

**Міністерство освіти та науки України
Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”
Факультет прикладної математики
Кафедра системного програмування і спеціалізованих комп’ютерних
систем**

**ЛАБОРАТОРНА РОБОТА №3
з дисципліни
“Програмування”
Тема: «Опис масивів. Робота з масивами »**

**Виконав: Фесенко Д.О.
Студент групи КВ-34
Варіант №23**

Київ 2023

Постановка задачі

1. Заповнити квадратну матрицю $A=\{a_{ij}\}$, $a_{ij}=f(i,j)$ при $(i, j=0, \dots, n-1)$, де $f(i,j)$ – задана функція .
2. Взяти за вхідні дані елементи матриці A , за вказаною умовою отримати елементи вектора $X=\{x_0, x_1, \dots, x_{n-1}\}$.
3. Взявши як вхідні дані елементи вектора X , обчислити значення функції $U=g(x)$, де $g(x)$ – задана функція .

Вимоги до програми

1. Значення n вибирається довільно ($6 \leq n \leq 10$).
2. Функції $f(i,j)$, $g(x)$, а також умова отримання вектора X визначаються варіантом завдання.
3. Логічно відокремлені частини алгоритму (заповнення матриці, заповнення вектора, обчислення функції, виведення результатів, тощо) реалізувати за допомогою окремих функцій.
4. Як результат вивести початкову матрицю A , перетворену матрицю A (якщо така буде), елементи вектора X і значення U .
5. Додаткові масиви використовувати не дозволяється.

Завдання за варіантом №23

23.	$2e^j - 0.879i^2 + 13.2j$
-----	---------------------------

23. Транспонувати матрицю A відносно побічної діагоналі. Третій рядок транспонованої матриці взяти за вектор X .

23.	$\prod_{i=1}^{n-1} \left(\frac{1}{ x_i + 1} + x_{i+1} \right)$
-----	--

Код програми

```
#include <stdio.h>
#include <math.h>
#define size 6 //Compiler doesn't initialize the matrix with 'const', so...

void CreatMatrix(double Matrix[size][size], int n);
void TrnsposeMatrix(double Matrix[size][size], int n);
void PrintMatrix(double Matrix[size][size], int n);
void PrintVector(double Matrix[size], int n);
void FormatMatrix(double Matrix[size][size], double Vector[size], int n);
void DevisionFunc(double Vector[size], double* FunctionG, int n);

int main() {
    double m[size][size], x[size], g=1.0;

    CreatMatrix(&m, size);
    printf("-----|Filling matrix A:\n");
    PrintMatrix(&m, size);

    TrnsposeMatrix(&m, size);
    printf("-----|Transpose matrix A:\n");
    PrintMatrix(&m, size);

    FormatMatrix(&m, &x, size);
    printf("-----|Vector X:\n");
    PrintVector(&x, size);

    DevisionFunc(&x, &g, size);
    printf("-----|Function U : \n [%11f]\n\n", g);

    return 0;
}

void PrintVector(double Vector[size], int n) {
    for (int i = 0; i <= n - 1; i++) { // Print vector
        printf(" [%11f]", Vector[i]);
    }
    printf("\n\n");
}

void PrintMatrix(double Matrix[size][size], int n) {
    for (int i = 0; i <= n - 1; i++) { // Print matrix
        for (int j = 0; j <= n - 1; j++) {
            printf(" [%11f]", Matrix[i][j]);
        }
        printf("\n\n");
    }
}

void CreatMatrix(double Matrix[size][size], int n) {
    for (int i = 0; i <= n-1; i++) { //Filling array
        for (int j = 0; j <= n-1; j++) {
            Matrix[i][j] = 2 * exp(j) - 0.879 * i * i + 13.2 * j;
        }
    }
}

void TrnsposeMatrix(double Matrix[size][size], int n) {
    for (int i = 0; i <= n - 1; i++) { //Transposition array
        for (int j = 0; j <= n - 1; j++) {
            if (Matrix[i][j] == Matrix[i][n - 1 - i]) { // If [i][j] - on the
diagonal
                break;
            }
        }
    }
}
```

```

    }
    else {
        double temp = Matrix[i][j];
        Matrix[i][j] = Matrix[n-1-j][n-1-i];
        Matrix[n-1-j][n-1-i] = temp;
    }
}
}

void FormatMatrix(double Matrix[size][size], double Vector [size], int n){
    for (int i = 2;;) { //Vector X is:
        for (int j = 0; j <= n - 1; j++) {
            Vector[j] = Matrix[i][j];
        }
        break;
    }
}

void DevisionFunc(double Vector[size], double* FunctionG, int n) {
    for (int i = 0; i < n - 1; i++) { // Product of series
        *FunctionG *= 1 / (fabs(Vector[i]) + 1) + Vector[i + 1];
    }
}

