

LAB-2

Write the basic structure of C.

```
/*Program to find the area & circumference of a circle*/  
#include <stdio.h>  
#define PI 3.14  
float r, a, c;  
float area();  
float circumference();  
int main()  
{  
    printf("Radius = ");  
    scanf("%f", &r);  
    a = area();  
    c = circumference();  
    printf("Area = %.f", a);  
    printf("\nCircumference = %.f", c);  
    return 0;  
}
```

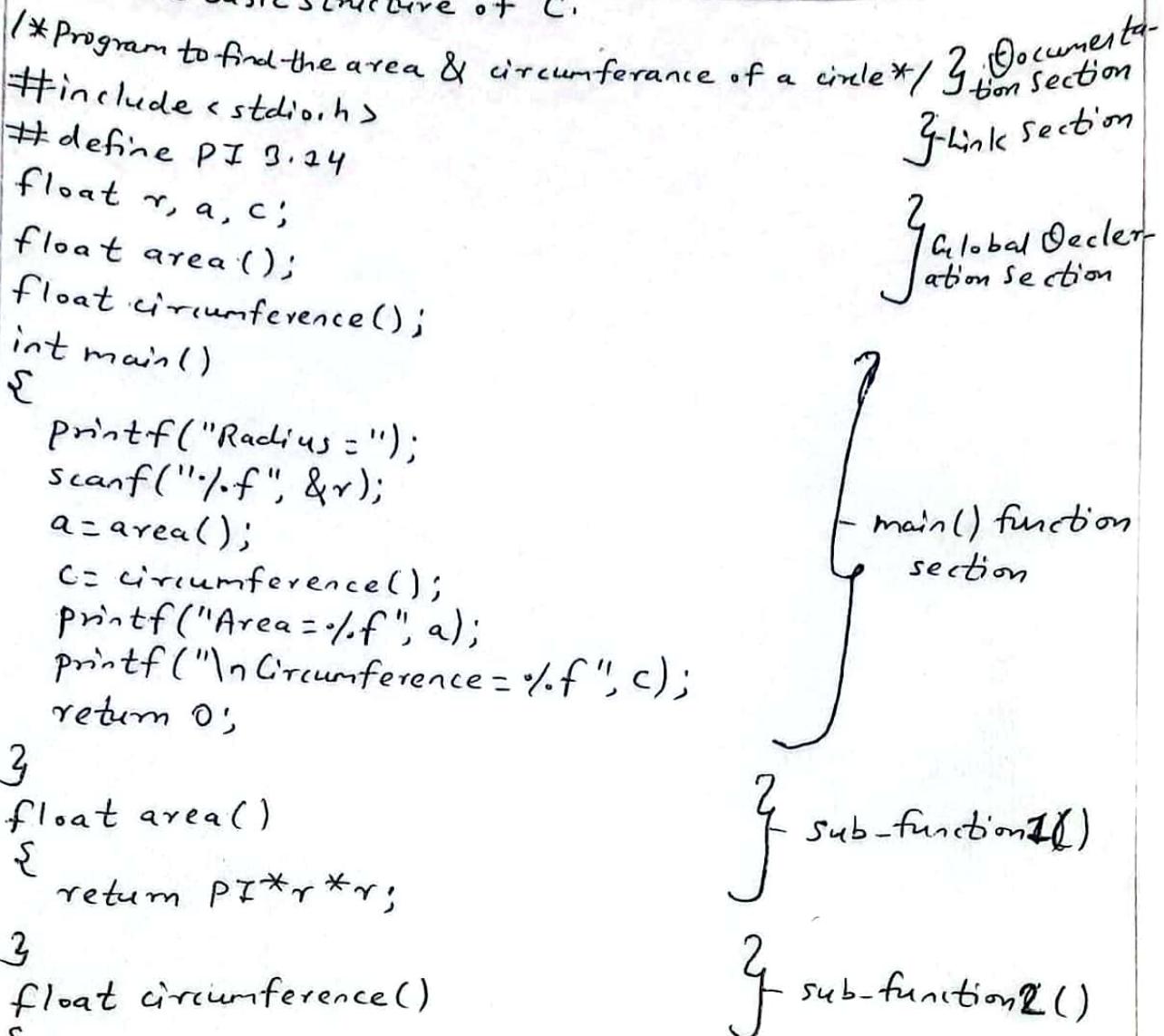
```
float area()  
{  
    return PI * r * r;
```

```
float circumference()  
{  
    return 2 * PI * r;
```

}

Output:

Radius = 7
Area = 153.86001
Circumference = 43.959999



LAB - 2

// WAP to display your details in five different lines.

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

```
int main() {
```

```
    printf("My name is Maish Dhusal.\n");
```

```
    printf("I follow Hindu Religion.\n");
```

```
    printf("I have completed +2 from Gyan Jyoti.\n");
```

```
    printf("I love reading books.\n");
```

```
    printf("I am a hardworking person.\n");
```

```
    return 0;
```

3

Output:

My name is Maish Dhusal.

I follow Hindu Religion.

I have completed +2 from Gyan Jyoti.

I love reading books.

I am a hardworking person.

```
//WAP to add, subtract, multiply and divide two numbers.
#include <stdio.h>
int main() {
    int num1, num2;
    int sum, sub, mult;
    float div;
    printf("Enter two numbers:");
    scanf("%d %d", &num1, &num2);
    sum = num1 + num2;
    sub = num1 - num2;
    mult = num1 * num2;
    div = (float)num1 / num2;
    printf("Sum of %d & %d is %d\n", sum);
    printf("Difference: %d\n", sub);
    printf("Multiply: %d\n", mult);
    printf("Division: %.2f\n", div);
}

```

3

Output

Enter two numbers: 4 2

Sum of 4 & 2 is 6.

Difference: 2.

Multiply: 8

Division: 2.0

LAB-4

```
// WAP to find the remainder of a number after dividing it by 2.  
#include <stdio.h>  
int main()  
{  
    int n;  
    printf("Enter a no: ");  
    scanf("%d", &n);  
    int remainder = n%2;  
    printf("The remainder of a number %d is %d", n, remainder);  
    return 0;  
}
```

Output

Enter a no: 27

The remainder of a number 27 is 1.

