GALAXY INTERNATIONAL INTERDISCIPLINARY RESEARCH JOURNAL (GIIRJ)
ISSN (E): 2347-6915
Vol. 12, Issue 4, April (2024)

THE APPLICATION OF LINEAR ALGEBRA TO THE PROGRAMMING LANGUAGES C# AND C++

G‘ayniddinov Shayxislom Tolibjon o‘g‘li,
Namangan State University, Assistant at the
Department of “Algebra and Methods of Teaching Mathematics”

Daughter of Zokirova Sevarakhan Zafarbek,
Ismonjanova Sumbula Sherzodbek daughter,
Karimjonova Mohinabonu Ilhomjon daughter ,
Students of the Department of Applied Mathematics and
Digital Technology, Faculty of Mathematics, Namangan State University

ABSTRACT

Through this article we will address some issues of this discipline through the C# and C++ programming languages of linear algebra .

Keywords: matrix , array, mixed arrays, multidimensional array, determinant, inverse Matrix, second order straight lines, and program.

INTRODUCTION

In the language of the software comes full to use in multiple uses of a different assistant to the memory of the computer. The data given to perform in practice in the first place with one variable must belong to one type. In programming languages, the data in a list or table view is called an array. The meaning of the mass word means size, measure. All elements of the array belong to one type, they are named by one name and differ from each other.

A complex of ordered data with a single name, which belongs to one species, says the public.

C # gives us 3 different types of arrays for dating:

One-dimensional array. It contains only one hard data, so it enters the element stored in the array with a single-digit index (0, 1, 2, and hokazo).

Multidimensional array. Contains several data rows, so its index will consist of even numbers that determine the row and determine the column. Such an array Street is called a rectangular array because it takes the shape of a rectangle when described schematically.

The ridge is massive. It is a shallow array from a small array(and these small arrays can be of any size).

In the program, the Assistant works a lot on one-dimensional arrays, in multidimensional arrays it does not work at all. Multidimensional is an array of two or more dimensions, and each element of such an array can be accessed with the help of a combination of the meaning of two or more indices.

Is a simple two-dimensional series of a multidimensional array. The position of any element in a two-dimensional array is indicated by two indices. Such an array can be seen in the table language, the lines of which are indicated by the bittain index, the columns of which are indicated by the bittain index.

Below is a 10x20 scale;

int [,] table = new INT[10, 20];

Pay attention to the announcement of this array. Both of its dimensions are separated by a comma. The first part of this announcement is the syntax:

[,] a variable to a two-dimensional array implies an appeal. If memory is allocated for the array using the new operator, then the following syntax applies.

int[10, 20]

This announcement creates a 10x20-dimensional array, but in this case its dimensions are separated by commas.

To access a two-dimensional array element, it is necessary to separate both indices with commas. For example, an element of a table array with indexes (3,5) in the following line is valued at 10.

table[3, 5] = 10;

An example of launching into a 2×3×4-dimensional integer three-dimensional array:

```
namespace Matrix
{
 internal class Program
 {
 static void Main (string [] args
 {
 short [,,] B = {
 {
 { 1, 2, 3, 4 },
 { 5, 6, 7, 8 },
 { 9, 10, 11, 12 }
 },
 {
 { 13, 14, 15, 16 },
 { 17, 18, 19, 20 },
 { 21, 22, 23, 24 }
 }
 };
 for (int i = 0; i < 2; I++
 {
 for (Int j = 0; j < 3; j++
 {
 for (int k = 0; k < 4; k++
 {
 Console.Write (B[i, j, k] + " ");
 }
 Console.WriteLine();
 }
 Console.WriteLine();
 }
```

```
 Console.ReadKey()may refer to;
 }
 }
}
```

The result comes out as follows:

```
 1 2 3 4
 5 6 7 8
 9 10 11 12
 13 14 15 16
 17 18 19 20
 21 22 23 24
```

