

Name	Adwait S Purao
UID no.	2021300101
Experiment No.	6

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.
ALGORITHM:	<ol style="list-style-type: none"> 1. START 2. Define void function zero with a float 2D array mat[m][n] as parameter 3. Initialize all elements to 0 4. Define void function print with a float 2D array mat[m][n] as parameter 5. I=0 6. J=0 7. Print mat[i][j] 8. J++ 9. Repeat 7,8 till j<n 10. I++ 11. Repeat 6,7,8,9 and 10 till i<m 12. Define void function add with 2 2D float array mat1[m][n] and mat2[a][b] as parameters 13. I=0 14. J=0 15. Print mat1[i][j]+mat2[i][j] 16. J++ 17. Repeat 15,16 till j<n 18. I++ 19. Repeat 14,15,16,17 and 18 till i<m 20. Define void function sub with 2 2D float array mat1[m][n] and mat2[a][b] as parameters 21. I=0 22. J=0 23. Print mat1[i][j]-mat2[i][j] 24. J++ 25. Repeat 23,24 till j

	<p>26. I++</p> <p>27. Repeat 22,23,24,25 and 26 till i</p> <p>28. Define void function multiply with 2 2D float array mat1[m][n] and mat2[a][b] as parameters</p> <p>29. Initialize 2D array mat3</p> <p>30. Call function zero(m,b,mat3)</p> <p>31. I=0</p> <p>32. J=0</p> <p>33. K=0</p> <p>34. mat3[i][j] += mat1[i][k]*mat2[k][j]</p> <p>35. k++</p> <p>36. repeat 34 and 35 till k</p> <p>37. j++</p> <p>38. repeat 33, 34, 35, 36 and 37 till j<b</p> <p>39. i++</p> <p>40. repeat 32, 33, 34, 35, 36, 37, 38 and 39 till i<m</p> <p>41. call function print(m,b,mat3)</p> <p>42. Define void function transpose with a 2D float array mat[m][n] as parameter</p> <p>43. Initialize 2D array newmat of dimension n x m</p> <p>44. I=0</p> <p>45. J=0</p> <p>46. Newmat[i][j]=mat[j][i]</p> <p>47. J++</p> <p>48. Repeat 46 and 47 till j</p> <p>49. I++</p> <p>50. Repeat 45, 46, 47, 48 and 49 till i<n</p> <p>51. Call function print(m,b,newmat)</p> <p>52. Define int function matrixnorm with a 2D float array mat[m][n]</p> <p>53. Initialize sum = 0.00</p> <p>54. I=0</p> <p>55. J=0</p> <p>56. Sum += square of mat[i][j]</p> <p>57. J++</p> <p>58. Repeat 54 and 55 till j<n</p> <p>59. I++</p> <p>60. Repeat 55, 56, 57 ,58 and 59 till i<m</p> <p>61. Sum = square root of sum</p> <p>62. Return sum</p> <p>63. Define integer main function</p>
--	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

64. Input dimensions of matrix 1 m and n
65. Input matrix 1 [m][n]
66. Input dimensions of matrix 2 a and b
67. Input matrix 2 [a][b]
68. If (m=a and b=n)
 call function matrixadditon(m,n,mat1,a,b,mat2)
 else
 print Addition not possible
69. If (m=a and b=n)
 call function matrixsubtraction(m,n,mat1,a,b,mat2)
 else
 print subtraction not possible
70. If(n=a)
 call function matrixmultiplication(m,n,mat1,a,b,mat2)
 else
 print multiplication not possible
71. Call function transpose(m,n,mat1)
72. Call function norm(a,b,mat2)
73. Print value of function matrixnorm(m,n,mat1)
74. Print value of function matrixnorm(a,b,mat2)
75. STOP

FLOWCHART:

PROGRAM:

```
#include<stdio.h>
#include<math.h>
void zero(int m,int n,float mat[m][n])
{
    for(int i=0;i<m;i++)
        for(int j=0;j<n;j++)
            mat[i][j]=0.0;
}
void print(int m,int n,float mat[m][n])
{
    for(int i=0;i<m;i++)
    {
        for(int j=0;j<n;j++)
            printf("%.2f\t",mat[i][j]);
        printf("\n");
    }
}
void matrixaddition(int m,int n,float mat1[m][n],int a,int b,float mat2[a][b])
{
    for(int i=0;i<m;i++)
    {
        for(int j=0;j<n;j++)
            printf("%.2f\t",mat1[i][j]+mat2[i][j]);
        printf("\n");
    }
}
void matrixsubtraction(int m,int n,float mat1[m][n],int a,int b,float mat2[a][b])
{
    for(int i=0;i<m;i++)
    {
        for(int j=0;j<n;j++)
            printf("%.2f\t",mat1[i][j]-mat2[i][j]);
        printf("\n");
    }
}
void matrixmultiplication(int m,int n,float mat1[m][n],int a,int b,float mat2[a][b])
{
    float mat3[m][b];
    zero(m,b,mat3);
    for(int i=0;i<m;i++)
        for(int j=0;j<b;j++)
            for(int k=0;k<n;k++)
                mat3[i][j] += mat1[i][k]*mat2[k][j];
    print(m,b,mat3);
}
```

```

}
void matrixtranspose(int m,int n,float mat[m][n])
{
    float newmat[n][m];
    for(int i=0;i<n;i++)
    for(int j=0;j<m;j++)
    newmat[i][j]=mat[j][i];
    print(n,m,newmat);
}
double matrixnorm(int m,int n,float mat[m][n])
{
    double sum=0.0;
    for(int i=0;i<m;i++)
    {
        for(int j=0;j<n;j++)
        {
            sum += pow(mat[i][j],2);
        }
    }
    sum = sqrt(sum);
    return sum;
}
int main()
{
    int m,n,a,b;
    printf("Enter dimensions of Matrix 1:\n");
    scanf("%d %d",&m,&n);
    float mat1[m][n];
    printf("Enter elements of Matrix 1:\n");
    for(int i=0;i<m;i++)
    for(int j=0;j<n;j++)
    scanf("%f",&mat1[i][j]);
    print(m,n,mat1);
    printf("Enter dimensions of Matrix 2:\n");
    scanf("%d %d",&a,&b);
    float mat2[a][b];
    printf("Enter elements of Matrix 2:\n");
    for(int i=0;i<a;i++)
    for(int j=0;j<b;j++)
    scanf("%f",&mat2[i][j]);
    print(a,b,mat2);
    printf("\n Addition of Matrices:\n");
    if(m==a && n==b)
    matrixaddition (m,n,mat1,a,b,mat2);
}

```

```

else
printf("Addition of matrices is not possible");
printf("\n Subtraction of Matrices:\n");
if(m==a && n==b)
matrixsubtraction(m,n,mat1,a,b,mat2);
else
printf("Subtraction of matrices is not possible");
printf("\n Multiplication of Matrices:\n");
if(n==a)
matrixmultiplication(m,n,mat1,a,b,mat2);
else
printf("Multiplication of matrices is not possible");
printf("\nTranspose of the 2 Matrices:\n");
matrixtranspose(m,n,mat1);
printf("\n");
matrixtranspose(a,b,mat2);
printf("Norm of Matrix 1 : %.2f\n",matrixnorm(m,n,mat1));
printf("Norm of Matrix 2 : %.2f",matrixnorm(a,b,mat2));
return 0;
}

```

RESULT:

PROBLEMS	OUTPUT	DEBUG CONSOLE	TERMINAL	PROBLEMS	OUTPUT	DEBUG CONSOLE	TERMINAL
			Enter dimensions of Matrix 1: 2 2 Enter elements of Matrix 1: 12 33 54 34 12.00 33.00 54.00 34.00 Enter dimensions of Matrix 2: 2 2 Enter elements of Matrix 2: 54 33 66 55 54.00 33.00 54.00 34.00				54.00 34.00 Enter dimensions of Matrix 2: 2 2 Enter elements of Matrix 2: 54 33 66 55 54.00 33.00 66.00 55.00
			Addition of Matrices: 66.00 66.00 120.00 89.00				Subtraction of Matrices: -42.00 0.00 -12.00 -21.00

Multiplication of Matrices:

2826.00 2211.00

5160.00 3652.00

Transpose of the 2 Matrices:

12.00 54.00

33.00 34.00

54.00 66.00

33.00 55.00

Norm of Matrix 1 is: 72.84

Norm of Matrix 2 is: 106.71

PS C:\Users\aspur\C PROGRAMS\mydirectory> █

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

ALGORITHM:

1. START
2. Define void function selection sort with an 2D integer array mat[n][2]
3. Define integer variables min, ind
4. I=0
5. Ind = i
6. J=I+1
7. If(mat[j][0] < mat[ind][0]) ,index = j
8. J++
9. Repeat 7 and 8 till j
10. Initialize temp1 to mat[ind][0]
11. Mat[ind][0] = mat[i][0]
12. Mat[i][0] = temp1
13. Initialize temp2 to mat[index][1]
14. Mat[ind][1] = mat[i][1]
15. Mat[i][1] = temp1
16. I++
17. Repeat steps 5 to 16 till i
18. Define integer main function
19. Input current year year

	<p>20. Input the number of employees n</p> <p>21. I=0</p> <p>22. Input current age mat[i][0] and employee number mat[i][1]</p> <p>23. Mat[i][0] = year + 65 – mat[i][0]</p> <p>24. Call function selection sort(n,mat)</p> <p>25. I=0</p> <p>26. If(I not equal to 0 and mat[i][0]=mat[i-1][0]) print Tabspace mat[i][1] else print mat[i][0] Tabspace mat[i][1]</p> <p>27. STOP</p>
FLOWCHART:	
PROGRAM:	<pre> #include<stdio.h> void selectionsort(int n,int mat[n][2]) { int min,ind; for(int i=0;i<n-1;i++) { ind=i; for(int j=i+1;j<n;j++) { if(mat[j][0]<mat[ind][0]) { </pre>


```

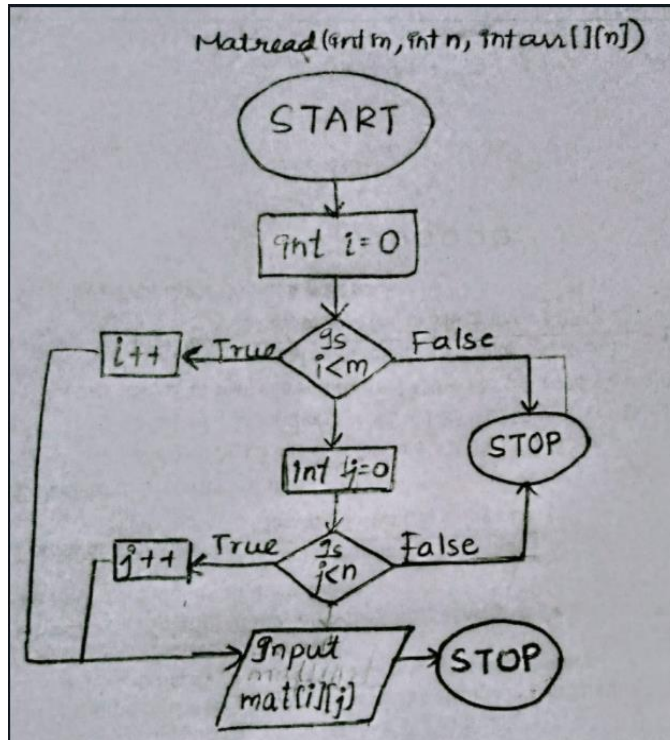
        ind=j;
    }
}
int temp1=mat[ind][0];
mat[ind][0]=mat[i][0];
mat[i][0]=temp1;
int temp2=mat[ind][1];
mat[ind][1]=mat[i][1];
mat[i][1]=temp2;
}
}
int main()
{
    int year,n;
    printf("Enter current year: ");
    scanf("%d",&year);
    printf("Enter the number of employees: ");
    scanf("%d",&n);
    int mat[n][2];
    for(int i=0;i<n;i++)
    {
        printf("Enter Employee Number and current age: ");
        scanf("%d %d",&mat[i][1],&mat[i][0]);
        mat[i][0] = year + 65 - mat[i][0];
    }
    selectionsort(n,mat);
    printf("Retiring Year\tEmployee Number\n");
    for(int i=0;i<n;i++)
    {
        if(i!=0 && mat[i][0]==mat[i-1][0])
            printf("\t\t%d\n",mat[i][1]);
        else
            printf("%d\t\t%d\n",mat[i][0],mat[i][1]);
    }
    return 0;
}