LAB - 5

```
// WAP to find Simple Interest.  
#include <stdio.h>  
int main()  
{  
    int p, t;  
    float r, si;  
    printf("Enter the value of p, t, r: ");  
    scanf("%d %d %f", &p, &t, &r);  
    si = (p * t * r) / 100;  
    printf("Simple Interest = %f", si);  
    return 0;  
}
```

Output

Enter the value of p, t, r: 2000 2 22.5
Simple Interest = 862.500000

LAB-6

```
// WAP to find area and perimeter of a rectangle  
#include <stdio.h>  
int main()  
{  
    float radius, area, perimeter;  
    printf("Enter the radius of a circle: ");  
    scanf("%f", &radius);  
    area = 3.14 * radius * radius;  
    perimeter = 2 * 3.14 * radius;  
    printf("Area = %f\n", area);  
    printf("Perimeter = %f\n", perimeter);  
    return 0;  
}
```

Output

Enter the radius of a circle: 2.3
Area of the circle = 16.960059
Perimeter = 14.44002

LAD-7

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

Output

```
Enter temperature in Celsius: 32.5  
Temperature in Fahrenheit: 90.56
```

LAB-8

```
// WAP that finds the larger among two integers using conditional operator.  
#include <stdio.h>  
int main() {  
    int a, b;  
    printf("Enter two numbers: ");  
    scanf("%d %d", &a, &b);  
    a > b ? printf("%d is larger", a) : printf("%d is larger", b);  
    return 0;  
}
```

Output

Enter two numbers: 200 200

200 is larger.

LAB-9

```
1//WAP that demonstrates the use of bitwise logical operators  
#include <stdio.h>  
int main()  
{  
    int num1 = 505, num2 = 225;  
    int AND, OR, XOR;  
    AND = num1 & num2;  
    OR = num1 | num2;  
    XOR = num1 ^ num2;  
    printf("AND = %.d\n", AND);  
    printf("OR = %.d\n", OR);  
    printf("XOR = %.d\n", XOR);  
    return 0;  
}
```

3

Output

AND = 209

OR = 522

XOR = 302

LAB-20

// WAP that demonstrates the use of sizeof() operator.

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

```
int main()
```

```
{
```

```
    int num;
```

```
    printf("Size of int: %.d bytes\n", sizeof(num));
```

```
    printf("Size of char: %.d bytes\n", sizeof(char));
```

```
    printf("Size of float: %.d bytes\n", sizeof(float));
```

```
    printf("Size of double: %.d bytes\n", sizeof(double));
```

```
    printf("double constant occupies %d bytes\n", sizeof(3.14));
```

```
    return 0;
```

```
}
```

Output

Size of int: 4 bytes

Size of char: 2 bytes

Size of float: 4 bytes

Size of double: 8 bytes

double constant occupies 8 bytes

```
/* WAP to read two characters from keyboard; one using getch() &
another using getche() function & display the character using putch() function
*/
#include <stdio.h>
#include <conio.h>
int main()
{
    char a, b;
    printf("Enter a character: ");
    a = getch();
    printf("\n Enter another character: ");
    b = getche();
    printf("\n Your character is: ");
    putch(a);
    printf("\n Your character is: ");
    putch(b);
    return 0;
}
```

3

Output

Enter a character:
Enter another character: b
Your character is: a
Your character is: b

```
/* WAP to read a string with multiple words (i.e. string with space)
using gets() function & display on the screen using puts() function. */
#include <stdio.h>
#include <conio.h>
int main()
{
    char name[20];
    printf("Enter your name: ");
    gets(name);
    printf("Your name is: ");
    puts(name);
    return 0;
}
```

Output

Enter your name: Manish Bhusal

Your name is: Manish Bhusal

LAB - 23

```
//WAP to determine whether the entered no. is even or odd.  
#include <stdio.h>  
int main()  
{  
    int num;  
    printf("Enter a number: ");  
    scanf("%d", &num);  
    if (num % 2 == 0)  
    {  
        printf("The no. %d is even", num);  
    } else {  
        printf("The no. %d is odd", num);  
    }  
    return 0;  
}
```

Output

Enter a number: 23

The number 23 is odd.



LAB-24

```
// WAP to determine the roots of quadratic eq^n ax2+bx+c = 0.  
#include <stdio.h>  
#include <math.h>  
int main()  
{  
    float a, b, c, d, root1, root2, real, img;  
    printf("Enter the coefficients a, b and c: ");  
    scanf("%f %f %f", &a, &b, &c);  
    d = b*b - 4*a*c;  
    if (d < 0)  
    {  
        printf("Imaginary Roots: ");  
        d = sqrt(fabs(d));  
        real = -b / (2*a);  
        img = d / (2*a);  
        printf("\nRoot1 = %.2f + i %.2f", real, img);  
        printf("\nRoot2 = %.2f - i %.2f", real, img);  
    }  
    else  
    {  
        printf("Real Roots");  
        d = sqrt(d);  
        root1 = (-b+d) / (2*a);  
        root2 = (-b-d) / (2*a);  
        printf("\nRoot1 = %.2f \t Root2 = %.2f", root1, root2);  
    }  
    return 0;  
}
```

Output

Enter the coefficient a, b and c; 2 8 3

Real Roots

Root1 = -0.42 Root2 = -3.58

LAB-15

```
/* WAP to find the largest among three numbers using logical operator  
& else if statement */  
#include <stdio.h>  
int main()  
{  
    int n1, n2, n3;  
    printf("Enter three numbers: ");  
    scanf("%d %d %d", &n1, &n2, &n3);  
    if(n1 > n2 && n1 > n3)  
    {  
        printf("%d is the largest number", n1);  
    } else if(n2 > n1 && n2 > n3)  
    {  
        printf("%d is the largest no.", n2);  
    } else  
    {  
        printf("%d is the largest no.", n3);  
    }  
    return 0;  
}
```

Output

Enter three numbers: 23 40 55

55 is the largest no.

LAB-16

//WAP that performs the arithmetic operations using switch statement.

```
#include <stdio.h>
int main()
{
    int a, b;
    char op;
    printf("Enter the value of a and b: ");
    scanf("%d %d", &a, &b);
    printf("Enter the operator: ");
    scanf(" %c", &op);
    switch (op)
    {
        case '+':
            printf("\n The sum of %d & %d is %d", a, b, a+b);
            break;
        case '-':
            printf("\n The difference of %d & %d is %d", a, b, a-b);
            break;
        case '*':
            printf("\n Multiplication: %d", a*b);
            break;
        case '/':
            printf("\n Division: %d", a/b);
            break;
        default:
            printf("\n Invalid Operator");
    }
    return 0;
}
```

3

Output

Enter the value of a and b: 3 6

Enter the operator: +

The sum of 3 & 6 is 9. ✓

LAB-17

```
/*WAP that prints the day on the basis of the number entered by the user. (Repetitive / Looping Statement) */  
#include <stdio.h>  
int main() {  
    int dayNumber;  
    do {  
        printf("Enter a no. (1-7): ");  
        scanf("%d", &dayNumber);  
        switch (dayNumber) {  
            case 1:  
                printf("Sunday\n");  
                break;  
            case 2:  
                printf("Monday\n");  
                break;  
            case 3:  
                printf("Tuesday\n");  
                break;  
            case 4:  
                printf("Wednesday\n");  
                break;  
            case 5:  
                printf("Thursday\n");  
                break;  
            case 6:  
                printf("Friday\n");  
                break;  
            case 7:  
                printf("Saturday\n");  
                break;  
            default:  
                printf("Invalid input! Please enter a week no. betn 1-7.\n");  
        }  
    } while (dayNumber < 1 || dayNumber > 7);  
    return 0;  
}
```

Output

Enter a no. (1-7): 55

Invalid input! Please enter a week no. betn 1-7.

