

Name:	Vineet Parmar
UID no.	2021300092
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<n 26. I++ 27. Repeat 22,23,24,25 and 26 till i<m 28. Define void function multiply with 2 2D float array mat1[m][n] and mat2[a][b] as parameters 29. Intialize 2D array mat3 30. Call function zero(m,b,mat3) 31. I=0 32. J=0 33. K=0 34. mat3[i][j] += mat1[i][k]*mat2[k][j] 35. k++ 36. repeat 34 and 35 till k<n 37. j++ 38. repeat 33, 34, 35, 36 and 37 till j<b 39. i++ 40. repeat 32, 33, 34, 35, 36, 37, 38 and 39 till i<m 41. call function print(m,b,mat3) 42. Define void function transpose with a 2D float array mat[m][n] as parameter 43. Initialize 2D array newmat of dimension n x m 44. I=0 45. J=0 46. Newmat[i][j]=mat[j][i] 47. J++ 48. Repeat 46 and 47 till j<m 49. I++ 50. Repeat 45, 46, 47, 48 and 49 till i<n 51. Call function print(m,b,newmat) 52. Define int function norm with a 2D float array mat[m][n] 53. Initialize sum = 0.00 54. I=0 55. J=0 56. Sum += square of mat[i][j] 57. J++ 58. Repeat 54 and 55 till j<n 59. I++
--	---

	60. Repeat 55, 56, 57, 58 and 59 till $i < m$ 61. Sum = square root of sum 62. Return sum 63. Define integer main function 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 $(m=a \text{ and } b=n)$ call function add(m,n,mat1,a,b,mat2) else print Addition not possible 69. If $(m=a \text{ and } b=n)$ call function sub(m,n,mat1,a,b,mat2) else print subtraction not possible 70. If $(n=a)$ call function multiplication(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 norm(m,n,mat1) 74. Print value of function norm(a,b,mat2) 75. Return 0 76. STOP
PROGRAM:	<pre> #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++) </pre>

```

        printf("%.2ft",mat[i][j]);
        printf("\n");
    }
}
void add(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("%.2ft",mat1[i][j]+mat2[i][j]);
        printf("\n");
    }
}
void sub(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("%.2ft",mat1[i][j]-mat2[i][j]);
        printf("\n");
    }
}
void multiplication(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 transpose(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 norm(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)
        add(m,n,mat1,a,b,mat2);
    else
        printf("Addition not possible");
    printf("\n Subtraction of Matrices:\n");
    if(m==a && n==b)
        sub(m,n,mat1,a,b,mat2);
}

```

```

else
    printf("Subtraction not possible");
printf("\n Multiplication of Matrices:\n");
if(n==a)
    multiplication(m,n,mat1,a,b,mat2);
else
    printf("Multiplication not possible");
printf("\nTranspose of the 2 Matrices:\n");
transpose(m,n,mat1);
printf("\n");
transpose(a,b,mat2);
printf("Norm of Matrix 1 is: %.2f\n",norm(m,n,mat1));
printf("Norm of Matrix 2 is: %.2f",norm(a,b,mat2));
return 0;
}

```

```

Enter dimensions of Matrix 1:
2 2
Enter elements of Matrix 1:
1
2
3
4
1.00    2.00
3.00    4.00
Enter dimensions of Matrix 2:
2 2
Enter elements of Matrix 2:
5
7
1
3
5.00    7.00
1.00    3.00

Addition of Matrices:
6.00    9.00
4.00    7.00

Subtraction of Matrices:
-4.00   -5.00
2.00    1.00

Multiplication of Matrices:
7.00    13.00
19.00   33.00

Transpose of the 2 Matrices:
1.00    3.00
2.00    4.00

5.00    1.00
7.00    3.00

Norm of Matrix 1 is: 5.48Norm of Matrix 2 is: 9.17

...Program finished with exit code 0
Press ENTER to exit console.

