

List of Practical: (Can be done in any imperative language)

1. Basic Programs:

a. Write a program to display the message HELLO WORLD.

```
#include <stdio.h>
main( )
{
    printf("HELLO WORLD");
}
```

- b. Write a program to declare some variables of type int, float and double. Assign some values to these variables and display these values.

```
#include <stdio.h>
main( )
{
    int    a, b, c=11;
    char   d, e, f='m';
    float  g, h, i=2.0;
    double j, k, l=3.0;
    a = 12;
    d = 'a';
    g = 4.0;
    j = 5.0;
    printf("Enter character value: ");
    scanf("%c",&e);
    printf("Enter integer value: ");
    scanf("%d",&b);
    printf("Enter floating-point value: ");
    scanf("%f",&h);
    printf("Enter double value: ");
    scanf("%lf",&k);
    printf("Integer values are %d %d %d\n", a, b, c);
    printf("Character values are %c %c %c\n", d, e, f);
    printf("Floating-point values are %f %f %f\n", g, h, i);
    printf("Double values are %lf %lf %lf\n", j, k, l);
}
```

- c. Write a program to find the addition, subtraction, multiplication and division of two numbers.

```
#include <stdio.h>
main()
{
    int f, s, add, sub, mul;
    float div;
    printf ("Enter two integers\n");
    scanf ("%d %d", &f, &s);
    add = f + s;
    sub = f - s;
    mul = f * s;
    div = f / (float)s;
    printf ("Sum = %d\n", add);
    printf ("Difference = %d\n", sub);
    printf ("Multiplication = %d\n", mul);
    printf ("Division = %.2f\n", div);
}
```

2. Programs on variables:

a. Write a program to swap two numbers without using third variable.

```
#include <stdio.h>
main()
{
    int a, b;
    printf ("Enter First Number: ");
    scanf ("%d", &a);
    printf ("Enter Second Number: ");
    scanf ("%d", &b);
    b=b-a;
    a=b+a;
    b=a-b;
    printf ("\nNew Value of First Number: %d", a);
    printf ("\nNew Value of Second Number: %d", b);
}
```

b. Write a program to find the area of rectangle, square and circle.

```
#include<stdio.h>
#define PI 3.14
main()
{
    int side, radius, length, breadth;
    printf ("\nArea of Square");
    printf ("\nEnter the side:");
    scanf ("%d", &side);
    printf ("\nThe Area of a square is: %d", side*side);
    printf ("\nArea of Rectangle");
    printf ("\nEnter the length:");
    scanf ("%d", &length);
    printf ("\nEnter the breadth:");
    scanf ("%d", &breadth);
    printf ("\nThe Area of rectangle is: %d", length*breadth);
    printf ("\nArea of Circle");
    printf ("\nEnter the radius:");
    scanf ("%d", &radius);
    printf ("\nThe Area of a circle is: %.2f", PI*radius*radius);
}
```

c. Write a program to find the volume of a cube, sphere, and cylinder.

```
#include<stdio.h>
#define PI 3.1415
main()
{
    int s, r, h;
    printf ("\nVolume of Cube");
    printf ("\nEnter the side:");
    scanf ("%d", &s);
    printf ("\nThe Volume of a cube is: %d", 6*s*s);
    printf ("\nVolume of Sphere");
    printf ("\nEnter the radius:");
    scanf ("%d", &r);
    printf ("\nThe Volume of a sphere is: %.2f", 4*PI*r*r);
    printf ("\nVolume of Cylinder");
    printf ("\nEnter the radius:");
    scanf ("%d", &r);
    printf ("\nEnter the height:");
    scanf ("%d", &h);
    printf ("\nThe Volume of a cylinder is: %.2f", 2*PI*r*(r+h));
}
```