Enter a no. (1-7): 3

Tuesday

WAP to sum all the integers from 1 to 50 using while loop.

```
#include <stdio.h>
int main()
{
    int sum = 0, i = 1;
    while (i <= 50)
    {
        sum += i;
        i++;
    }
    printf("Sum is %d", sum);
    return 0;
}
```

Output

Sum is 1275



LAB-19

// WAP to check whether the entered no. is Armstrong Number.

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

```
int main()
```

```
{ int num, digit, sum = 0, originalNum;  
printf("Enter a number: ");  
scanf("%d", &num);  
originalNum = num;  
while (num > 0) {
```

```
    digit = num % 10;  
    sum += digit * digit * digit;  
    num /= 10;
```

```
}
```

```
if (originalNum == sum)
```

```
{ printf("The no. is an Armstrong no.");
```

```
} else
```

```
{ printf("The no. is not an Armstrong no.");
```

```
}
```

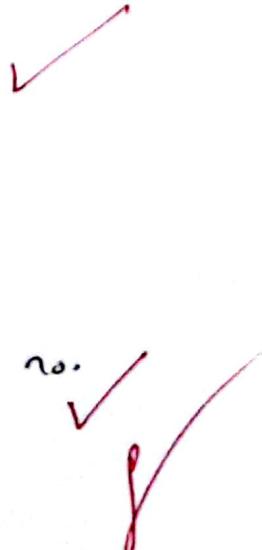
```
return 0;
```

```
}
```

Output

Enter a number: 153

The no. is an Armstrong no.



LAB - 20

//WAP to determine whether the no. entered is Palindrome or not.

```
#include <stdio.h>
int main()
{
    int num, rev = 0, rem, originalNum;
    printf("Enter a number: ");
    scanf("%d", &num);
    originalNum = num;
    while (num != 0)
    {
        rem = num % 10;
        rev = rev * 10 + rem;
        num /= 10;
    }
    if (originalNum == rev)
    {
        printf("The no. is a palindrome");
    }
    else
    {
        printf("The no. is not a palindrome");
    }
    return 0;
}
```

Output

Enter a number: 54345

The no. is a palindrome

LAB-21

//WAP to print out all numbers from 1 to 20 using do-while loop.

```
#include<stdio.h>
int main()
{
    int x=1;
    do {
        printf("./d\n", x);
        x++;
    } while (x <= 20);
    return 0;
}
```

Output

1
2
3
4
5
6
7
8
9
20



LAB - 22

// WAP to display the multiplication table of the entered no.

```
#include <stdio.h>
int main()
{
    int n;
    printf("Enter a number: ");
    scanf("./d", &n);
    for (int i = 1; i <= 20; i++)
    {
        printf("./d * ./d = ./d\n", n, i, n * i);
    }
    return 0;
}
```

Output

Enter a number: 2

2 * 1 = 2
2 * 2 = 4
2 * 3 = 6
2 * 4 = 8
2 * 5 = 10
2 * 6 = 12
2 * 7 = 14
2 * 8 = 16
2 * 9 = 18
2 * 20 = 20

✓ ✓

LAB-23

// WAP to illustrate the use of goto statement.

```
#include <stdio.h>
int main()
{
    int sum = 0;
    for (int i = 0; i <= 20; i++) {
        sum += i;
        if (i == 5) {
            goto addition;
        }
    }
    addition:
    printf("%d", sum);
    return 0;
}
```

Output

25



LAB - 24

// WAP to display all prime numbers less than 200.

```
#include <stdio.h>
int main()
{
    int prime;
    printf("Prime numbers from 2 to 200 are:\n");
    for (int i = 2; i < 200; i++)
    {
        prime = 1;
        for (int j = 2; j < i; j++)
        {
            if (i % j == 0)
            {
                prime = 0;
                break;
            }
        }
        if (prime)
        {
            printf("./d\n", i);
        }
    }
    return 0;
}
```

Output

Prime numbers from 2 to 200 are:

2 3 5 7 11 13 17 19 23 29 31 37
41 43 47 53 59 61 67 71 73 79 83
89 97

LAB - 25

// WAP to display all the leap years starting from 1900 to 2000.

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

```
int main()
```

```
{
```

```
    int year;
```

```
    printf("Leap years starting from 1900 to 2000: \n");
```

```
    for (year = 1900; year <= 2000; year++)
```

```
{
```

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

```
    {
```



```
        printf("%d\t", year);
```

```
}
```

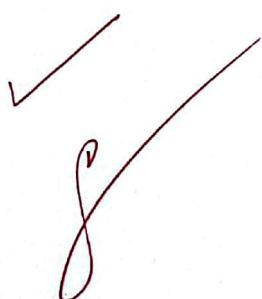
```
}
```

```
    return 0;
```

```
}
```

Output

1904 1908 1912 1916 1920 1924 1928 1932 1936
1940 1944 1948 1952 1956 1960 1964 1968
1972 1976 1980 1984 1988 1992 1996 2000.



LAB-26

/* WAP to print the following pattern using for loop.

\$ \$ \$ \$ \$
\$ \$ \$ \$
\$ \$ \$
\$ \$
\$ */

```
#include <stdio.h>
int main()
{
    for(int i = 5; i >= 2; i--) {
        for (int j = 2; j <= i; j++) {
            printf("$ ");
        }
        printf("\n");
    }
    return 0;
}
```

Output

\$ \$ \$ \$ \$
\$ \$ \$ \$
\$ \$ \$
\$ \$
\$

LAB - 27

/* WAP to print the following pattern using for loop.

```
1  
2 2  
3 3 3  
4 4 4 4  
5 5 5 5 5  
*/
```

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

Output

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

LAB- 28

```
/*WAP to read 20 integer numbers from the user & display the entered no. in  
the screen . */  
#include <stdio.h>  
int main( ) {  
    int numbers[20];  
    printf("Enter 20 integer numbers: \n");  
    for(int i=0; i<20; i++) {  
        scanf("./d", &numbers[i]);  
    }  
    printf("\nEnterd numbers: \n");  
    for(int i=0; i<20; i++) {  
        printf("./d ", numbers[i]);  
    }  
    return 0;  
}
```