Arrays are characterized by such a concept as rank or number of measurements. Arrays can be 1, 2, 3, or N-dimensional. Looking at multi-dimensional arrays using the example of 2-dimensional arrays we can imagine one-dimensional arrays as a single horizontal array, for example:

```
 int [] nums1 = new int[] { 0, 1, 2, 3, 4 };
```

Visual appearance of this array

```
0
 1
 2
 3
 4
```

Now let's take an example of a 2-dimensional array:

```
 int[,] nums2 = { { 0, 1, 2 }, { 3, 4, 5 } };
```

Visually, this array:

```
0
1
2
3
4
 5
```

To declare multidimensional arrays, a comma is placed inside a square bracket, i.e. [,] a two-dimensional array, [,,] a three-dimensional array, etc.

```
int [,] arr2d; / / two-dimensional array
int [,,] arr3d; / / three-dimensional array
int [,,,] arr4d; / / four-dimensional array
int[,,,,] arr5d; / / five-dimensional array
```

In C++ algorithmic language, not only with one-dimensional arrays, but many dimensional arrays can also be worked with. If the array in turn is again consisting of an array, it means a two-dimensional array, that is, called a matrix. The measurement of arrays does not prevent them from working on the computer, since they are stored in memory as linear sequential elements. Multidimensional arrays are declared as if they were a 1-dimensional array, except that the index category as the category of rows (rows) and columns of the array is displayed,

and they are in separate [] [] brackets. For example: 2 consisting of integers named A given a dimensional array and the number of rows is n, the number of columns is m: int a[n] [m Input-output two-dimensional array elements, perform actions on them is within the parametric cycles in which ichma-ich is located, i.e. Cycle 1 for the rows, 2- for cycle columns. For example:

for (i=0; i<=3; I++
for (j=0; j<=3; j++
China > > a[i] [c];

If you need to enter them from the keyboard, that is, using the cin operator when established, it is entered as follows:

1 2 3
4 5 6
7 8 9

It is also possible to declare the elements of the array, as well as make them initialization can also be done:

int a[3][3] = {{1,2,3},{4,5,6},{7,8,9}};

To make the results look beautiful, the output operator is as follows it is necessary to organize:

for (int i=0; i<3; I++
{ for (Int j=0; j<3; j++
cout << < " a ["<<i<","<<j<<< < a[i][j]; < b="">
cout

Example:

Construct a program to find the inverse matrix to a given matrix a, given matrix A.

Algorithm of the issue :

1.First we calculate it i.e. given, leta the determinant of the Matrix ;
2.There is no inverse matrix to a given matrix if the determinant is zero i.e. detA=0;
3.Otherwise we can construct a transponerated matrix into a given matrix ;
4. And in the next step, we find its inverse matrix through this formula ;
5.And in this step, we write the program code using the above;
6.We analyze the program code;
7.We enter into the computer;
8.We analyze the result of the program we entered on the computer;