3. Conditional statements and loops(basic)

- a. Write a program to enter a number from the user and display the month name. If number >13 then display invalid input using switch case.

```
#include<stdio.h>
void main()
{
    int n;
    printf ("Enter Month Number: ");
    scanf ("%d", &n);
    switch (n)
    {
        case 1:
            printf ("January\n");
            break;
        case 2:
            printf ("February\n");
            break;
        case 3:
            printf ("March\n");
            break;
        case 4:
            printf ("April\n");
            break;
        case 5:
            printf ("MAy\n");
            break;
        case 6:
            printf ("June\n");
            break;
        case 7:
            printf ("July\n");
            break;
        case 8:
            printf ("August\n");
            break;
        case 9:
            printf ("September\n");
            break;
        case 10:
            printf ("October\n");
            break;
        case 11:
            printf ("November\n");
            break;
        case 12:
            printf ("December\n");
            break;
        default:
            printf ("Invalid Input\n");
            break;
    }
}
```

b. Write a program to check whether the number is even or odd.

```
#include <stdio.h>
main()
{
    int n;
    printf ("Enter a number: ");
    scanf ("%d", &n);
    if (n%2==0)
        printf ("Number entered is EVEN");
    else
        printf ("Number entered is ODD");
}
```


c. Write a program to check whether the number is positive, negative or zero.

```
#include <stdio.h>
main()
{
    int n;
    printf ("Enter a number: ");
    scanf ("%d", &n);
    if (n<0)
        printf ("Number entered is NEGATIVE");
    else if (n>0)
        printf ("Number entered is POSITIVE");
    else
        printf ("Number entered is ZERO");
}
```

d. Write a program to find the factorial of a number.

```
#include <stdio.h>
int factorial (int n);
main()
{
    int n, res;
    printf ("Enter number to find factorial: ");
    scanf ("%d", &n);
    res = factorial (n);
    printf ("Factorial is %d", res);
}
int factorial(int n)
{
    int i;
    long int prod = 1;
    if (n > 1)
    {
        for (i = 2; i <= n; ++i)
            prod *= i;
    }
    return(prod);
}
```

e. Write a program to check whether the entered number is prime or not.

```
#include <stdio.h>
main()
{
    int n, i, flag = 0;
    printf ("Enter a positive number: ");
    scanf ("%d", &n);
    for (i=2; i<=n/2; ++i)
    {
        if (n%i==0)
        {
            flag=1;
            break;
        }
    }
    if (flag==0)
        printf ("%d is a prime number", n);
    else
        printf ("%d is not a prime number", n);
}
```

f. Write a program to find the largest of three numbers.

```
#include <stdio.h>
int maximum (int x, int y)
{
    int z;
    z = (x >= y) ? x : y;
    return (z);
}
main ()
{
    int a, b, c;
    printf ("\na = ") ;
    scanf ("%d", &a) ;
    printf ("\nb = ") ;
    scanf ("%d", &b) ;
    printf ("\nc = ") ;
    scanf ("%d", &c) ;
    printf ("\n\nMaximum = %d", maximum (c, maximum (a, b)));
}
```

4. Conditional statements and loops(advanced)

a. Write a program to find the sum of squares of digits of a number.

```
#include <stdio.h>
main()
{
    int a, tot=0, b;
    printf ("\nEnter a number: ");
    scanf ("%d", &a);
    while (a>0)
    {
        b=(a%10);
        tot=tot+(b*b);
        a=a/10;
    }
    printf("\nSum of squares of digits is %d\n", tot);
}
```

b. Write a program to reverse the digits of an integer.

```
#include <stdio.h>
main()
{
    int n;
    printf ("\nEnter a number : ");
    scanf ("%d", &n);
    printf ("\nReverse is : ");
    while (n>0)
    {
        printf ("%d", n%10);
        n=n/10;
    }
}
```

c. Write a program to find the sum of numbers from 1 to 100.

```
#include <stdio.h>
main()
{
    int n, tot=0;
    for (n=1;n<=100;n++)
        tot+=n;
    printf("\nSum of numbers is %d", tot);
}
```