Output

Enter 20 integer numbers:

1
2
3
4
5
6
7
8
9
10

Entered numbers:

1 2 3 4 5 6 7 8 9 10

LAB-29

//WAP to find the largest & smallest element in the array.

```
#include <stdio.h>
int main() {
    int arr[200], n;
    printf("Enter no. of elements in the array: ");
    scanf("%d", &n);
    printf("Enter the elements of the array:\n");
    for(int i=0; i<n; i++) {
        scanf("%d", &arr[i]);
    }
}
```

```
int smallest = arr[0];
int largest = arr[0];
for(int i=1; i<n; i++) {
    if(arr[i] < smallest)
        smallest = arr[i];
    if(arr[i] > largest)
        largest = arr[i];
}
```

```
printf("The smallest no. is %d\n", smallest);
printf("The largest no. is %d\n", largest);
return 0;
}
```

Output

Enter no. of elements in the array: 5

Enter the elements of the array:

39 22 7 96 4

The smallest no. is 4

The largest no. is 96

LAB-30

```
/*WAP for sorting elements of one dimensional array in ascending order using
bubble sort */
#include <stdio.h>
int main() {
    int arr[200], n;
    printf("How many no. you want to sort? : ");
    scanf("/d", &n);
    for(int i=0; i<n; i++) {
        scanf("./d", &arr[i]);
    }
    printf("The numbers before sorting are: \n");
    for(int i=0; i<n; i++) {
        printf("./d\t", arr[i]);
    }
    //Bubble sort algorithm for ascending order
    for(int i=0; i<n-1; i++) {
        for(int j=0; j<n-1-i; j++) {
            if(arr[j] > arr[j+1]) {
                int temp = arr[j];
                arr[j] = arr[j+1];
                arr[j+1] = temp;
            }
        }
    }
    printf("\n The no. in ascending order are: \n");
    for(int i=0; i<n; i++) {
        printf("./d\t", arr[i]);
    }
    return 0;
}
```

Output

How many no. you want to sort? : 5
2 43 22 26 3

The numbers before sorting are : .

2 43 22 26 3

The no. in ascending order are :

2 3 26 22 43

LAB-32

// WAP to read two M*N matrices & display their sum.

```
#include <stdio.h>
#define M 3
#define N 3

int main() {
    int matrix1[M][N], matrix2[M][N];
    printf("Enter the elements of the first matrix:\n");
    for(int i=0; i<M; i++) {
        for(int j=0; j<N; j++) {
            scanf("./d", &matrix1[i][j]);
        }
    }

    printf("\nEnter elements of the second matrix:\n");
    for(int i=0; i<M; i++) {
        for(int j=0; j<N; j++) {
            scanf("./d", &matrix2[i][j]);
        }
    }

    printf("\nThe sum of the matrix is:\n");
    for(int i=0; i<M; i++) {
        for(int j=0; j<N; j++) {
            printf("./d\t", matrix1[i][j] + matrix2[i][j]);
        }
        printf("\n");
    }

    return 0;
}
```

Output

Enter the elements of first matrix:

9 8 7 6 5 4 3 2 1

Enter elements of second matrix:

22 22 32 34 33 5 78 2 9

The sum of the matrix is :

20	20	39
40	38	9
82	4	20

LAB-32

```
1/WAP to find transpose of a matrix.  
#include <stdio.h>  
#define M 3  
#define N 3  
int main() {  
    int matrix[M][N];  
    printf("Enter the elements of the matrix: \n");  
    for(int i=0; i<M; i++) {  
        for(int j=0; j<N; j++) {  
            scanf("./d", &matrix[i][j]);  
        }  
    }  
    printf("\n The matrix to be transposed is: \n");  
    for(int i=0; i<M; i++) {  
        for(int j=0; j<N; j++) {  
            printf("./d\n", matrix[i][j]);  
        }  
    }  
    printf("\n The transpose of the matrix: \n");  
    for(int i=0; i<N; i++) {  
        for(int j=0; j<M; j++) {  
            printf("./d\n", matrix[j][i]);  
        }  
    }  
    printf("\n");  
}  
return 0;
```

Output

Enter the elements of matrix:

1 2 3 4 5 6 7 8 9

The matrix to be transposed is:

1	2	3
4	5	6
7	8	9

The transpose of matrix:

1	4	7
2	5	8
3	6	9

//WAP to read the name list of 20 students & display them in alphabetical form.

```
#include <stdio.h>
#include <string.h>
#define MAX_NAME_LENGTH 50
#define NUM_STUDENTS 20

int main() {
    char names[NUM_STUDENTS][MAX_NAME_LENGTH];
    printf("Enter names of 20 students:\n");
    for(int i=0; i<NUM_STUDENTS; i++) {
        scanf("%s", names[i]);
    }

    for(int i=0; i<NUM_STUDENTS-1; i++) {
        for(int j=i+1; j<NUM_STUDENTS; j++) {
            if(strcmp(names[i], names[j])>0) {
                char temp[MAX_NAME_LENGTH];
                strcpy(temp, names[i]);
                strcpy(names[i], names[j]);
                strcpy(names[j], temp);
            }
        }
    }

    printf("\nNames in alphabetical order:\n");
    for(int i=0; i<NUM_STUDENTS; i++) {
        printf("%s\t", names[i]);
    }
    return 0;
}
```

Output

Enter names of 20 students:

pal ana bina raj ben gris man tom tong ben

Names in alphabetical order:

ana ben ben bina gris man pal raj tom tong

LAB-34

```
//WAP to read a string & check whether its length is greater than 5 or not.  
#include <stdio.h>  
#include <string.h>  
#define MAX_STRING_LENGTH 200  
int main() {  
    char str[MAX_STRING_LENGTH];  
    printf("Enter a string: ");  
    scanf("%s", str);  
    if (strlen(str) > 5) {  
        printf("The length is greater than 5.\n");  
    } else {  
        printf("The length is not greater than 5.\n");  
    }  
    return 0;  
}
```

Output

Enter a string: Manish
The length is greater than 5.

LAB-35

```
/* C program to check whether two strings entered by the user are same or not using string function. */
#include <stdio.h>
#include <string.h>
#define MAX_STRING_LENGTH 200
int main() {
    char str1[MAX_STRING_LENGTH];
    char str2[MAX_STRING_LENGTH];
    printf("Enter the first string: ");
    scanf("%s", str1);
    printf("Enter the second string: ");
    scanf("%s", str2);
    if (strcmp(str1, str2) == 0) {
        printf("The two strings are the same.\n");
    } else {
        printf("The two strings are different.\n");
    }
    return 0;
}
```

Output

Enter the first string: Marish
Enter the second string: Bhusal
The two strings are different.

