%s", line);
```

}

```
fclose (fp);
```

```
return 0;
```

}

Teacher's Signature: _____

File Handling :-

File handling is a process of writing or reading data from a file.

File handling is done by opening a file.

File handling is done by closing a file.

Terminal (Output):-

This is a simple file handling example in C.

Experiment No. 16

Name: Bitwise operator

PAGE NO.:

DATE: / /

~~Exp-1~~

Aim:- Write a program to apply bitwise OR, AND and NOT operators on bit level.

```
code → #include <stdio.h>
int main() {
    unsigned int a, b;
    printf("Enter first number: ");
    scanf("%u", &b);
    printf("Enter second number: ");
    scanf("%u", &a);
    printf("\nBitwise AND (a&b) = %u", a & b);
    printf("\nBitwise OR (a|b) = %u", a | b);
    printf("\nBitwise NOT (~a) = %u", ~a);
    printf("\nBitwise NOT (~b) = %u", ~b);
    return 0;
}
```

Exp-2Aim:-

```
code → #include <stdio.h>
int main() {
    int num, ls, rs;
    printf("Enter a number: ");
    scanf("%d", &num);
    ls = num < 1;
    rs = num >> 1;
```

Teacher's Signature:

Terminal (Output) :-

Enter first number:- 5

Enter second number:- 6

Bitwise AND ($a \& b$) = 4

Bitwise OR ($a | b$) = 7

Bitwise NOT ($\sim a$) = 4294967290

Bitwise NOT ($\sim b$) = 4294967289

```

printf("Original number = %d", num);
printf("After left shift (num << 1) = %d", ls);
printf("After right shift (num >> 1) = %d", rs);
return 0;
}

```

3

Exp-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() {
    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;
}

```

Teacher's Signature:

Terminal :- (Output)

Enter a number :: 3

Original number = 3

After left shift ($num \ll 1$) = 6

After right shift ($num \gg 1$) = 1

Expt. No. 12 / Processors and Directives inc

Date _____

Page No. _____

Exp-1

Aim:- write a program to define some constant variables in preprocessor.

Codes:-

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

```
int main()
{
    printf("Value of PI : %.2f\n", PI);
    printf("Maximum value : %d\n", MAX);
    printf("Minimum value : %d\n", MIN);
    return 0;
}
```

y

Exp-2

Aim:- Write a program to define a function in directives.

Codes:-

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

```
#define SQUARE(n) ((n)*(n))
```

```
int main()
```

```
int num = 5;
```

```
printf("Number: %d\n", num);
```

```
printf("Square: %d\n", SQUARE(num));
```

```
return 0;
```

y

Terminal (Output) :-

Value of PI : 3.14

Maximum Value : 100

Minimum Value : 1

Terminal (Output) :-

Number :- 5

Square :- 25

Aim:- write a program to define multiple macros to perform arithmetic functions

Exp-1

codes:-

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

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

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

```
#define MUL(a,b) ((a)*(b))
```

```
#define DIV(a,b) ((b) != 0 ? ((a)/(b)) : 0)
```

```
int main()
```

```
int n=20, y = 10;
```

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

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

```
printf("Multiplication = %d \n", MUL(n,y));
```

```
printf("Division = %d \n", DIV(n,y));
```

```
return 0;
```

```
}
```

Terminal Output):-

Addition = 30

Subtraction = 10

Multiplication = 200

Division = 2

Exps - I

Bim:- write a program to create a static library for performing arithmetic functions

Coder:-

```
#ifndef ARITH_H  
#define 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);
```

end if

```
#include "arith.h"
```

```
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;  
}
```

}

```
# include < stdio.h >
# include "arith.h"
int main() {
    int a = 10, b = 5;
    printf("Addition: %d \n", add(a, b));
    printf("Subtraction: %d \n", sub(a, b));
    printf("Multiplication: %d \n", mul(a, b));
    printf("Division: %d \n", div(a, b));
    return 0;
}
```

Terminal :- (Output)

Addition : 15

Subtraction : 5

Multiplication: 50

Division: 2