d. Write a program to print the Fibonacci series.

```
#include <stdio.h>
main()
{
    int a=0,b=1,c=0, n;
    printf ("\nEnter a number: ");
    scanf ("%d", &n);
    printf ("\nFibonacci Series: \n");
    printf ("%d\n%d\n",a,b);
    c=a+b;
    while (c<=n)
    {
        printf ("%d\n", c);
        a=b;
        b=c;
        c=a+b;
    }
}
```


e. Write a program to find whether a given number is palindrome or not.

```
#include <stdio.h>
main()
{
    int n, revnum = 0, rem, orignum;
    printf ("Enter an number: ");
    scanf("%d", &n);
    orignum = n;
    while( n!=0 )
    {
        rem = n%10;
        revnum = revnum*10 + rem;
        n /= 10;
    }
    if(orignum == revnum)
        printf("%d is a palindrome.", orignum);
    else
        printf("%d is not a palindrome.", orignum);
}
```

f. Write a program that solves the quadratic equation

```
#include <stdio.h>
main()
{
    double a, b, c, root, x1, x2;
    printf ("Enter value for a: ");
    scanf ("%d", &a);
    printf ("Enter value for b: ");
    scanf ("%d", &b);
    printf ("Enter value for c: ");
    scanf ("%d", &c);
    root = sqrt(b * b - 4 * a * c);
    x1 = ( - b + root) / (2 * a);
    x2 = ( - b - root) / (2 * a);
    printf ("\nSolution of Quadratic Equation is %f and %f", x1, x2);
}
```

Possible Variations:

Write a program to solve the quadratic equation $4x^2 + 8x + 5 = 0$

Write a program to solve the quadratic equation $3x^2 + 5x + 3 = 0$

Write a program to solve the quadratic equation $5x^2 + 7x + 7 = 0$

Write a program to solve the quadratic equation $8x^2 + 2x + 1 = 0$

Write a program to solve the quadratic equation $2x^2 + 3x + 9 = 0$

g. Write a program to check whether the entered number is Armstrong or not.

```
#include <stdio.h>
main()
{
    int a, tot, b, n;
    printf ("Enter a number: ");
    scanf ("%d", &n);
    a=n;
    tot=0;
    while (a>0)
    {
        b=(a%10);
        tot=tot+(b*b*b);
        a=a/10;
    }
    if (n==tot)
        printf ("%d is an Armstrong Number\n", n);
    else
        printf ("%d is not an Armstrong Number\n", n);
}
```

h. Write a program to count the digit in a number

```
#include <stdio.h>
main()
{
    int n, i=0;
    printf ("\nEnter a number : ");
    scanf ("%d", &n);
    while (n>0)
    {
        n=n/10;
        i++;
    }
    printf ("\nNumber has %d digits", i);
}
```

5. Programs on patterns:

a. Programs on different patterns.

a) Write a program to display the following design output:

```
*  
**  
***  
****  
*****
```

```
#include <stdio.h>  
main()  
{  
    int i, a;  
    for (i=0;i<=5;i++)  
    {  
        for (a=1;a<=i;a++)  
            printf("*");  
        printf("\n");  
    }  
}
```

b) Write a program to display the following design output:

**

*

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

```
main()
```

```
{
```

```
    int i, a;
```

```
    for (i=5;i>=1;i--)
```

```
    {
```

```
        for (a=1;a<=i;a++)
```

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

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

```
    }
```

```
}
```

c) Write a program to display the following design output:

```
1
12
123
1234
12345
```

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

d) Write a program to display the following design output:

55555

4444

333

22

1

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

```
main()
```

```
{
```

```
    int i, a;
```

```
    for (i=5;i>=1;i--)
```

```
    {
```

```
        for (a=1;a<=i;a++)
```

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

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

```
    }
```

```
}
```


e) Write a program to display the following design output:

12345

1234

123

12

1

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

```
main()
```

```
{
```

```
    int i, a;
```

```
    for (i=5;i>=1;i--)
```

```
    {
```

```
        for (a=1;a<=i;a++)
```

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

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

```
    }
```

```
}
```

f) Write a program to display the following design output:

1

22

333

4444

55555

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

```
main()
```

```
{
```

```
    int i, a;
```

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

```
    {
```

```
        for (a=1;a<=i;a++)
```

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

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

```
    }
```

```
}
```

g) Write a program to display the following design output:

```
*****
****  ****
***   ***
**    **
*     *
```

```
#include<stdio.h>
void main()
{
    int i, j, k;
    for (i=5; i>=1; i--)
    {
        for (j=1; j<=i; j++)
            printf("*");
        for (k=5; k>i; k--)
            printf(" ");
        for (k=5; k>i; k--)
            printf(" ");
        for (j=1; j<=i; j++)
            printf("*");
        printf("\n");
    }
}
```

h) Write a program to display the following design output:

```
*           *
**          **
***         ***
****        ****
*****       *****
```

```
#include<stdio.h>
void main()
{
    int i, j, k;
    for (i=1; i<=5; i++)
    {
        for (j=1; j<=i; j++)
            printf("*");
        for (k=5; k>i; k--)
            printf(" ");
        for (k=5; k>i; k--)
            printf(" ");
        for (j=1; j<=i; j++)
            printf("*");
        printf("\n");
    }
}
```

i) Write a program to display the following design output:

```
*****
****  ****
***   ***
**    **
*     *
*     *
**    **
***   ***
****  ****
*****
```

```
#include<stdio.h>
void main()
{
    int i, j, k;
    for (i=5; i>=1; i--)
    {
        for (j=1; j<=i; j++)
            printf ("*");
        for (k=5; k>i; k--)
            printf (" ");
        for (k=5; k>i; k--)
            printf (" ");
        for (j=1; j<=i; j++)
            printf ("*");
        printf ("\n");
    }
    for (i=1; i<=5; i++)
    {
        for (j=1; j<=i; j++)
            printf ("*");
        for (k=5; k>i; k--)
            printf (" ");
        for (k=5; k>i; k--)
            printf (" ");
        for (j=1; j<=i; j++)
            printf ("*");
        printf ("\n");
    }
}
```

j) Write a program to display the following design output:

```
*           *
**          **
***         ***
****        ****
*****       *****
*****       *****
***          ***
**           **
*            *
```

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

```
void main()
```

```
{
    int i, j, k;
    for (i=1; i<=5; i++)
    {
        for (j=1; j<=i; j++)
            printf ("*");
        for (k=5; k>i; k--)
            printf (" ");
        for (k=5; k>i; k--)
            printf (" ");
        for (j=1; j<=i; j++)
            printf ("*");
        printf ("\n");
    }
    for (i=4; i>=1; i--)
    {
        for (j=1; j<=i; j++)
            printf ("*");
        for (k=5; k>i; k--)
            printf (" ");
        for (k=5; k>i; k--)
            printf (" ");
        for (j=1; j<=i; j++)
            printf ("*");
        printf ("\n");
    }
}
```

k) Write a program to display the following design output:

```
 *
 ***
 *****
 *******
 *********
```

```
#include<stdio.h>
void main()
{
    int i, j, k;
    for (i=1; i<=5; i++)
    {
        for (k=5; k>i; k--)
            printf (" ");
        for (j=1; j<i; j++)
            printf ("*");
        for (j=1; j<=i; j++)
            printf ("*");
        printf ("\n");
    }
}
```

1) Write a program to display the following design output:

*

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

```
void main()
```

```
{
```

```
    int i, j, k;
```

```
    for (i=4; i>=0; i--)
```

```
    {
```

```
        for (k=5; k>i; k--)
```

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

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

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

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

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

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

```
    }
```

```
}
```


m) Write a program to display the following design output:

```
  *
 ***
 *****
 *****
 *****
 *****
 *****
 ***
  *
```