//WAP to find the greatest number among three numbers using function.

```
#include <stdio.h>
int findGreatest(int num1, int num2, int num3) {
    if (num1 >= num2 && num1 >= num3)
        return num1;
    else if (num2 >= num1 && num2 >= num3)
        return num2;
    else
        return num3;
}
```

```
int main() {
    int num1, num2, num3;
    printf("Enter three numbers: ");
    scanf("%d %d %d", &num1, &num2, &num3);
    int greatest = findGreatest(num1, num2, num3);
    printf("The greatest number is: %d\n", greatest);
    return 0;
}
```

Output.

Enter three numbers: 3 54 29

The greatest number is: 54.

// WAP to illustrate the "function with argument & return value".

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

```
int add(int num1, int num2) {
```

```
    return num1 + num2;
```

3

```
int main() {
```

```
    int num1, num2, result;
```

```
    printf("Enter two numbers: ");
```

```
    scanf("%d %d", &num1, &num2);
```

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

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

```
    return 0;
```

3

Output

Enter two numbers: 3 5.

The sum of 3 and 5 is 8.

/*WAP to illustrate the "function without argument & without return value*/

```
#include <stdio.h>
void welMessage() {
    printf("Hello World!");
```

{

```
int main() {
    welMessage();
    return 0;
```

{

-Output

Hello World!

LAB-39

1/WAP to find the factorial of a no. using recursion.

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

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

```
}
```

```
int main() {
    int num;
    printf("Enter a non-negative integer: ");
    scanf("%d", &num);
    if(num<0) {
        printf("Factorial is not defined for negative numbers.\n");
    } else {
        printf("Factorial of %d = %d\n", num, factorial(num));
    }
    return 0;
}
```

Output

Enter a non-negative integer: 5

Factorial of 5 = 120

LAB-40

//WAP to swap two numbers using call by reference (address) method.

```
#include <stdio.h>
void swap(int *a, int *b) {
    int temp = *a;
    *a = *b;
    *b = temp;
}
```

```
int main() {
    int num1, num2;
    printf("Enter two numbers: ");
    scanf("%d %d", &num1, &num2);
    printf("Before swapping: num1 = %d, num2 = %d\n", num1, num2);
    swap(&num1, &num2);
    printf("After swapping: num1 = %d, num2 = %d\n", num1, num2);
    return 0;
}
```

Output

Enter two numbers: 5 22

Before swapping: num1 = 5, num2 = 22

After swapping: num1 = 22, num2 = 5

LAB-42

// WAP to illustrate the use of pointer.

```
#include<stdio.h>
int main() {
    int num = 42;
    int *ptr;
    ptr = &num;
    printf("Value of num: %d\n", num);
    printf("Address of num: %p\n", &num);
    printf("Value at the address pointed by ptr: %d\n", *ptr);
    printf("Address stored in ptr: %p\n", ptr);
    *ptr = 99;
    printf("Modified value of num: %d\n", num);
    return 0;
}
```

Output

Value of num: 42

Address of num: 0x7ffc276

value at the address pointed by ptr: 42

Address stored in ptr: 0x7ffc276

Modified value of num: 99

```
//WAP for illustrating the use of double pointer.
#include<stdio.h>
int main() {
    int num = 42;
    int *ptr1;
    int **ptr2;
    ptr1 = &num;
    ptr2 = &ptr1;
    printf("Value of num using ptr1: %d\n", *ptr1);
    printf("Value of num using ptr2: %d\n", **ptr2);
    **ptr2 = 99;
    printf("Modified value of num: %d\n", num);
    return 0;
}
```

Output

Value of num using ptr1: 42

Value of num using ptr2: 42

Modified value of num: 99

LAB-43

//WAP for illustrating the use of array of pointer.

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

```
int main() {
    char *days[] = {"Sun", "Mon", "Tue", "Wed", "Thurs"};
    for (int i = 0; i < 5; i++) {
        printf("./s\n", days[i]);
    }
    return 0;
}
```

Output,

Sun

Mon

Tue

Wed

Thurs

LAB-44

```
/*WAP that reads the marks of n courses & finds the sum &
average using Dynamic Memory Allocation with malloc();*/
#include<stdio.h>
#include<stdlib.h>
int main() {
    int n;
    printf("Enter the no. of courses: ");
    scanf("%d", &n);
    int *marks = (int *)malloc(n * sizeof(int));
    if(marks == NULL) {
        printf("Memory allocation failed");
        return 1;
    }
    printf("Enter the marks for each course:\n");
    for(int i=0; i<n; i++) {
        scanf("%d", &marks[i]);
    }
    int sum = 0;
    for(int i=0; i<n; i++) {
        sum += marks[i];
    }
    double average = (double)sum/n;
    printf("Sum of marks: %.d\n", sum);
    printf("Average of marks: %.2f\n", average);
    free(marks);
    return 0;
}
```

Output

Enter the no. of courses: 3

Enter the marks for each course:

90 32 45

Sum of marks: 167

Average of marks: 55.67

LAB-45

```
/*WAP that reads the n elements of an array using malloc() & display  
the elements using pointer variable. */  
#include<stdio.h>  
#include<stdlib.h>  
int main() {  
    int n;  
    printf("Enter no. of elements: ");  
    scanf("./d", &n);  
    int *arr = (int *)malloc(n, sizeof(int));  
    if(arr == NULL) {  
        printf("Memory allocation failed.");  
        return 2;  
    }  
    printf("Enter the elements of the array:\n");  
scanf for(int i=0; i<n; i++) {  
        scanf("./d", &arr[i]);  
    }  
    printf("Elements of the array are: ");  
    for(int i=0; i<n; i++) {  
        printf("./d ", *(arr + i));  
    }  
    free(arr);  
    return 0;  
}
```

Output,

Enter the no. of elements: 3

Enter the elements of the array:

32

22

22

Elements of the array are: 32 22 22

LAB-46

```
/* Create a structure named student that has name, roll, marks, & remarks as members. Assume appropriate types & size of member. WAP to using structure to read & display the data entered by the user. */

#include <stdio.h>
struct student {
    char name[50];
    int roll;
    float marks;
    char remarks[50];
};

int main() {
    int n;
    printf("Enter the no. of students: ");
    scanf("./d", &n);
    struct student students[n];
    for(int i=0; i<n; i++) {
        printf("Name: ");
        scanf("./s", students[i].name);
        printf("Roll Number: ");
        scanf("./d", &students[i].roll);
        printf("Marks: ");
        scanf("./f", &students[i].marks);
        printf("Remarks: ");
        scanf("./s", students[i].remarks);
    }
    printf("Details of students:\n");
    for(int i=0; i<n; i++) {
        printf("Name: ./s\n", students[i].name);
        printf("Roll Number: ./d\n", students[i].roll);
        printf("Marks: ./2f\n", students[i].marks);
        printf("Remarks: ./s\n", students[i].remarks);
    }
    return 0;
}
```