Program code in C# :

```
USING SYSTEM;
USING SYSTEM.LINQ;

NAMESPACE TESKARI_MATRITSA;

PUBLIC STATIC CLASS PROGRAM
```

```
{
 PUBLIC STATIC VOID MAIN()
 {DOUBLE A11,A12,A13,A21,A22,A23,A31,A32,A33,D,T,M;
 DOUBLE[,] A = NEW DOUBLE[3,3];
 DOUBLE[,] A = NEW DOUBLE[3,3];

 FOR(INT I = 0; I < 3; I++)
 {
        STRING[] S = CONSOLE.READLINE().SPLIT();
     FOR(INT J = 0; J < 3; J ++)
 {
        A[I,J]= DOUBLE.PARSE(S[J]);
     }
  }
 A11 = A[0,0];
   A12=A[0,1];
   A13= A[0,2];
   A21=A[1,0];
   A22=A[1,1];
   A23=A[1,2];
 A31=A[2,0];
   A32=A[2,1];
   A33=A[2,2];
 D=A11*A22*A33+A21*A32*A13+A31*A12*A23-A13*A22*A31-A23*A32*A11-
A33*A12*A21;
 IF(D==0)
 {CONSOLE.WRITELINE("USHBU MATRITSAGA TESKARI MATRITSA MAVJUD EMAS");}
 ELSE
 {
 A[0,0]=A22*A33-A23*A32;
 A[0,1]=(-(A21*A33-A23*A31));
 A[0,2]=A21*A32-A22*A31;
 A[1,0]=(-(A12*A33-A13*A32));
 A[1,1]=A11*A33-A13*A31;
 A[1,2]=(-(A11*A32-A12*A31));
 A[2,0]=A12*A23-A13*A22;
 A[2,1]=(-(A11*A23-A21*A13));
 A[2,2]=A11*A22-A21*A12;
FOR(INT I = 0; I < 3; I++)
 {
        //STRING[] S = CONSOLE.READLINE().SPLIT();
     FOR(INT J = 0; J < 3; J ++)
 {
```

```
        A[I,J]= A[J,I]/D;
      }
   }

 FOR(INT I = 0; I < 3; I++)
 {
        //STRING[] S = CONSOLE.READLINE().SPLIT();
      FOR(INT J = 0; J < 3; J ++)
 {
        CONSOLE.WRITE(A[I,J]/D+" ");
      }
  CONSOLE.WRITELINE();
   }

 CONSOLE.READKEY();

   }

 }
}
```

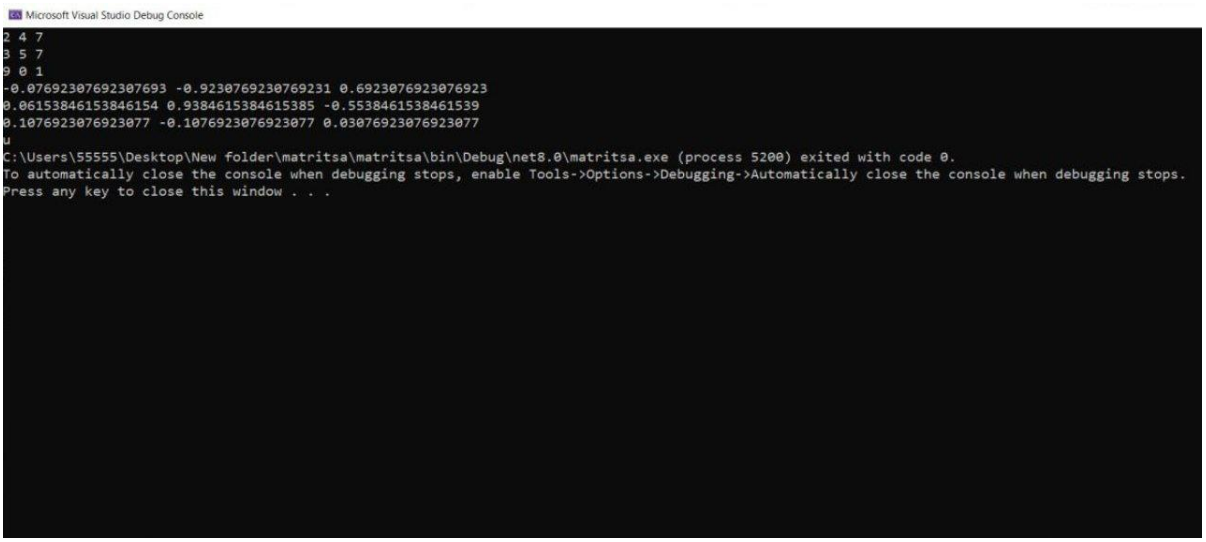
INCLUDED MATRIX:

2 4 7
3 5 7
9 0 1

WHICH IS THE INVERSE MATRIX OF THE INTRODUCED MATRIX:

-0.07692307692307693 -0.9230769230769231 0.6923076923076923
0.06153846153846154 0.9384615384615385 -0.5538461538461539
0.1076923076923077 -0.1076923076923077 0.03076923076923077