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

```
void main()
```

```
{
    int i, j, k;
    for (i=1; i<=5; i++)
    {
        for (k=5; k>i; k--)
            printf (" ");
        for (j=1; j<i; j++)
            printf ("*");
        for (j=1; j<=i; j++)
            printf ("*");
        printf ("\n");
    }
    for (i=4; i>=1; i--)
    {
        for (k=5; k>i; k--)
            printf (" ");
        for (j=1; j<i; j++)
            printf ("*");
        for (j=1; j<=i; j++)
            printf ("*");
        printf ("\n");
    }
}
```

n) Write a program to display the following design output:

```
*****
*****
*****
***
*
***
*****
*****
*****
```

```
#include<conio.h>
```

```
void main()
```

```
{
    int i, j, k;
    for (i=5; i>1; i--)
    {
        for (k=5; k>i; k--)
            printf (" ");
        for (j=1; j<i; j++)
            printf ("*");
        for (j=1; j<=i; j++)
            printf ("*");
        printf ("\n");
    }
    for (i=1; i<=5; i++)
    {
        for (k=5; k>i; k--)
            printf (" ");
        for (j=1; j<i; j++)
            printf ("*");
        for (j=1; j<=i; j++)
            printf ("*");
        printf ("\n");
    }
}
```

6. Functions:

a. Programs on Functions.

i. Write a program to have two functions that perform the following tasks:

1. First function will accept two numbers and print the sum
2. Second function will accept one number and print the cube

```
#include <stdio.h>
sum();
cube();
main()
{
    sum();
    cube();
}
sum()
{
    int a, b;
    printf("ENTER A and B: ");
    scanf("%d %d", &a, &b);
    printf("\nSUM IS %d", a+b);
}
cube()
{
    int x;
    printf("\nENTER X: ");
    scanf("%d", &x);
    printf("\ncube is %d", x*x*x);
}
```

ii. Write a program to accept 2 numbers. Using functions that accept parameter perform the following tasks:

1. Add 2 numbers
2. Find the square of the first number
3. Find cube of the second number

Print the sum, square and cube.

```
#include <stdio.h>
sum(int x, int y);
square(int a);
cube(int b);
main()
{
    int a, b;
    printf("ENTER A and B: ");
    scanf("%d %d", &a, &b);
    sum(a, b);
    square(a);
    cube(b);
}
sum(int x, int y)
{
    int c;
    c=x+y;
    printf("\nSUM is: %d", c);
}
square(int a)
```

```

{
    printf("\nSQUARE is: %d", a*a);
}
cube(int b)
{
    printf("\nCUBE is: %d", b*b*b);
}

```

iii. Write a program to accept 2 numbers. Using functions that accept parameter and returns value to perform the following tasks:

1. Add 2 numbers
2. Find the square of the first number
3. Find cube of the second number

Print the sum, square and cube.

```

#include <stdio.h>
int sum(int x, int y);
int square(int a);
disp (int x);
main(){
    int a, b, c;
    printf("ENTER A and B: ");
    scanf("%d %d", &a, &b);
    c = sum(a, b);
    printf ("\nSum of A & B is\t: ");
    disp (c);
    c = square (a);
    printf ("\nSquare of A is\t: ");
    disp (c);
}
int sum (int x, int y){
    int c;
    c=x+y;
    return c;
}
int square (int a){
    return a*a;
}
disp (int x){
    printf("%d", x);
}

```

7. Recursive functions

- a. Write a program to find the factorial of a number using recursive function.

```
#include <stdio.h>
int n_fact(int n);
main()
{
    int n, res;
    printf ("Enter number to find factorial: ");
    scanf ("%d", &n);
    res=n_fact(n);
    printf ("Factorial is %d", res);
}
int n_fact (int n)
{
    int result;
    if ( n == 0 )
        result = 1;
    else
        result = n * n_fact ( n-1 );
    return (result);
}
```

b. Write a program to find the sum of natural number using recursive function.

```
#include <stdio.h>
int addNumbers(int n);
main()
{
    int n;
    printf("Enter a positive integer: ");
    scanf("%d", &n);
    printf("Sum = %d", addNumbers(n));
}
int addNumbers(int n)
{
    if(n!=0)
        return n+addNumbers(n-1);
    else
        return n;
}
```

8. Arrays

- a. Write a program to find the largest value that is stored in the array.

