

AIM:	Demonstrate the use of two-dimensional arrays to solve a given problem
Program 1	
PROBLEM STATEMENT:	Write a program to perform Matrix Addition, Subtraction, Multiplication, Transpose of Matrix and Norm of Matrix. Dimensions of matrices will be decided by user.
PROGRAM:	<pre> #include <stdio.h> #include<math.h> //Function to take values of element of matrices from the user void read_matrix(int row, int columns,int mat[][columns]){ for (int i=0;i<row;i++){ for(int j=0;j<columns;j++){ scanf("%d",&mat[i][j]); } } } //Function to print the values of each element of matrices void print_matrix(int row, int columns,int mat[][columns]){ for (int i=0;i<row;i++){ for(int j=0;j<columns;j++){ printf("%d ",mat[i][j]); } printf("\n"); } } //Function to add two matrices void add_matrix(int row1,int col1, int mat1[][col1],int row2, int col2, int mat2[][col2],int res[][col1]){ if (row1!=row2 col1!=col2){ printf("\nAddition of this matrix is not possible"); printf("====="); } else{ printf("\nA+B: \n"); for (int i=0;i<row1;i++){ for(int j=0;j<col1;j++){ res[i][j]=mat1[i][j]+mat2[i][j]; } } print_matrix(row1,col1,res); printf("====="); } } //Function to subtract two matrices </pre>

```

void sub_matrix(int row1,int col1, int mat1[][col1],int row2, int col2, int mat2[][col2],int
res[][col1]){
    if (row1!=row2 || col1!=col2){
        printf("\nSubtraction of this matrix is not possible");
        printf("=====");
    }
    else{
        printf("\nA-B: \n");
        for (int i=0;i<row1;i++){
            for(int j=0;j<col1;j++){
                res[i][j]=mat1[i][j]-mat2[i][j];
            }
        }
        print_matrix(row1,col1,res);
        printf("=====");
    }
}

//Function to multiply two matrices
void multi_matrix(int row1,int col1, int mat1[][col1],int row2, int col2, int mat2[][col2],int
res[][col1]){
    if(col1==row1){
        printf("\nA X B: \n");
        for(int i=0;i<row1;i++){
            {
                for(int j=0;j<col2;j++)
                {
                    res[i][j]=0;
                    for(int k=0;k<col1;k++)
                    {
                        res[i][j]+=mat1[i][k]*mat2[k][j];
                    }
                }
            }
        }
        print_matrix(row1,col2,res);
        printf("=====");
    }
    else{
        printf("Multiplication of this matrix is not possible");
        printf("=====");
    }
}

//Function to find a norm of a matrix
void norm_matrix(int row1, int col1, int mat1[][col1]){
    float sum=0;
    for(int i=0;i<row1;i++){
        for(int j=0;j<col1;j++){

```

```

        sum+=pow(mat1[i][j],2);
    }
}
sum=pow(sum,0.5);
printf("\nNorm of matrix A is: ");
printf("%.2f",sum);
printf("\n=====");
}

int main()
{
    int row1,col1;
    printf("Enter the number of rows of matrices A: ");
    scanf("%d",&row1);
    printf("Enter the number of columns of matrices A: ");
    scanf("%d",&col1);
    int matri1[row1][col1];
    read_matrix(row1,col1,matri1);

    int row2,col2;
    printf("Enter the number of rows of matrices B: ");
    scanf("%d",&row2);
    printf("Enter the number of columns of matrices B: ");
    scanf("%d",&col2);
    int matri2[row2][col2];
    read_matrix(row2,col2,matri2);
    printf("=====");
    printf("\nMatrix A is:\n");
    print_matrix(row1,col1,matri1);
    printf("=====");
    printf("\nMatrix B is:\n");
    print_matrix(row2,col2,matri2);
    printf("=====");

    int res[row1][col1];

    add_matrix(row1,col1,matri1,row2,col2,matri2,res);

    sub_matrix(row1,col1,matri1,row2,col2,matri2,res);

    multi_matrix(row1,col1,matri1,row2,col2,matri2,res);

    norm_matrix(row1,col1,matri1);

    return 0;
}

```

RESULT:

```
Enter the number of rows of matrices A: 2
Enter the number of columns of matrices A: 2
2
2
2
2
Enter the number of rows of matrices B: 2
Enter the number of columns of matrices B: 2
2
2
2
2
=====
Matrix A is:
2 2
2 2
=====
Matrix B is:
2 2
2 2
=====
A+B:
4 4
4 4
=====
A-B:
0 0
0 0
=====
A X B:
8 8
8 8
=====
Norm of matrix A is: 4.00
=====
```