Output,

Enter the no. of students: 2

Name: bengamin

Roll Number: 2

Marks: 49

Remarks: nice

Name: Robert

Roll Number: 33

Marks: 44

Remarks: wow

Details of students:

Name: bengamin

Roll Number: 2

Marks: 49

Remarks: nice

Name: Robert

Roll Number: 33

Marks: 44

Remarks: wow

LAB-47

/* Define a structure of employee having data members name, address, age & salary. Take the data for n employees in an array & find the average salary */

```
#include<stdio.h>
struct employee {
    char name[50];
    char address[200];
    int age;
    float salary;
};
```

```
int main() {
```

```
    int n;
```

```
    printf("Enter the no. of employees: ");
    scanf("%d", &n);
```

```
    struct employee employees[n];
```

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

```
        printf("Name: ");
```

```
        scanf("%s", employees[i].name);
```

```
        printf("Address: ");
```

```
        scanf("%s", employees[i].address);
```

```
        printf("Age: ");
```

```
        scanf("%d", &employees[i].age);
```

```
        printf("Salary: ");
```

```
        scanf("%.1f", &employees[i].salary);
```

```
}
```

~~for~~

```
float totalSalary = 0;
```

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

```
    totalSalary += employees[i].salary;
```

```
}
```

```
float averageSalary = totalSalary / n;
```

```
printf("\n Average Salary of %d employees: %.2f\n", n, averageSalary);
```

```
return 0;
```

3

Output

Enter the no. of employees: 2

Name: Benjamin

Address: Gehorahi

Age: 25

Salary: 43245.22

Name: Robert

Address: Tulsipur

Age: 43

Salary: 12945.234

Average Salary of 2 employees: 27795.18

```

/* WAP to read 3 students record with fields (roll-no, name, class & marks
in 5 subjects) & display their records along with their percentage of marks
obtained */

#include<stdio.h>
struct student {
    int roll-no;
    char name[50];
    char class-name[20];
    float marks[5];
};

int main() {
    int num-students = 3;
    struct student students[num-students];
    for(int i=0; i<num-students; i++) {
        printf("Roll Number: ");
        scanf("%d", &students[i].roll-no);
        printf("Name: ");
        scanf("%s", students[i].name);
        printf("Class: ");
        scanf("%s", students[i].class-name);
        printf("Enter marks in 5 subjects:\n");
        for(int j=0; j<5; j++) {
            printf("Subject %d: ", j+1);
            scanf("%f", &students[i].marks[j]);
        }
    }

    printf("\n Student Records:\n");
    for(int i=0; i<num-students; i++) {
        printf("\n Student %d:\n", i+1);
        printf("Roll Number: %d\n", students[i].roll-no);
        printf("Name: %s\n", students[i].name);
        printf("Class: %s\n", students[i].class-name);
        float totalMarks = 0;
        for(int j=0; j<5; j++) {
            totalMarks += students[i].marks[j];
        }
        float percentage = (totalMarks / 5.0);
    }
}

```

```

printf("Marks in 5 subjects: ");
for(int j= 0; j<5; j++) {
    printf("%.2f ", students[i].marks[j]);
}
printf("\nPercentage: %.2f%.1n", percentage);
}
return 0;
}

```

Output

RollNumber: 1
 Name: Benjamin
 Class: 22
 Enter marks in 5 subjects:
 Subject 1: 54
 Subject 2: 23
 Subject 3: 55
 Subject 4: 22
 Subject 5: 45
 RollNumber: 2
 Name: Robert
 Class: 22
 Enter marks in 5 subjects:
 Subject 1: 22.44
 Subject 2: 44.32
 Subject 3: 90
 Subject 4: 65.45
 Subject 5: 32.22
 RollNumber: 3
 Name: Nepolian
 Class: 22
 Enter marks in 5 subjects:
 Subject 1: 89.5
 Subject 2: 80
 Subject 3: 79
 Subject 4: 99
 Subject 5: 95.54

→ Student Records:

Student 1 :
 Roll Number: 2
 Name: Benjamin
 Class: 22
 Marks in 5 subjects: 54.00 23.00 55.00 22.00-
 45.00
 Percentage: 97.80%

 Student 2:
 Roll Number: 2
 Name: Robert
 Class: 22
 Marks in 5 subjects: 22.44 44.32 90.00 65.45 32.22
 Percentage: 48.89%

 Student 3:
 Roll Number: 3
 Name: Nepolian
 Class: 22
 Marks in 5 subjects: 89.50 80.00 79.00 99.00
 95.54
 Percentage: 88.60%

LAB-49

```
// WAP to demonstrate the use of union.  
#include<stdio.h>  
union myUnion {  
    int i;  
    float f;  
    char str[20];  
};  
int main()  
{  
    union myUnion u;  
    u.i = 20;  
    printf("Value of i: %d\n", u.i);  
    u.f = 220.5;  
    printf("Value of f: %.f\n", u.f);  
    strcpy(u.str, "C programming");  
    printf("Value of str: %s\n", u.str);  
    return 0;  
}
```

Output

Value of i: 20

Value of f: 220.500000

Value of str: C programming

// WAP to create a file named test.txt & write some text in it.

```
#include <stdio.h>
int main() {
    FILE *fp = fopen("test.txt", "w");
    if(fp == NULL) {
        printf("Error opening the file.\n");
        return 1;
    }
    fprintf(fp, "Hello, this is a test file.\n");
    fclose(fp);
    return 0;
}
```

LAB-52

//WAP to open a file & copy all its contents to another file.