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

b. Write a program using pointers to compute the sum of all elements stored in an array.

```
#include <stdio.h>
main()
{
    int num[5] = {75, 10, 15, 8, 7};
    int *pt;
    int i, sum=0;
    pt = num;
    for(i=0; i<5; i++)
    {
        sum = sum + *(pt+i);
    }
    printf("Sum = %d\n", sum);
}
```


- c. Write a program to arrange the 'n' numbers stored in the array in ascending and descending order.

```
#include <stdio.h>
main(void)
{
    int a[10], i=0, j=0, n, t;
    printf ("\n Enter the no. of elements: ");
    scanf ("%d", &n);
    printf ("\n");
    for (i = 0; i <n; i++)
    {
        printf ("\n Enter element %d: ", (i+1));
        scanf ("%d", &a[i]);
    }
    for (j=0; j<(n-1); j++)
    {
        for (i=0; i<(n-1); i++)
        {
            if (a[i+1] < a[i])
            {
                t = a[i];
                a[i] = a[i + 1];
                a[i + 1] = t;
            }
        }
    }
    printf ("\n Ascending order: ");
    for (i=0; i<n; i++)
    {
        printf (" %d", a[i]);
    }
    printf ("\n Descending order: ");
    for (i=n; i>0; i--)
    {
        printf (" %d", a[i-1]);
    }
}
```

d. Write a program that performs addition and subtraction of matrices.

```
#include <stdio.h>
#define MROW 20
#define MCOL 30
readinput (int a[][MCOL], int nr, int nc);
compute (int a[][MCOL], int b[][MCOL], int c[][MCOL], int nr, int nc, int o);
writeoutput (int c[][MCOL], int nr, int nc);
main()
{
    int nrows, ncols, noper, a[MROW][MCOL], b[MROW][MCOL], c[MROW][MCOL];
    printf ("Enter 1 to Add / Enter 2 to Subtract? ");
    scanf ("%d", &noper);
    printf ("How many rows? ");
    scanf ("%d", &nrows);
    printf ("How many columns? ");
    scanf ("%d", &ncols);
    printf ("\n\nFirst table: \n");
    readinput(a, nrows, ncols);
    printf ("\n\nSecond table: \n");
    readinput (b, nrows, ncols);
    compute (a, b, c, nrows, ncols, noper);
    printf ("\n\nComputation of the elements:\n\n");
    writeoutput (c, nrows, ncols);
}

readinput (int a[][MCOL], int m, int n)
{
    int row, col;
    for (row = 0; row < m; ++row)
    {
        printf ("\nEnter data for row no. %2d\n", row + 1);
        for (col = 0; col < n; ++col)
            scanf ("%d", &a[row][col]);
    }
}

compute (int a[][MCOL], int b[][MCOL], int c[][MCOL], int m, int n, int o)
{
    int row, col;
    for (row = 0; row < m; ++row)
    {
        for (col = 0; col < n; ++col)
        {
            if (o==1) c[row][col] = a[row][col] + b[row][col];
            if (o==2) c[row][col] = a[row][col] - b[row][col];
        }
    }
}

writeoutput (int a[][MCOL], int m, int n)
{
    int row, col;
    for (row = 0; row < m; ++row)
    {
        for (col = 0; col < n; ++col)
            printf ("\t%d", a[row][col]);
        printf ("\n");
    }
}
```

