- **Project Idea summary**

The program is a matrix calculator, which performs operations on two matrices, such as addition, subtraction, multiplication, division, finding the transpose of the matrix, and also multiplying a matrix by an integer scalar. So, the user can select any option they want in the calculations.

Source Code Of The Project

```
#include <stdio.h>
#define size 100 /* maximum size of array */
// function prototype:
void ReadMatrices(int x[size][size],int k,int l );// function to read the elements of
the array from the user
void AddMatrices(int x[size][size],int y[size][size],int k,int l); //function to do
addition operation of two arrays
void SubMatrices(int x[size][size],int y[size][size],int k,int l);//function to do
subtraction operation of two arrays
void MulMatrices(int x[size][size],int y[size][size],int k,int l,int e); //function to do
multiplication operation of two arrays
void DivMatrices(int x[size][size],int y[size][size],int k,int l); //function to do
division operation of two arrays
void TransMatrices(int x[size][size],int y[size][size],int k,int l); //function to do
transpose of the array
void ScalarmulMatrices(int x[size][size],int k,int l ,int s); //function to do scalar
multiplication operation of two arrays
// function main begins program execution :
int main()
{
  int a[size][size],b[size][size]; //declaration of array 1 & array 2
  int m,n,r,c,operation,scalar; //declaration of rows of array 1 & columns 1 &
rows 2 & columns 2 & operation & scalar respectively
  char again = 'Y'; // declaration variable again of type char
  printf("\tWelcome to Matrix Calculator\n\n");
```

```
    while (again == 'Y'||again=='y') //iteration condition which refer to asking the user if he wants to do another operation or no, 'Y' refer to yes
{
  printf("What is the operation that you want to do?\n");
  printf("\n Enter 1 to do addition operation");
  printf("\n Enter 2 to do subtraction operation");
  printf("\n Enter 3 to do multiplication operation");
  printf("\n Enter 4 to do division operation");
  printf("\n Enter 5 to do Transpose operation");
  printf("\n Enter 6 to do Scalar Multiplication operation\n");
  scanf("%d",&operation);// read which operation that the user want to do
  switch(operation)
    {
  case 1: // to do addition operation
      printf("\nPlease enter the rows and columns for the first matrix:\n");
      printf("\nNumber of Rows= ");
      scanf("%d",&m); // read the rows of array 1
      printf("\nNumber of Columns= ");
      scanf("%d",&n);// read the columns of array 1
      printf("\nPlease enter the rows and columns for the second matrix:\n");
      printf("\nNumber of Rows= ");
      scanf("%d",&r);// read the rows of array 2
      printf("\nNumber of Columns= ");
      scanf("%d",&c); // read the columns of array 2
    if((m!=r) || (n!=c)) // if rows 1 not equal rows 2 || columns 1 not columns 2 in this case it can't add the two arrays

    {

      printf("\nError!! Matrices must be the same size\n");

    }

    else

    {

       printf("\nPlease enter the elements of the first matrix:\n");

       ReadMatrices(a,m,n);  // function ReadMatrices call to read the elements of the array 1

       printf("\nPlease enter the elements of the second matrix:\n");

       ReadMatrices(b,m,n); // function ReadMatrices call to read the elements of the array 2

       AddMatrices(a,b,r,c); // function AddMatrices call to add the two arrays

    }

       break;

  case 2: // to do subtraction operation

      printf("\nPlease enter the rows and columns for the first matrix:\n");

      printf("\nNumber of Rows= ");

      scanf("%d",&m);// read the rows of array 1

      printf("\nNumber of Columns= ");

      scanf("%d",&n);// read the columns of array 1

      printf("\nPlease enter the rows and columns for the second matrix:\n");

