* Example 1:

Matrix Addition using Pointers

```
#include <stdio.h>
int main() {
    int a[3][3], b[3][3], c[3][3];
    int *p1 = &a[0][0], *p2 = &b[0][0],
        *p3 = &c[0][0];
    int i, j;
    printf("Enter first matrix:\n");
    for(i=0; i<9; i++) scanf("%d", p2+i);
    printf("Enter second matrix:\n");
    for(i=0; i<9; i++) scanf("%d", p2+i);
    for(i=0; i<9; i++) *(p3+i) = *(p1+i) + *(p2+i);
    printf("Resultant Matrix:\n");
    for(i=0; i<9; i++){
        printf("%d", *(p3+i));
        if((i+1)%3==0) printf("\n");
    }
    return 0;
}
```

Output:

Enter first matrix:

(User inputs 1 2 3 4 5 6 7 8 9)

Enter second matrix:

(User inputs 1 2 3 4 5 6 7 8 9):

Resultant Matrix:

$$\begin{bmatrix} 2 & 4 & 6 \\ 8 & 10 & 12 \\ 14 & 16 & 18 \end{bmatrix}$$

* Example 2;
Find largest and smallest in 3 x 3 Matrix using pointer

⟹
```
#include <stdio.h>
int main() {
int a[3][3], *p = &a[0][0];
int i, max, min;
printf("Enter elements of 3x3 matrix:\n");
for(i=0; i<9; i++) scanf("%d", p+i);
max = min = *p;
for(i=1; i<9; i++) {
if(*(p+i) > max) max = *(p+i);
if(*(p+i) < min) min = *(p+i);
}
printf("Largest = %d\nsmallest = %d\n",
max, min);
return 0;
}
```

⟹ output : largest = 30
smallest = 5

Q 1 Perform addition, subtraction and multiplication operation on two matrices.

⟹
```
#include <stdio.h>
int main() {
int a[2][2], b[2][2], sum[2][2],
sub[2][2], mul[2][2];
int i, j, k;
printf("Enter elements of first 2x2 matrix:\n");
for(i=0; i<2; i++)
for(j=0; j<2; j++)
scanf("%d", &a[i][j]);
```

```
printf("Enter elements of second 2x2 matrix:
\n");
for (i=0; i<2; i++)
for (j=0; j<2; j++)
scanf("%d", &b[i][j];
for (i=0; i<2; i++)
for (j=0; j<2; j++)
scanf("%d", &b[i][j]);
for (i=0; i<2; i++)
for (j=0; j<2; j++){
sum[i][j] = a[i][j] + b[i][j];
sub[i][j] = a[i][j] - b[i][j];
mul[i][j] = 0;
for (k=0; k<2; k++)
mul[i][j] += a[i][k] * b[k][j];
}
printf("\nAddition of matrices:\n");
for (i=0; i<2 i++){
for (j=0; j<2; j++)
printf("%d", sub[i][j]);
printf("\n");
}
printf("\nsubtraction of matrices:\n");
for (i=0; i<2 i++){
for (j=0; j<2; j++)
printf("%d", sub[i][j]);
printf("\n");
}
Printf("\nmultiplication of matrices:\n");
for (i=0; i<2; i++){
for (j=0; j<2; j++)
printf("%d", mul[i][j]);
printf("\n");
}
return 0; }
```

Date

⇒ Output:

Enter elements of first 2x2 matrix:

1 2

3 4

Enter elements of second 2x2 matrix:

5 6

7 8

Addition of matrices:

6 8

10 12

Subtraction of matrices:

-4 -4

-4 -4

Multiplication of Matrices:

19 22

43 50

Q2 Sort all the elements of a 4x4 matrix and store result in a single dimension array.

⇒

```
#include <stdio.h>
int main() {
int a[4][4], arr[16], i, j, k, temp;
printf("Enter elements of 4x4 matrix: \n");
for (i=0; i<4; i++)
for (j=0; j<4; j++)
  scanf("%d", &a[i][j]);
k=0;
for (i=0; i<4; i++)
for (j=0; j<4; j++)
  arr[k++] = a[i][j];
```

```
int main() {
    int year;
    printf("Enter a year: ");
    scanf("%d", &year);
    if (isLeap(year))
        printf("%d is a leap year\n", year);
    else
        printf("%d is not a leap year\n", year);
    return 0;
}
```

⟹ Output:

Enter a year: 2024
2024 is a leap year

Q3/ Recursive function to calculate factorial

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

⇒ Output:

Enter number: 5

Factorial of 5 is 120.

Q4 Swap two integers using call by value.

→

```
#include <stdio.h>
void swap(int x, int y) {
    int temp = x;
    x = y;
    y = temp;
    printf("Inside function after
        swap: x=%d, y=%d\n", x, y);
}
int main() {
    int a, b;
    printf("Enter two numbers: ");
    scanf("%d %d", &a, &b);
    printf("Before function call: a=%d, b=%d\n",
        a, b);
    swap(a, b);
    printf("After function call: a=%d, b=%d\n",
        a, b);
    return 0;
}
```

⇒ Output:-

Enter two numbers: 5 10

Before function call: a=5, b=10

Inside function after swap: x=10, y=5

After function call: a=5, b=10.

Q5 Function to find max and min using call by reference

→
```
#include <stdio.h>
void findMinMax(int arr[], int n, int *min,
int *max) {
    *min = *max = arr[0];
    for(int i = 1; i<n ; i++) {
        if (arr[i] > *max)
            *max = arr[i];
        if (arr[i] < *min)
            *min = arr[i];
    }
}
int main() {
    int arr[5], i, min, max;
    printf("Enter 5 elements: ");
    for (i=0; i<5 ; i++)
        scanf ("%d", &arr[i]);

    findMinMax(arr, 5, &min, &max);
    printf("Maximum = %d \n Minimum = %d \n",
    max, min);
    return 0;
}
```

⟹ Output:- Enter 5 elements: 2 8 5 1 9
Maximum = 9
Minimum = 1

Page No.
Date

Q6 Calculator using separate functions.

```
#include <stdio.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}
float divi(int a, int b) {return (float)a/b;}
int main() {
    int a, b, choice;
    printf("Enter two numbers: ");
    scanf("%d %d", &a, &b);
printf("1. Add\n2. Subtract\n3. Multiply\n4. Divide\nEnter your choice: ");
    scanf("%d", &choice);
switch(choice) {
    case 1: printf("Sum = %d\n", add(a,b));
break;
    case 2: printf("Difference = %d\n", sub(a,b));
break;
    case 3; Printf("Product = %d\n", mul(a,b));
break;
    default: Printf("Invalid choice\n");
}
    return 0;
}
```

Output:-

Enter two number : 12 4
1. Add
2. Substract
3. Multiply
4. Divide
Enter any choice : 3
Product = 48

Page No.
Date

Q8 All loop programs using recursion.

→
```
#include <stdio.h>
int sum(int n) {
  if (n == 0)
    return 0;
  else
    return n + sum(n-1);
}
int main() {
  int n;
  printf("Enter n: ");
  scanf("%d", &n);
  printf("Sum of first %d numbers = %d\n", n,
  sum(n));
  return 0;
}
```

⇒ Output :-

```
Enter n: 5
Sum of first 5 numbers = 15
```