**Theory**

The two-dimensional array can be defined as an array of arrays. The 2D array is organized as matrices which can be represented as the collection of rows and columns. However, 2D arrays are created to implement a relational database lookalike data structure.

**Syntax:**

data\_type array\_name[rows][columns];

**(\*) String**

A string in C is merely an array of characters. The length of a string is determined by a terminating null character: '\0'.

* **<String.h> header file**

string.h is the header in the C standard library for the C programming language which contains macro definitions, constants and declarations of functions and types used not only for string handling but also various memory handling functions; the name is thus something of a misnomer. Functions declared in string.

[**strlen**](https://www.google.com/url?q=https://www.google.com/url?q%3Dhttps://www.codesdope.com/blog/article/predefined-string-functions-of-c-in-stringh-librar/%2523strlen%26amp;sa%3DD%26amp;ust%3D1583515717212000&sa=D&ust=1583515717246000&usg=AFQjCNFG7iFVU37z00-O2PAHEP6gZlsB_Q) :calculates the length of string

[**strcat**](https://www.google.com/url?q=https://www.google.com/url?q%3Dhttps://www.codesdope.com/blog/article/predefined-string-functions-of-c-in-stringh-librar/%2523strcat%26amp;sa%3DD%26amp;ust%3D1583515717213000&sa=D&ust=1583515717247000&usg=AFQjCNFLWpdfNCerkuwaQSG4_u75Rd0kbA)**:** String concatenation

[**strcpy**](https://www.google.com/url?q=https://www.google.com/url?q%3Dhttps://www.codesdope.com/blog/article/predefined-string-functions-of-c-in-stringh-librar/%2523strcpy%26amp;sa%3DD%26amp;ust%3D1583515717214000&sa=D&ust=1583515717247000&usg=AFQjCNHw8tYFtFzfDynPe_fBhBpmubY2hA) :Copies a string into another

[**strcmp**](https://www.google.com/url?q=https://www.google.com/url?q%3Dhttps://www.codesdope.com/blog/article/predefined-string-functions-of-c-in-stringh-librar/%2523strcmp%26amp;sa%3DD%26amp;ust%3D1583515717215000&sa=D&ust=1583515717247000&usg=AFQjCNHLH1v4ofOsVIrhOQdsDj0yqR4MZg): Compares two strings

**Question 1**

**Write a C program to find the sum of square of diagonal matrix of a square matrix.**

*Algorithm:*

Step 1: Start

Step 2: Enter matrix elements for 3X3 matrix and assign it to A[3][3]

Step 3: Let us declare two integer variable i,j

             And sum= 0

Step 4: for(i=0;i<3;i++)

            {

                 For(j=0;j<3;j++)

                  {

                      If(i==j)

                          Sum = sum + (A[i][j]\*A[i][j])

                    }

                  }

Step 5: Display sum

Step 6: End

*Program Code:*

#include<stdio.h>

int main()

{

    int A[3][3];

    int i,j,sum=0;

    printf("Enter matrix elements:\n");

    for(i=0;i<3;i++)

    {

        for(j=0;j<3;j++)

        {

        scanf("%d", &A[i][j]);

    }

    }

    for(i=0;i<3;i++)

    {

        for(j=0;j<3;j++)

        {

            if(i==j)

            {

                sum = sum +(A[i][j]\*A[i][j]);

            }

        }

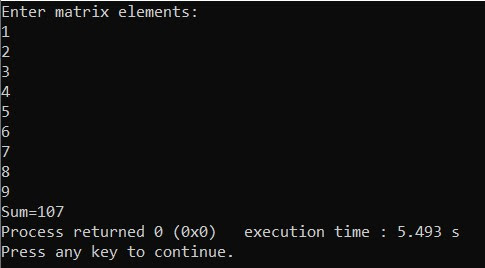
    }

    printf("Sum=%d", sum);

    return 0;

}

*Output:*



**Question 2**

**Write a C program to find the transpose of 2 by 3 matrix.**

*Algorithm:*

Step 1: Start

Step 2: Enter the value for 2 by 3 matrix in A[2][3];

Step 3: Declare i,j and B[3][2]