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

```
int main() {
```

```
FILE *sourcefile = fopen("source.txt", "r");
```

```
if (sourcefile == NULL) {
```

```
printf("Error opening the source file.\n");
```

```
return 2;
```

3

```
FILE *destinationfile = fopen("destination.txt", "w");
```

```
if (destinationfile == NULL) {
```

```
printf("Error opening the destination file.\n");
```

```
fclose(sourcefile);
```

```
return 2;
```

3

```
char ch;
```

```
while ((ch = fgetc(sourcefile)) != EOF) {
```

```
fputc(ch, destinationfile);
```

3

```
fclose(sourcefile);
```

```
fclose(destinationfile);
```

```
return 0;
```

3

```

/*WAP to create a file named employee.txt & write name, id, address, &
salary of a employee to this file.*/
#include<stdio.h>
struct Employee{
    char name[50];
    int id;
    char address[200];
    float salary;
};
int main(){
    struct Employee employee;
    FILE *fp = fopen("employee.txt", "w");
    if(fp == NULL){
        printf("File cannot be created or opened.");
        return 2;
    }
    printf("Enter name of a employee: ");
    gets(employee.name);
    printf("Enter id of ./s: ", employee.name);
    scanf("./d", &employee.id);
    printf("Enter address of ./s: ", employee.name);
    scanf("./s", employee.address);
    printf("Enter salary of ./s: ", employee.name);
    scanf("./f", &employee.salary);
    printf("\nNow writing data to file---");
    fprintf(fp, "Name=./s\nID=./d\nAddress=./s\nSalary=./.2f",
            employee.name, employee.id, employee.address, employee.salary);
    printf("\nCompleted");
    fclose(fp);
    return 0;
}

```

Output

Enter name of a employee: Poe
 Enter id of Poe: 1
 Enter address of Poe: Dang
 Enter salary of Poe: 22222.22
 Now writing data to file---
 Completed

LAB-53

```
/* Some text file is given, create another text file replacing the following words  
"Ram" to "Hari", "Sita" to "Gita" & "Gorinda" to "Shiva".  
#include<stdio.h>  
#include<string.h>  
int main() {  
    FILE *fp, *fpp;  
    char c[20];  
    fp = fopen("replace-name.txt", "r");  
    if (fp == NULL) {  
        printf("Cannot open file");  
        return 1;  
    }  
    fpp = fopen("newReplaced-name.txt", "w");  
    if (fpp == NULL) {  
        printf("Cannot create file");  
        fclose(fp);  
        return 1;  
    }  
    while (fscanf(fp, "%s", c) != EOF) {  
        if (strcmp(c, "Ram") == 0)  
            fprintf(fpp, "Hari ");  
        else if (strcmp(c, "Sita") == 0)  
            fprintf(fpp, "Gita ");  
        else if (strcmp(c, "Gorinda") == 0)  
            fprintf(fpp, "Shiva ");  
        else  
            fprintf(fpp, "%s ", c);  
    }  
    fclose(fp);  
    fclose(fpp);  
    printf("Words replaced successfully.\n");  
    return 0;  
}
```

Output
Words replaced successfully.

```

/* Program to create a file named "employee.dat" & store records of N employee
in the file. These records contain name, identification number, office name &
occupation of the employee. Also display name of those employees whose office name
is "Everest Bank" & occupation is "manager". */

#include<stdio.h>
#include<string.h>
struct Employee {
    char name[30];
    int id;
    char office_name[30];
    char occupation[30];
};

int main() {
    struct Employee emp;
    int N;
    FILE *fp;
    fp=fopen("employee.dat", "wb+");
    if(fp==NULL) {
        printf("Cannot open the file.");
        return 1;
    }
    printf("\nEnter the no. of employees: ");
    scanf("%d", &N);
    for(int i=0; i<N; i++) {
        printf("Name: ");
        scanf("%s", emp.name);
        printf("ID: ");
        scanf("%d", &emp.id);
        printf("Office Name: ");
        scanf("%s", emp.office_name);
        printf("Occupation: ");
        scanf("%s", emp.occupation);
        fwrite(&emp, sizeof(emp), 1, fp);
    }
    rewind(fp);
    printf("\nEmployees with office name 'Everest Bank' & occupation 'manager': \n");
    while(fread(&emp, sizeof(emp), 1, fp)==1) {
        if(strcmp(emp.office_name, "Everest Bank")==0 && strcmp(emp.occupation,
            "manager")==0) {
            printf("./s\n", emp.name);
        }
    }
    fclose(fp);
    return 0;
}

```

Output

Enter the no. of employees: 2

Name: Benjamin

ID: 2

Office Name: Everest Bank

Occupation: manager

Name: Poe

ID: 2

Office Name: Google

Occupation: manager

Employees with office name "Everest Bank" & occupation 'manager':

Benjamin

LAB-55

//WAP to draw a line in graphics window.

```
#include <graphics.h>
```

```
int main() {
```

```
    initwindow(640, 480, "Line Drawing Example");
```

```
    setcolor(RED);
```

```
    line(200, 200, 300, 200);
```

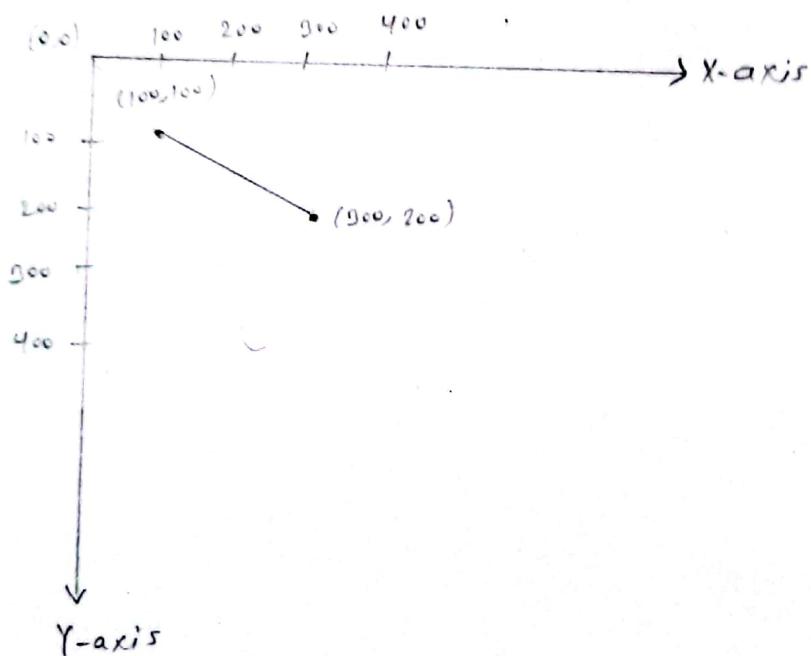
```
    getch();
```

```
    closegraph();
```

```
    return 0;
```

3

Output



LAB-56

//WAP to draw a rectangle in graphics window.

```
#include <graphics.h>
```

```
int main() {
```

```
    initwindow(640, 480, "Rectangle Drawing Example");
```

```
    setcolor(BLUE);
```

```
    rectangle(200, 200, 300, 200);
```