PROGRAM



RESULT

C++ program code:

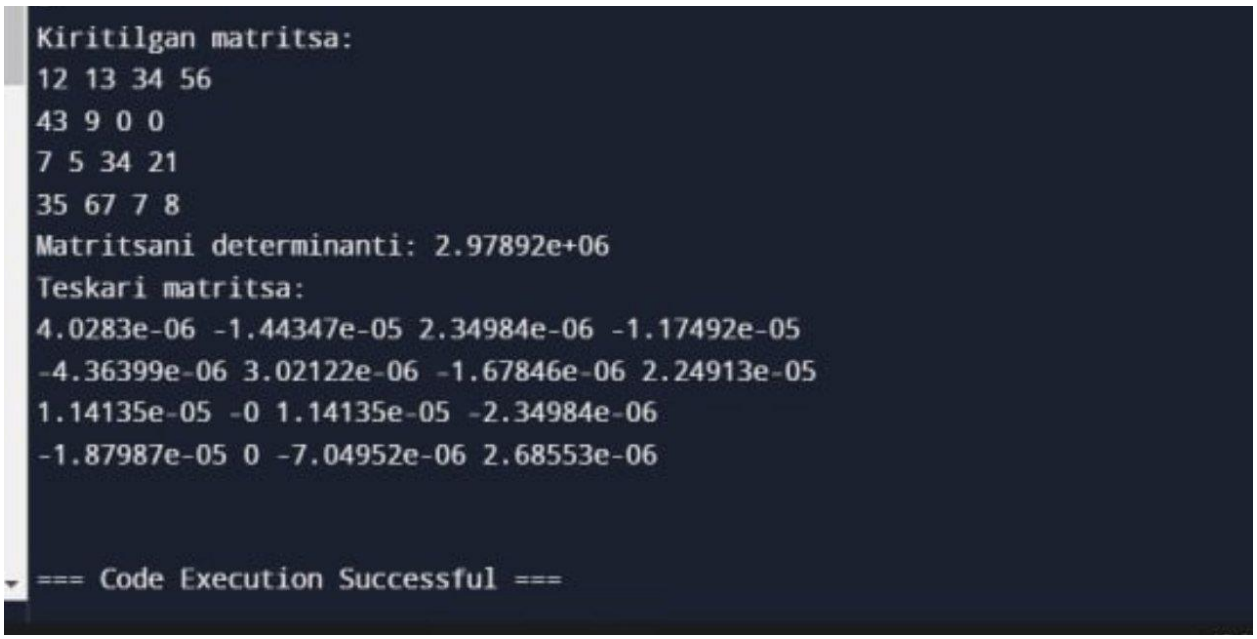
```
#include <iostream>
#include <vector>

// Matritsaning determinantini hisoblash funktsiyasi
int determinant(const std::vector<std::vector<double>>& mat) {
    int n = mat.size(); // Size of the square matrix

    // Base case: If the matrix is 1x1, determinant is the only element
    if (n == 1) {
        return mat[0][0];
    }

    double det = 0; // Matritsani Determinanti
    for (int j = 0; j < n; ++j) {
        std::vector<std::vector<double>> submat(n - 1, std::vector<double>(n - 1, 0));
        for (int p = 1; p < n; ++p) {
            int q = 0;
            for (int k = 0; k < n; ++k) {
                if (k != j) {
                    submat[p - 1][q++] = mat[p][k];
                }
            }
        }

          double subdet = determinant(submat);

        if (j % 2 == 0) {
            det += mat[0][j] * subdet;
        } else {
            det -= mat[0][j] * subdet;
        }
    }

    return det;
}

int main() {
    int k;
      int n; // Matritsa o'lchami (n x n)
    std::cout << "Matritsa o'lchamini kiriting: ";
```

```
    std::cin >> n;

    // Matritsani kiritish
    std::vector<std::vector<double>> mat(n, std::vector<double>(n));
    std::cout << "Matritsa elementlarini kiriting:\n";
    for (int i = 0; i < n; ++i) {
        for (int j = 0; j < n; ++j) {
            std::cin >> mat[i][j];
        }
    }

    // Kiritilgan matritsaning chiqarish
    std::cout << "Kiritilgan matritsa:\n";
    for (int i = 0; i < n; ++i) {
        for (int j = 0; j < n; ++j) {
            std::cout << mat[i][j] << " ";
        }
        std::cout << std::endl;
    }
    // determinantni hisoblash
    double det = determinant(mat);
    // Determinantni chiqarish
    std::cout << "Matritsani determinanti: " << det << std::endl;
std::cout << "Teskari matritsa: " << std::endl;
      k=1;
        for (int j = 0; j < n; ++j) {
            for (int i = 0; i < n; ++i) {
                if ((i+j)%2==1)
                {k=-1;

                }
                else {k=1;};
            std::cout << mat[i][j]*k/det << " ";
        }
        std::cout << std::endl;
    }

    return 0;
}
```