```

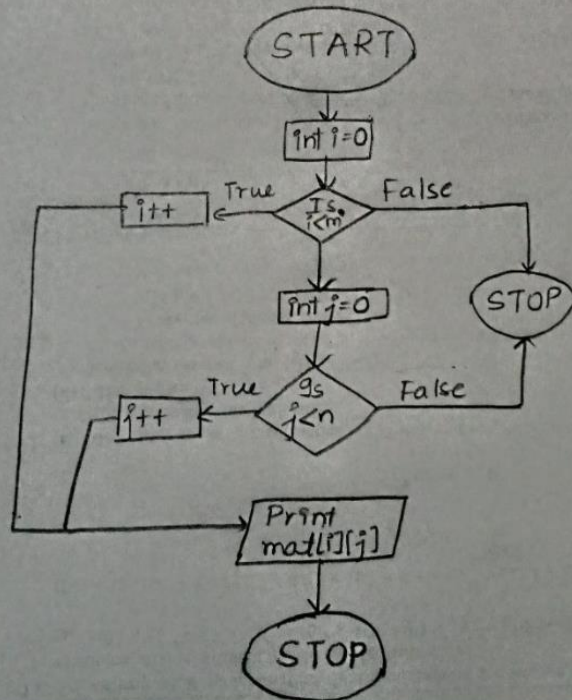
RESULT:	<div data-bbox="87 401 1127 961"> <div> PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL </div> <pre> e } ; if (\$?) { .\employee } Enter current year: 2022 Enter the number of employees: 3 Enter Employee Number and current age: 1 23 Enter Employee Number and current age: 2 50 Enter Employee Number and current age: 3 59 Retiring Year Employee Number 2028 3 2037 2 2064 1 PS C:\Users\aspur\C PROGRAMS\mydirectory> </pre> </div>
Program 3	
PROBLEM STATEMENT:	<p>Given a nxn matrix, find whether it is an upper triangular matrix or not. Also print the upper triangle of the matrix.</p>
ALGORITHM:	<ol style="list-style-type: none"> 1) START 2) Declare function as void matread with integer parameters m, n and mat[][n]) 3) For(int i=0;i<m ;i++) For(int j=0;j<n;j++) Read mat[i][j] 4) Declare function as void matprint with integer parameters m, n and int mat[][n]) 5) For(int i=0;i<m ;i++) For(int j=0;j<n;j++) Print mat[i][j] 6) Declare function void matcheck with integer parameters n, arr[][n] 7) Declare int row,col, up=1 8) for (row = 0; row < n; row++) for (col = 0; col < n; col++) if (col < row && arr[row][col] != 0) up=0 9) if(up=1)

10) Print The entered matrix is a upper triangular matrix
 11) Print The matrix is
 12) Call function as Matprint(n,n,arr)
 13) Else
 Print This is Not a Upper triangular matrix
 14) Define function main
 15) Initialize integer variables m and n
 16) Input number of rows and columns i.e m and n
 17) Declare int arr[m][n]
 18) Input elements of array
 19) Call function matread(m,n,arr)
 20) Call function matchcheck(n,arr)
 21) STOP