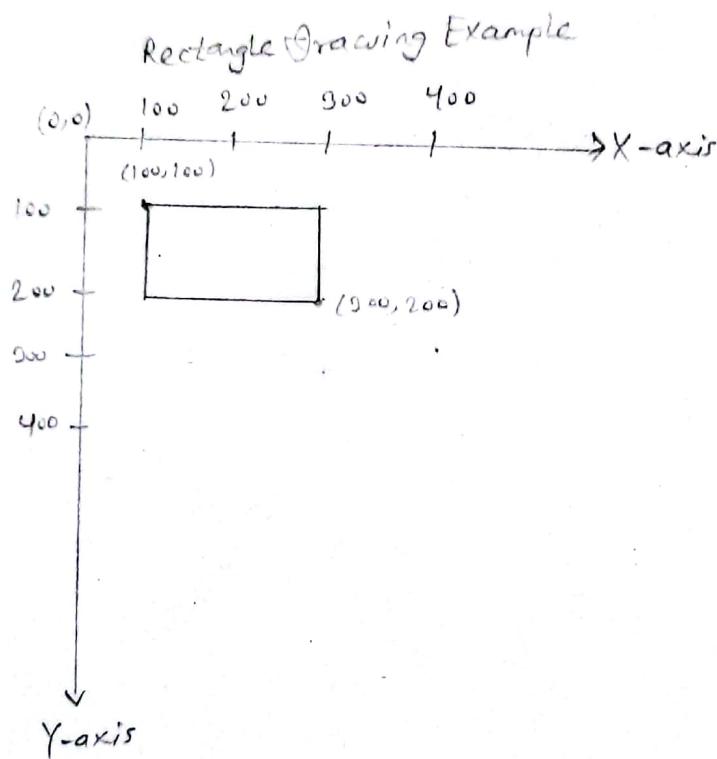
```
    getch();
```

```
    closegraph();
```

```
    return 0;
```

3

Output

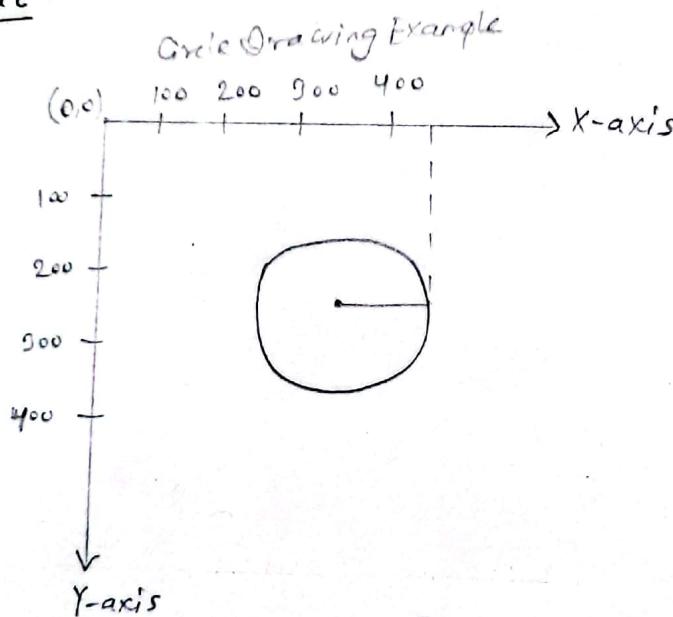


LAB-57

```
//WAP to draw a circle of radius 70 in graphics window.  
#include <graphics.h>  
int main() {  
    initwindow(640,480,"Circle Drawing Example");  
    setcolor(RED);  
    circle(320,240,70);  
    getch();  
    closegraph();  
    return 0;  
}
```

3

Output



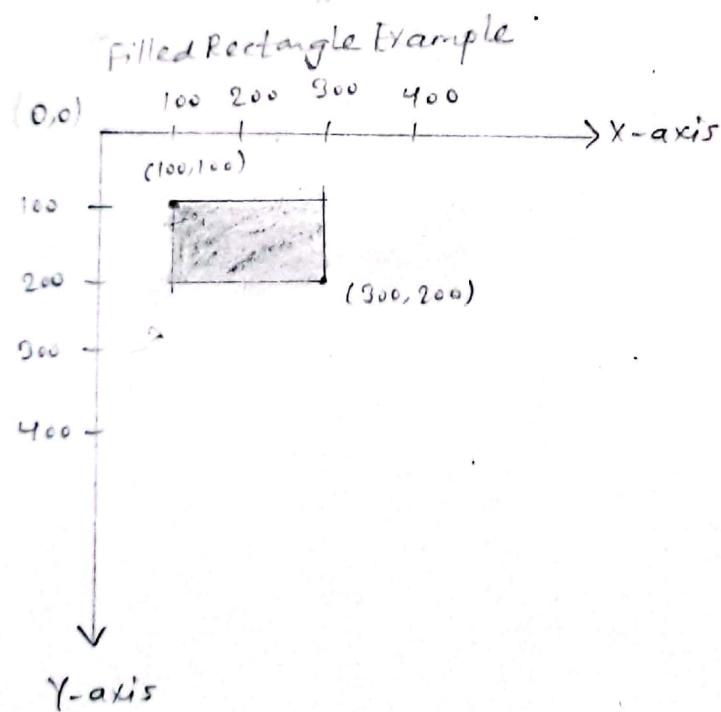
LAB-58

//WAP to draw a rectangle filled with color.

```
#include <stdio.h>
#include <graphics.h>
int main()
{
    initwindow(640, 480, "Filled Rectangle Example");
    setfillstyle(SOLID_FILL, GREEN);
    bar(200, 200, 300, 200);
    getch();
    closegraph();
    return 0;
}
```

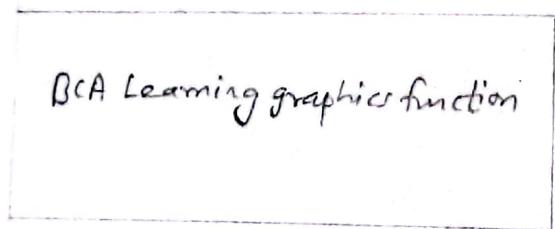
3

Output



```
/*WAP to display the text "BCA learning graphics function" in graphics  
window */  
#include <graphics.h>  
int main()  
{  
    initwindow(640, 480, "Text Display Example");  
    setcolor(BLUE);  
    settextstyle(DEFAULT_FONT, HORIZ_DIR, 2);  
    outtextxy(200, 200, "BCA learning graphics function");  
    getch();  
    closegraph();  
    return 0;  
}
```

3

Output

LAB-60

```
//WAP to draw a triangle in graphics window.  
#include <graphics.h>  
int main()  
{  
    initwindow(640, 480, "Triangle Drawing Example");  
    int x1 = 200, y1 = 300;  
    int x2 = 300, y2 = 200;  
    int x3 = 500, y3 = 900;  
    line(x1, y1, x2, y2);  
    line(x2, y2, x3, y3);  
    line(x3, y3, x1, y1);  
    getch();  
    closegraph();  
    return 0;  
}
```

3

Output