      printf("\nNumber of Rows= ");

      scanf("%d",&r);// read the rows of array 2
```

```
        printf("\nNumber of Columns= ");

        scanf("%d",&c);// read the columns of array 2

      if((m!=r) || (n!=c))// if rows 1 not equal rows 2 || columns 1 not columns 2 in
this case it can't subtract the two arrays

      {

        printf("\nError!! Matrices must be the same size\n");

      }

      else

      {

         printf("\nPlease enter the elements of the first matrix:\n");

        ReadMatrices(a,m,n);// function ReadMatrices call to read the elements of
the array 1

        printf("\nPlease enter the elements of the second matrix:\n");

        ReadMatrices(b,m,n);// function ReadMatrices call to read the elements of
the array 2

        SubMatrices(a,b,r,c);// function SubMatrices call to subtract the two arrays

      }

         break;

   case 3: // to do multiplication operation

        printf("\nPlease enter the rows and columns for the first matrix:\n");

        printf("\nNumber of Rows= ");

        scanf("%d",&m);// read the rows of array 1

        printf("\nNumber of Columns= ");

        scanf("%d",&n);// read the columns of array 1

        printf("\nPlease enter the rows and columns for the second matrix:\n");
        printf("\nNumber of Rows= ");
        scanf("%d",&r);// read the rows of array 2
        printf("\nNumber of Columns= ");
        scanf("%d",&c);// read the columns of array 2
      if(n!=r)// the condition when can't multiply two arrays
      {
        printf("\nError!! column of the first matrix not equal to row of the second\n");
      }
      else
      {
         printf("\nPlease enter the elements of the first matrix:\n");
         ReadMatrices(a,m,n);// function ReadMatrices call to read the elements of the array 1
         printf("\nPlease enter the elements of the second matrix:\n");
         ReadMatrices(b,r,c);// function ReadMatrices call to read the elements of the array 2
         MulMatrices(a,b,m,n,c);// function MulMatrices call to multiply the two arrays
      }
         break;
    case 4: // to do division operation
        printf("\nPlease enter the rows and columns for the first matrix:\n");
        printf("\nNumber of Rows= ");
        scanf("%d",&m);// read the rows of array 1
        printf("\nNumber of Columns= ");
        scanf("%d",&n);// read the columns of array 1
        printf("\nPlease enter the rows and columns for the second matrix:\n");
        printf("\nNumber of Rows= ");
        scanf("%d",&r);// read the rows of array 2
        printf("\nNumber of Columns= ");
        scanf("%d",&c);// read the columns of array 2
      if((m!=r) || (n!=c))// if rows 1 not equal rows 2 || columns 1 not columns 2 in this case it can't divide the two arrays
      {
        printf("\nError!! Matrices must be the same size\n");
      }
      else
      {
        printf("\nPlease enter the elements of the first matrix:\n");
         ReadMatrices(a,m,n);// function ReadMatrices call to read the elements of the array 1
         printf("\nPlease enter the elements of the second matrix:\n");
         ReadMatrices(b,r,c);// function ReadMatrices call to read the elements of the array 2
         DivMatrices(a,b,m,n);// function DivMatrices call to divide the two arrays
      }
         break;
   case 5: //to do transpose operation
        printf("\nPlease enter the rows and columns for the matrix:\n");
        printf("\nNumber of Rows= ");
        scanf("%d",&m);// read the rows of array 1
        printf("\nNumber of Columns= ");
        scanf("%d",&n);// read the columns of array 1
        printf("\nPlease enter the elements of the matrix:\n");
        ReadMatrices(a,m,n);// function ReadMatrices call to read the elements of the array
        TransMatrices(a,b,m,n);// function TransMatrices call to transpose the array