e. Write a program that performs multiplication of matrices.

```
#include <stdio.h>
#define MROW 20
#define MCOL 30
void readinput (int a[][MCOL], int nr, int nc);
void compute (int a[][MCOL], int b[][MCOL], int c[][MCOL], int nr, int nc);
void writeoutput (int c[][MCOL], int nr, int nc);
main()
{
    int nrows, ncols;
    int a[MROW][MCOL], b[MROW][MCOL], c[MROW][MCOL];
    printf ("How many rows? ");
    scanf ("%d", &nrows);
    printf ("How many columns? ");
    scanf ("%d", &ncols);
    printf ("\n\nFirst table: \n");
    readinput(a, nrows, ncols);
    printf ("\n\nSecond table: \n");
    readinput (b, nrows, ncols);
    compute (a, b, c, nrows, ncols);
    printf ("\n\nSums of the elements:\n\n");
    writeoutput (c, nrows, ncols);
}

void readinput (int a[][MCOL], int m, int n)
{
    int row, col;
    for (row = 0; row < m; ++row)
    {
        printf ("\nEnter data for row no. %2d\n", row + 1);
        for (col = 0; col < n; ++col)
            scanf ("%d", &a[row][col]);
    }
}

void compute (int a[][MCOL], int b[][MCOL], int c[][MCOL], int m, int n)
{
    int row, col;
    for (row = 0; row < m; ++row)
    {
        for (col = 0; col < n; ++col)
            c[row][col] = a[row][col] * b[row][col];
    }
}

void writeoutput (int a[][MCOL], int m, int n)
{
    int row, col;
    for (row = 0; row < m; ++row)
    {
        for (col = 0; col < n; ++col)
            printf ("%4d", a[row][col]);
        printf ("\n");
    }
}
```

9. Pointers

a. Write a program to demonstrate the use of pointers.

Write a program to swap two numbers without using third variable. Use the concept of pointers.

```
#include <stdio.h>
main()
{
    int a=10,b=20;
    printf("BEFORE SWAP\n");
    reprint(a, b);
    swap (&a, &b);
    printf("AFTER SWAP\n");
    reprint(a, b);
}
swap(int *a, int *b)
{
    *a=*a+*b;
    *b=*a-*b;
    *a=*a-*b;
}
reprint(int a, int b)
{
    printf("A=%d\nB=%d\n", a, b);
}
```

b. Write a program to perform addition and subtraction of two pointer variables.

```
#include <stdio.h>
main()
{
    int first, second, *p, *q, sum;
    printf ("Enter two integers to add\n");
    scanf ("%d%d", &first, &second);
    p = &first;
    q = &second;
    sum = *p + *q;
    printf ("Sum of entered numbers = %d\n", sum);
}
```

10. Structures and Unions

a. Programs on structures.

Write a program to accept and display month name, abbreviation and number of days. Use the concept of structure.

Example Output:

December abbreviated as Dec has 31 days

```
#include <stdio.h>
#include <string.h>
typedef struct month
{
    char name[10];
    char abbr[4];
    int days;
}MONTH;
main()
{
    MONTH m;
    strcpy(m.name, "January");
    strcpy(m.abbr, "Jan");
    m.days=31;
    printf("%s abbreviated as %s has %d days\n", m.name, m.abbr, m.days);
}
```

b. Programs on unions.

Write a program to accept and display employee number, name and salary. Use the concept of union.

Example Output:

Employee No: 1200

Employee Name: Harsh

Employee Salary: 1200.00

```
#include <stdio.h>
union emp
{
    char name[30];
    float salary;
    int emp_no;
}e;
main()
{
    printf("Enter Employee No:\n");
    scanf("%d", &e.emp_no);
    printf("\nEmployee No :%d\n", e.emp_no);
    printf("Enter Name:\n");
    scanf("%s", &e.name);
    printf("\nEmployee Name :%s\n", e.name);
    printf("Enter Salary: \n");
    scanf("%f", &e.salary);
    printf("\nEmployee Salary: %.2f", e.salary);
}
```