Step 4: for(i=0;i<3;i++)

              {

                     For(j=0;j<2;j++)

                       {

                          B[i][j]= A[j][i];

                            Display B[i][j];

                        }

                            Break line;

                     }

Step 5: End

*Program Code:*

#include<stdio.h>

int main()

{

    int A[2][3],i,j,B[3][2];

    printf("Enter elements for matrix A:\n");

    for(i=0;i<2;i++)

    {

        for(j=0;j<3;j++)

        {

            scanf("%d", &A[i][j]);

        }

    }

    printf("\n");

         for(i=0;i<2;i++)

    {

        for(j=0;j<3;j++)

        {

            printf("%d\t", A[i][j]);

        }

        printf("\n");

    }

    printf("\n");

    for(i=0;i<3;i++)

    {

        for(j=0;j<2;j++)

        {

            B[i][j]=A[j][i];

            printf("%d\t", B[i][j]);

        }

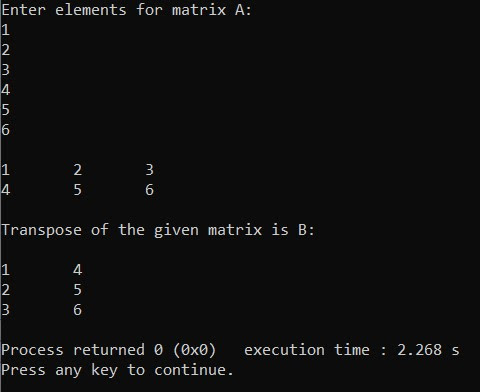
        printf("\n");

    }

    return 0;

    }

*Output:*



**Question 3**

**Write a C program to find the product of any two square matrix.**

*Algorithm:*

Step 1: Start

Step 2: Enter elements for two matrix A[3][3] and B[3][3]

Step 3: Declare variables i, j, k and C[3][3]

Step 4: for (i=0;i<3;i++)

{

for (j=0;j<3;j++)

{

    C[i][j]=0;

    for (k=0;k<3;k++)

    {

        C[i][j]=C[i][j]+(A[i][k]\*B[k][j])

    }

}

}

for (i=0;i<3;i++)

{

for (j=0;j<3;j++)

{

Display C[i][j])

}

Break line;

}

Step 5: End

*Program Code:*

#include <stdio.h>

int main()

{

int A[3][3],B[3][3] , C[3][3];

int i,j,k;

printf("Enter the matrix element for matrix A:\n");

for (i=0;i<3;i++)

{

for (j=0;j<3;j++)

{

scanf("%d",&A[i][j]);

}

}

printf("Enter the matrix element for matrix B:\n");

for (i=0;i<3;i++)

{

for (j=0;j<3;j++)

{

scanf("%d",&B[i][j]);

}

}

for (i=0;i<3;i++)

{

for (j=0;j<3;j++)

{

printf("%d\t",A[i][j]);

}

printf("\n");

}

printf("\n");

for (i=0;i<3;i++)

{

for (j=0;j<3;j++)

{

printf("%d\t",B[i][j]);

}

printf("\n");

}

printf("\n");

for (i=0;i<3;i++)

{

for (j=0;j<3;j++)

{

    C[i][j]=0;

    for (k=0;k<3;k++)

    {

        C[i][j]=C[i][j]+(A[i][k]\*B[k][j]);

    }

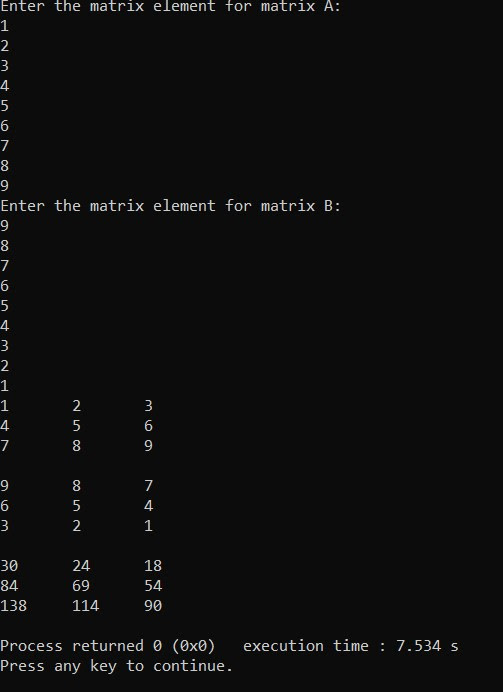
}

}

for (i=0;i<3;i++)