Included Matrix:
12 13 34 56
43 9 0 0

7 5 34 21
35 67 7 8
Determinant of the Matrix:2.97892e+06
Inverse Matrix:
4.0283e-06 -1.44347e-05 2.34984e-06 -1.17492e-05
-4.36399e-06 3.02122e-06 -1.67846e-06 2.24913e-05
1.14135e-05 -0 1.14135e-05 -2.34984e-06
-1.87987e-05 0 -7.04952e-06 2.68553e-06

Program result:



REFERENCES

1.N.A.Otaxanov “Algoritmik tillar va dasturlash bo`yicha masalalar to`plami” “Usmon Nosir media”,2023.-188b.
2. Tolibjon o‘g, G‘ayniddinov Shayxislom, and Polvanov Rashid Raximjanovich. "IKKINCHI TARTIBLI GRONUOLL CHEGARALANISHLI BOSHQARUVLAR UCHUN TUTISH MASALASI." *SCIENTIFIC APPROACH TO THE MODERN EDUCATION SYSTEM* 2.22 (2024): 209-214.
3. Xolmuradov, F. M. (2024). DIFFERENTSIAL TENGLAMALAR FANINI OQITISHDA KONPETENSIYAVIY VA ADAPTIV YONDASHUVLARDAN FOYDALANISH METOKASI. *Научный Фокус*, *1*(11), 172-178.
4. Б. Болтаев. Основы информатики и вычислительной техники. Ташкент: Издательство им. Чолпона, 2006.-125б [76-90].
5. Zafarbek, Zokirova Sevarakhon. "DEMONSTRATION OF A NETWORKABLE PROGRAMMING LANGUAGE IN MODERN PROGRAMMING LANGUAGES." *INNOVATIVE DEVELOPMENTS AND RESEARCH IN EDUCATION* 3.26 (2024): 95-101.6. Павловская Т.А. С++. Программирование на языке высокого уровня – СПб.: Питер. 2005.- 461 с.

7. Zafarbek Z. S. DEMONSTRATION OF A NETWORKABLE PROGRAMMING LANGUAGE IN MODERN PROGRAMMING LANGUAGES //INNOVATIVE DEVELOPMENTS AND RESEARCH IN EDUCATION. – 2024. – Т. 3. – №. 26. – С. 95-101.
8. Zafarbek, Z. S. (2024). DEMONSTRATION OF A NETWORKABLE PROGRAMMING LANGUAGE IN MODERN PROGRAMMING LANGUAGES. *INNOVATIVE DEVELOPMENTS AND RESEARCH IN EDUCATION*, *3*(26), 95-101.
9.Informatika va programmalash.O'quv qo'llanma. Mualliflar: A.A.Xaldjigitov, Sh.F.Madraximov, U.E.Adamboev, O'zMU, 2005 yil, 145 bet.
10.uzhttp://dasturchi.uz
11.uzdevs.