```

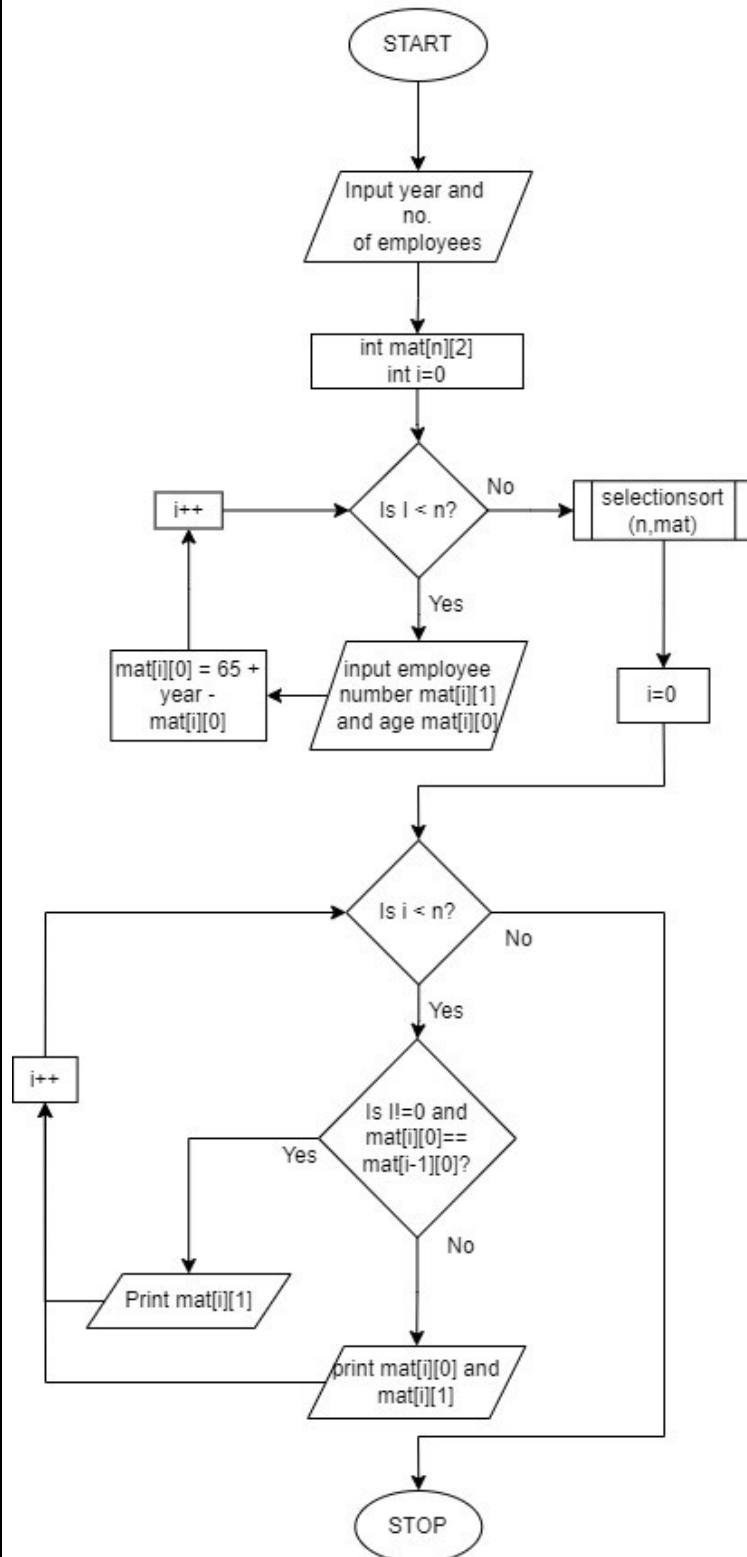
RESULT:

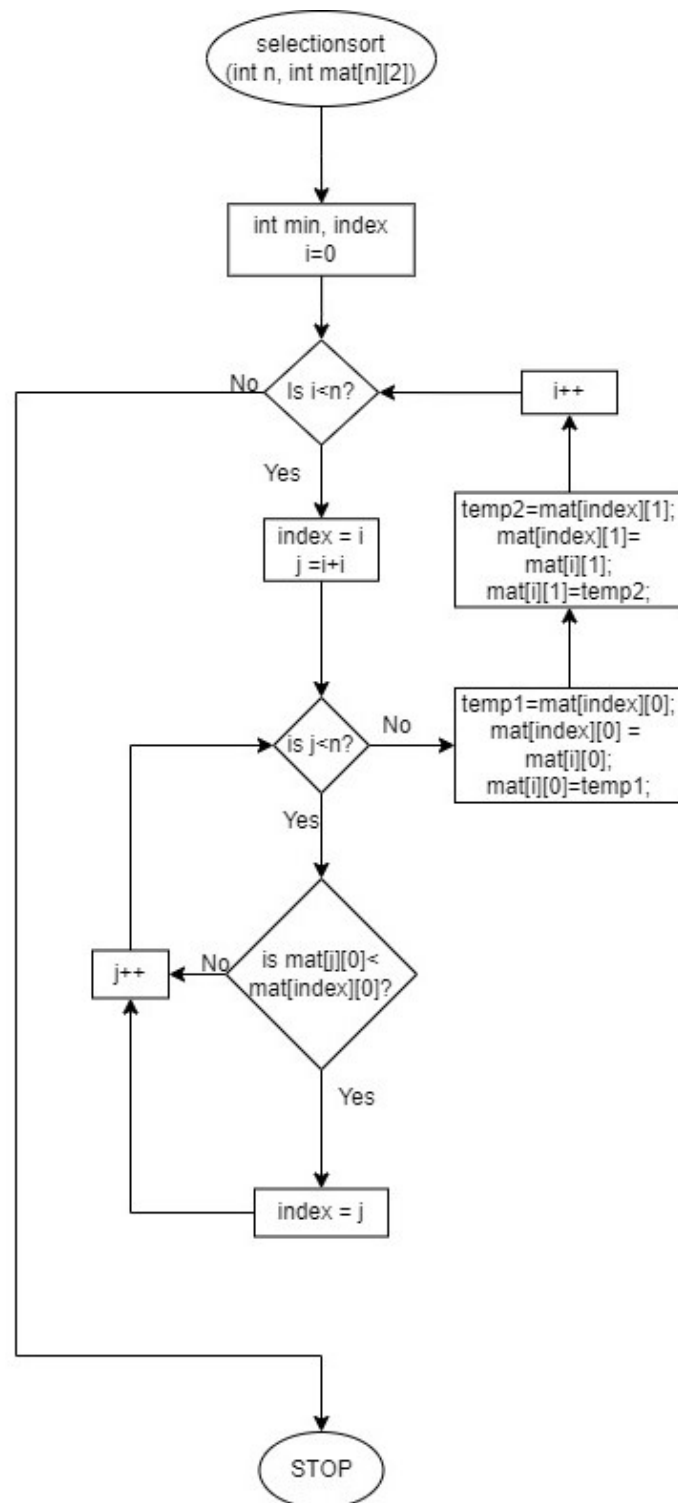
Program 2

PROBLEM STATEMENT :

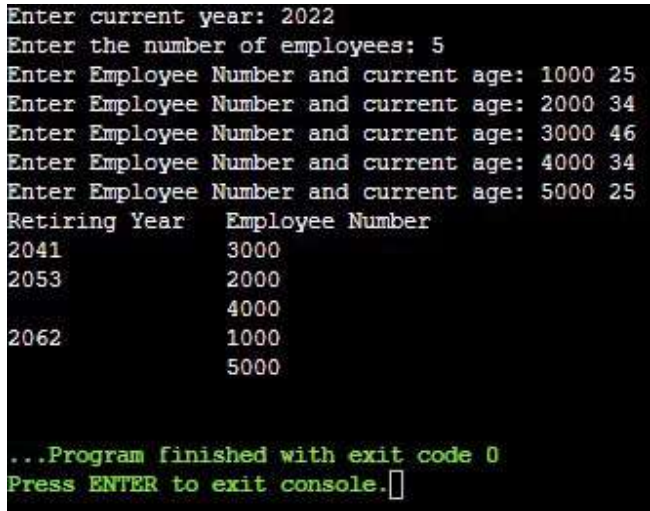
Write a program which reads the current year followed by N followed by a list of N employee numbers and their current ages. Produce a list showing the years in which the employees retire (become 65 years old). If more than one employee retires in a given year then include them all under the same heading.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

	<div>Year Number</div> <div>1986 896743</div> <div>1988 674501</div> <div> 450926</div>
ALGORITHM:	<ol style="list-style-type: none"> 1. START 2. Define void function selection sort with an 2D integer array mat[n][2] 3. Define integer variables min, index 4. l=0 5. Index = i 6. J=l+1 7. If(mat[j][0] < mat[index][0]) <div>index = j</div> 8. J++ 9. Repeat 7 and 8 till j<n 10. Initialize temp1 to mat[index][0] 11. Mat[index][0] = mat[i][0] 12. Mat[i][0] = temp1 13. Initialize temp2 to mat[index][1] 14. Mat[index][1] = mat[i][1] 15. Mat[i][1] = temp1 16. l++ 17. Repeat steps 5 to 16 till i<n-1 18. Define integer main function 19. Input current year year 20. Input the number of employees n 21. l=0 22. Input current age mat[i][0] and employee number mat[i][1] 23. Mat[i][0] = year + 65 – mat[i][0] 24. Call function selection sort(n,mat) 25. l=0 26. If(l not equal to 0 and mat[i][0]=mat[i-1][0]) <div>print Tabspace mat[i][1]</div> <div>else</div> <div>print mat[i][0] Tabspace mat[i][1]</div> 27. Return 0 28. STOP

FLOWCHART:



PROGRAM:	<pre> #include<stdio.h> void selectionsort(int n,int mat[n][2]) { int min,index; for(int i=0;i<n-1;i++) { index=i; for(int j=i+1;j<n;j++) { if(mat[j][0]<mat[index][0]) { index=j; } } int temp1=mat[index][0]; mat[index][0]=mat[i][0]; mat[i][0]=temp1; int temp2=mat[index][1]; mat[index][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"); for(int i=0;i<n;i++) { </pre>
-----------------	--

	<pre> if(i!=0 && mat[i][0]==mat[i-1][0]) printf("\t%d\n",mat[i][1]); else printf("%d\t%d\n",mat[i][0],mat[i][1]); } return 0; } </pre>
<p>RESULT:</p>	 <pre> Enter current year: 2022 Enter the number of employees: 5 Enter Employee Number and current age: 1000 25 Enter Employee Number and current age: 2000 34 Enter Employee Number and current age: 3000 46 Enter Employee Number and current age: 4000 34 Enter Employee Number and current age: 5000 25 Retiring Year Employee Number 2041 3000 2053 2000 4000 2062 1000 5000 ...Program finished with exit code 0 Press ENTER to exit console. </pre>
<p>CONCLUSION:</p>	<p>In these experiment, how to create and perform operations on 2D arrays. We learned how to code basic operations on Matrices like Addition, Subtraction, Multiplication and Transpose. We learned how we can sort entire rows of an 2D array and lastly how use of 2D matrices helps solving graphical problems easily.</p>