        break;
    case 6: //to scalar multiplication operation
        printf("\nPlease enter the rows and columns for the matrix:\n");
        printf("\nNumber of Rows= ");
        scanf("%d",&m);// read the rows of array 1
        printf("\nNumber of Columns= ");
        scanf("%d",&n);// read the columns of array 1
        printf("\nPlease enter the elements of the matrix:\n");
        ReadMatrices(a,m,n);// function ReadMatrices call to read the elements of the array
        printf("\nPlease enter a number to multiply with matrix:\n");
        scanf("%d",&scalar);// read the scalar number
        ScalarmulMatrices(a,m,n,scalar);// function ScalarmulMatrices call to multiply a scalar number in the array
        break;
    default: // if user enter a number not in the range of 1-6
        printf("\nIncorrect option! Please choose a number 1-6.");
        break;
```

```
  printf("\n-------------------------------------------------------------------------------------------
---------\n");
   printf("\nDo you want to calculate again? Y/N\n");
   scanf(" %c", &again);
   if(again == 'N'||again=='n') // the condition if user wants to stop do operations ,
'N' refer to no
  {
     printf("\n\nGoodbye!\n\n");
  }
}
  return 0;
}// end function main
void ReadMatrices(int x[size][size],int k,int l ) // function ReadMatrices to read
elements of the array
{
   for(int i=0;i<k;i++)
   {
      for(int j=0;j<l;j++)
      {
         scanf("%d",&x[i][j]);
      }
   }
}
void AddMatrices(int x[size][size],int y[size][size],int k,int l)  // function
AddMatrices to execute addition operation of two matrices
{
    printf("The Sum of the two matrix is:\n");
    for(int i=0;i<k;i++)
    {
        for(int j=0;j<l;j++)
        {
            printf("%d \t",x[i][j]+y[i][j]);
        }
        printf("\n");
    }
}
void SubMatrices(int x[size][size],int y[size][size],int k,int l)// function SubMatrices to execute subtraction operation of two matrices
{
    printf("The Subtract of the two matrix is:\n");
    for(int i=0;i<k;i++)
    {
        for(int j=0;j<l;j++)
        {
            printf("%d \t",x[i][j]-y[i][j]);
        }
        printf("\n");
    }
}
void MulMatrices(int x[size][size],int y[size][size],int k,int l,int e) // function MulMatrices to execute multiplication operation of two matrices
```

```
{
   int mul[size][size];
   for(int i=0;i<k;i++)
   {
      for(int j=0;j<e;j++)
      {
         mul[i][j]=0;
      }
   }
   for(int i=0;i<k;i++)
   {
      for(int j=0;j<e;j++)
      {
         for(int t=0;t<l;t++)
         {
            mul[i][j]+=x[i][t]*y[t][j];
         }
      }
   }
  printf("The Multiplication of the two matrix is:\n");
  for(int i=0;i<k;i++)
   {
      for(int j=0;j<e;j++)
      {
```

```
      printf("%d \t",mul[i][j]);
    }
    printf("\n");
  }
}
void DivMatrices(int x[size][size],int y[size][size],int k,int l)//function DivMatrices to execute division operation of two matrices
{
   printf("The Division of the two matrix is:\n");
  for(int i=0;i<k;i++)
  {
    for(int j=0;j<l;j++)
    {
      printf("%d \t",x[i][j]/y[i][j]);
    }
    printf("\n");
  }
}
void TransMatrices(int x[size][size],int y[size][size],int k,int l)//function TransMatrices to execute the transpose operation of the array
{
  for(int i=0;i<k;i++)
  {
    for(int j=0;j<l;j++)
    {
      y[j][i] =x[i][j];
    }
  }
  printf("The Transpose of the matrix is:\n");
  for(int i=0;i<l;i++)
  {
    for(int j=0;j<k;j++)
    {
      printf("%d \t",y[i][j]);
    }
    printf("\n");
  }
}
void ScalarmulMatrices(int x[size][size],int k,int l ,int s)//function
ScalarmulMatrices to execute the scalar multiplication operation
{
  printf("The result of a scalar matrix multiplication is:\n");
  for(int i=0;i<k;i++)
  {
    for(int j=0;j<l;j++)
    {
      printf("%d \t",x[i][j]*s);
    }
    printf("\n");   } }
```