Program 2**PROBLEM
STATEMENT :**

Write a program which reads the current year followed by N followed by a list of N employee numbers and their current ages. Produce a list showing the years in which the employees retire (become 65 years old). If more than one employee retires in a given year, then include them all under the same heading.
For example: Year Number

1986 896743
1988 674501
450926

PROGRAM:

```
#include <stdio.h>
//Takes the employee id and current age from user and calculate the year of retirement
void read_empl(int n, int ret_emp[][2]){
    int cur_yr;
    printf("Enter current year: ");
    scanf("%d",&cur_yr);
    for (int i=0;i<n;i++){
        printf("Enter employee number: ");
        scanf("%d",&ret_emp[i][1]);
        printf("Enter the current age of employee: ");
        scanf("%d",&ret_emp[i][0]);
        ret_emp[i][0]=cur_yr+65-ret_emp[i][0];
    }
}
//to swap two years and employee id
void swap(int *xp, int *yp)
{
    int temp = *xp;
    *xp = *yp;
    *yp = temp;
}
//To sort the year
void selectionSort(int n, int arr[][2])
{
    int i, j, min_idx;
    for (i = 0; i < n-1; i++)
    {
        min_idx = i;
        for (j = i+1; j < n; j++)
            if (arr[j][0] < arr[min_idx][0])
                min_idx = j;
        swap(&arr[min_idx][0], &arr[i][0]);
        swap(&arr[min_idx][1], &arr[i][1]);
    }
}
//To print the desired the output
void print_empl(int n, int ret_emp[n][2]){
    selectionSort(n,ret_emp);
    printf("Year\tEmployee ID\n");
    for (int i=0;i<n;i++){
        if(i!=0 && ret_emp[i][0]==ret_emp[i-1][0])
            printf("\t%d\n",ret_emp[i][1]);
```

```

        else
            printf("%d\t%d\n",ret_emp[i][0],ret_emp[i][1]);
        }
    }
}
int main()
{
    int n;
    printf("Enter the number of employee: ");
    scanf("%d",&n);
    int ret_emp[n][2];
    read_empl(n,ret_emp);
    print_empl(n,ret_emp);
    return 0;
}

```

RESULT:

```

Enter the number of employee: 5
Enter current year: 2022
Enter employee number: 400
Enter the current age of employee: 23
Enter employee number: 500
Enter the current age of employee: 45
Enter employee number: 600
Enter the current age of employee: 60
Enter employee number: 700
Enter the current age of employee: 23
Enter employee number: 800
Enter the current age of employee: 21
Year      Employee ID
2027      600
2042      500
2064      400
          700
2066      800

```

Program 3

PROBLEM STATEMENT:

Given a nxn matrix, find whether it is an upper triangular matrix or not. Also print the upper triangle of the matrix.

PROGRAM:

```

#include <stdio.h>
//Function to take values of element of matrices from the user
void read_matrix(int n,int mat[][n]){

```

```

    for (int i=0;i<n;i++){
        for(int j=0;j<n;j++){
            scanf("%d",&mat[i][j]);
        }
    }
}

//Function to print the values of each element of matrices
void print_matrix(int n,int mat[][n]){
    for (int i=0;i<n;i++){
        for(int j=0;j<n;j++){
            printf("%d ",mat[i][j]);
        }
        printf("\n");
    }
}

//Checks whether matrix is upper trigular matrix
int upper_trigular_matrices(int n,int mat[][n]){
    for(int i=0;i<n;i++){
        for(int j=0;j<n;j++){
            if (i>j && mat[i][j]!=0)
                return 0;
        }
    }
    return 1;
}

//Checks whether matrix is Zero matrix
int zero_matrix(int n, int mat[][n]){
    for(int i=0;i<n;i++){
        for(int j=0;j<n;j++){
            if (mat[i][j]!=0)
                return 0;
        }
    }
    return 1;
}

int main()
{
    int n;
    printf("Enter the dimension of square matrices A: ");
    scanf("%d",&n);
    int mat1[n][n];
    read_matrix(n,mat1);
    printf("\nMatrix A is:\n");
    print_matrix(n,mat1);
    if (zero_matrix(n,mat1))
        printf("Matrix A is Zero Matrix");
}

```

```
else if (upper_trigular_matrices(n,matri1))
    printf("Matrix A is Upper Trigular Matrix");
else
    printf("Matrix A is not Upper Trigular Matrix");

return 0;
}
```

RESULT:

```
Enter the dimension of square matrices A: 3
2 4 6
0 3 32
0 0 21

Matrix A is:
2 4 6
0 3 32
0 0 21
Matrix A is Upper Trigular Matrix
```

CONCLUSION:

We learned the use of two-dimensional arrays to solve a given problem