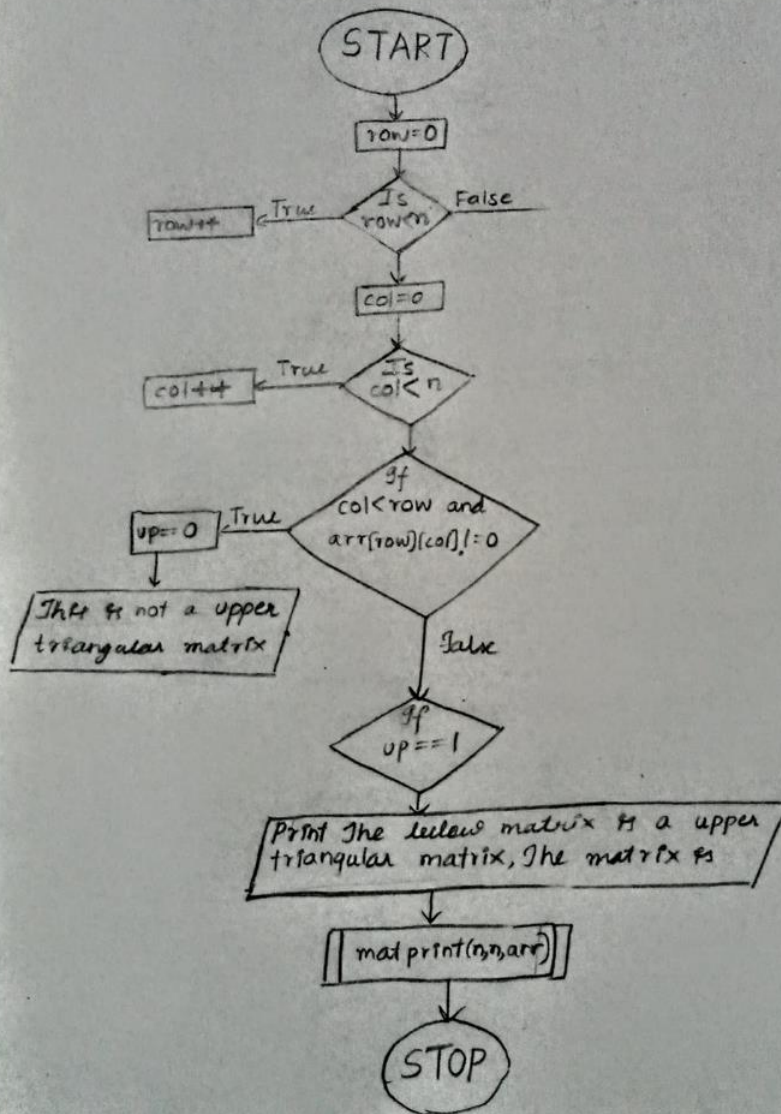
FLOWCHART:

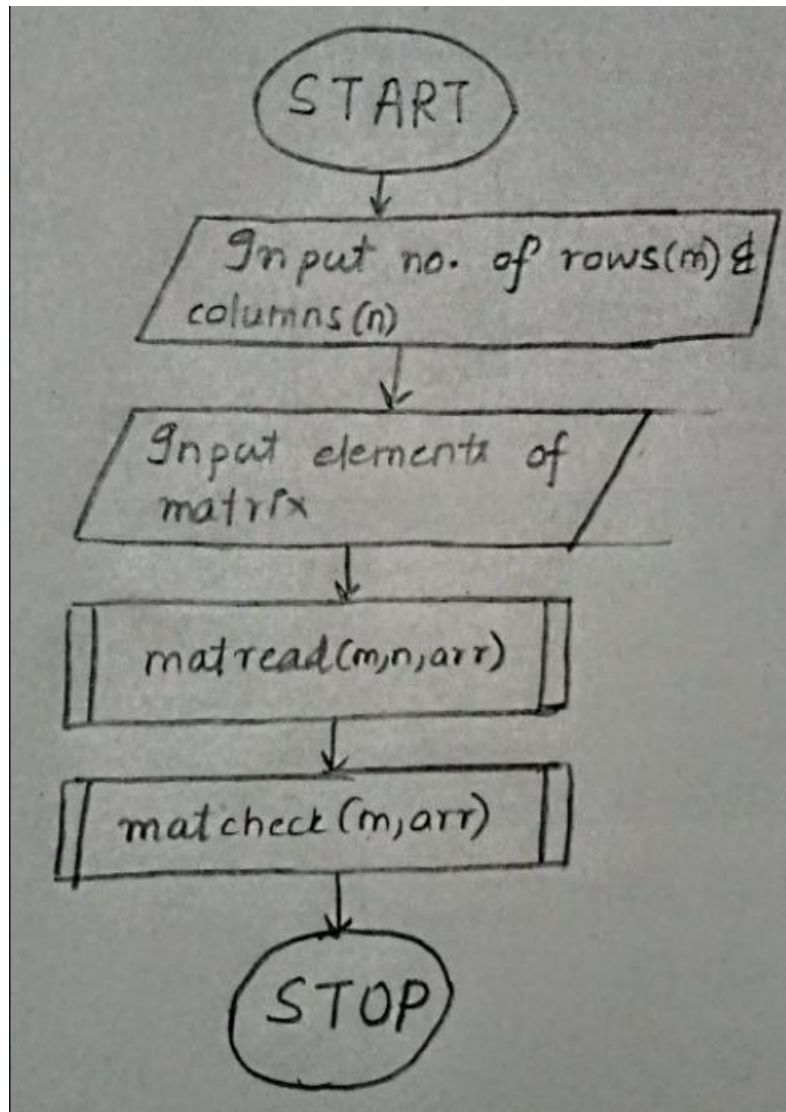


Function void matprint(int m, int n, int mat[][n])



Function void matcheck(int n, int arr[][n])





PROGRAM:

```
#include <stdio.h>
void matread(int m, int n, int mat[][n])
{
    for (int i = 0; i < m; i++)
        for (int j = 0; j < n; j++)
            scanf("%d", &mat[i][j]);
}
void matprint(int m, int n, int mat[][n])
{
    for (int i = 0; i < m; i++)
    {
        for (int j = 0; j < n; j++)
            printf("%d ", mat[i][j]);
        printf("\n");
    }
}
```

```

    }
}
void matchcheck(int n, int arr[][n])
{
    int row, col, up;
    up = 1;
    for (row = 0; row < n; row++)
    {
        for (col = 0; col < n; col++)
        {
            if (col < row && arr[row][col] != 0)
            {
                up = 0;
            }
        }
    }
    if (up == 1)
    {
        printf("\nThe entered matrix is a Upper triangular matrix.\n");
        printf("The matrix is printed below \n");
        matprint(n,n,arr);

    }
    else
    {
        printf("\nThis is Not a Upper triangular matrix.");
    }
}

int main()
{
    int m, n;
    printf("Enter the number of rows and columns \n");
    scanf("%d %d", &m, &n);
    int arr[m][n];
    printf("Enter elements of matrix\n");
    matread(m, n, arr);
    matchcheck(m, arr);

    return 0;
}

```

RESULT:	<pre>Enter the number of rows and columns 3 3 Enter elements of matrix 3 3 5 7 4 5 6 7 9 This is Not a Upper triangular matrix. PS C:\Users\aspur\C PROGRAMS\mydirectory></pre>
CONCLUSION:	We learnt about the 2-D arrays in the above experiment, we learnt about their functions and operations on matrix like matrix addition, subtraction, multiplication etc. and we learnt to use them in various problems.