{

for (j=0;j<3;j++)

{

printf("%d\t",C[i][j]);

}

printf("\n");

}

return 0;

}

*Output:*

**Question 4**

**Write a C program to find whether the entered string is a palindrome or not.**

*Algorithm:*

Step 1: Start

Step 2: Enter a string and assign it to A[30]

Step 3: strcpy(B,A)

Step 4: if(strcmp(strrev(b),A==0)

             Display “Palindrome”

             Else

               Display “Not palindrome”

Step 5: End

*Program Code:*

#include <stdio.h>

#include<string.h>

int main()

{

char A[30],B[30];

printf("Enter Any string\n");

gets(A);

strcpy(B,A);

strrev(B);

if (strcmp(A,B)==0)

    printf("\n  palindrome");

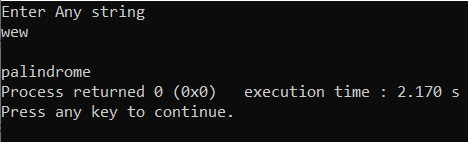
else

    printf("Not Palindrome");

return 0;

**}**

*Output***:**



**Question  5**

**Write a C program to display a string by removing vowels from it.**

*Algorithm*

Step 1: Start

Step 2: Enter a string and assign it to A[30]

Step 3: let i = 0

          l = strlen(A)

Step 4: While (i<l)

          {

               If(A[i]= {a,e,i,o,u})

                   i++

                 else

                  {

                     Display A[i]

                      i++

                     }

Step 5: End

*Program Code*

#include <stdio.h>

#include<string.h>

int main()

{

char A[30];

printf("Enter Any string\n");

gets(A);

int l=strlen(A),i=0;

 printf("Word without the vowel:\n");

while (i<l)

{

    if(A[i]=='A'||A[i]=='a'||A[i]=='E'||A[i]=='e'||A[i]=='I'|| A[i]=='i'||A[i]=='O'||A[i]=='o'|| A[i]=='u'||A[i]=='U')

    {

    i++;

    }

    else

    {

    printf("%c",A[i]);

    i++;

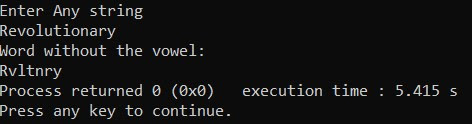
    }

}

return 0;

}

*Output*



**Question 6**

**Write a C program to sort strings.**

*Program Code*

#include <stdio.h>

#include<string.h>

int main()

{

char A[10][10] ,temp[10];

int i,j;

printf("Enter Any string\n");

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

{

gets(A[i]);

}

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

{

for (j=i+1;j<5;j++)

{

 if (strcmp(A[i],A[j])>0)

    {

    strcpy(temp,A[i]);

    strcpy(A[i],A[j]);

    strcpy(A[j],temp);

}

}

}

printf("The sorted Element are:\n");

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

{

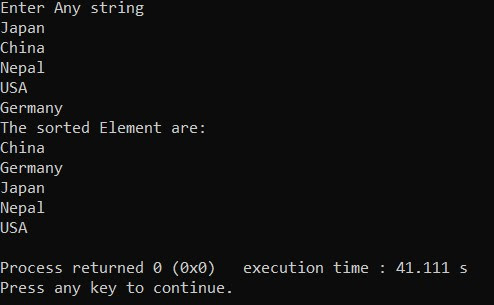
printf("%s\n",A[i]);

}

    return 0;

}

*Output*



**CONCLUSION:**

                After completing this lab work, we are now able to implement the function of 2-dimensional array for matrix related problems such as multiplication, addition and many others. We were also made familiar with the string functions and use them for solving string related problems.