```

Тестування

При $A = 6 \times 6$:

```

----|Filling matrix A:
[ 2.000000] [ 18.636564] [ 41.178112] [ 79.771074] [ 161.996300] [ 362.826318]
[ 1.121000] [ 17.757564] [ 40.299112] [ 78.892074] [ 161.117300] [ 361.947318]
[ -1.516000] [ 15.120564] [ 37.662112] [ 76.255074] [ 158.480300] [ 359.310318]
[ -5.911000] [ 10.725564] [ 33.267112] [ 71.860074] [ 154.085300] [ 354.915318]
[ -12.064000] [ 4.572564] [ 27.114112] [ 65.707074] [ 147.932300] [ 348.762318]
[ -19.975000] [ -3.338436] [ 19.203112] [ 57.796074] [ 140.021300] [ 340.851318]

----|Transpose matrix A:
[ 340.851318] [ 348.762318] [ 354.915318] [ 359.310318] [ 361.947318] [ 362.826318]
[ 140.021300] [ 147.932300] [ 154.085300] [ 158.480300] [ 161.117300] [ 161.996300]
[ 57.796074] [ 65.707074] [ 71.860074] [ 76.255074] [ 78.892074] [ 79.771074]
[ 19.203112] [ 27.114112] [ 33.267112] [ 37.662112] [ 40.299112] [ 41.178112]
[ -3.338436] [ 4.572564] [ 10.725564] [ 15.120564] [ 17.757564] [ 18.636564]
[ -19.975000] [ -12.064000] [ -5.911000] [ -1.516000] [ 1.121000] [ 2.000000]

----|Vector X:
[ 57.796074] [ 65.707074] [ 71.860074] [ 76.255074] [ 78.892074] [ 79.771074]

----|Function U :
[2268129693.101543]

```

При $A = 9 \times 9$:

```
----|Filling matrix A:
[ 2.000000] [ 18.636564] [ 41.178112] [ 79.771074] [ 161.996300] [ 362.826318] [ 886.057587] [ 2285.666317] [ 6067.515974]
[ 1.121000] [ 17.757564] [ 40.299112] [ 78.892074] [ 161.117300] [ 361.947318] [ 885.178587] [ 2284.787317] [ 6066.636974]
[ -1.516000] [ 15.120564] [ 37.662112] [ 76.255074] [ 158.480300] [ 359.310318] [ 882.541587] [ 2282.150317] [ 6063.999974]
[ -5.911000] [ 10.725564] [ 33.267112] [ 71.860074] [ 154.085300] [ 354.915318] [ 878.146587] [ 2277.755317] [ 6059.604974]
[ -12.064000] [ 4.572564] [ 27.114112] [ 65.707074] [ 147.932300] [ 348.762318] [ 871.993587] [ 2271.602317] [ 6053.451974]
[ -19.975000] [ -3.338436] [ 19.203112] [ 57.796074] [ 140.021300] [ 340.851318] [ 864.082587] [ 2263.691317] [ 6045.540974]
[ -29.644000] [ -13.007436] [ 9.534112] [ 48.127074] [ 130.352300] [ 331.182318] [ 854.413587] [ 2254.022317] [ 6035.871974]
[ -41.071000] [ -24.434436] [ -1.892888] [ 36.700074] [ 118.925300] [ 319.755318] [ 842.986587] [ 2242.595317] [ 6024.444974]
[ -54.256000] [ -37.619436] [ -15.077888] [ 23.515074] [ 105.740300] [ 306.570318] [ 829.801587] [ 2229.410317] [ 6011.259974]
----|Transpose matrix A:
[ 6011.259974] [ 6024.444974] [ 6035.871974] [ 6045.540974] [ 6053.451974] [ 6059.604974] [ 6063.999974] [ 6066.636974] [ 6067.515974]
[ 2229.410317] [ 2242.595317] [ 2254.022317] [ 2263.691317] [ 2271.602317] [ 2277.755317] [ 2282.150317] [ 2284.787317] [ 2285.666317]
[ 829.801587] [ 842.986587] [ 854.413587] [ 864.082587] [ 871.993587] [ 878.146587] [ 882.541587] [ 885.178587] [ 886.057587]
[ 306.570318] [ 319.755318] [ 331.182318] [ 340.851318] [ 348.762318] [ 354.915318] [ 359.310318] [ 361.947318] [ 362.826318]
[ 105.740300] [ 118.925300] [ 130.352300] [ 140.021300] [ 147.932300] [ 154.085300] [ 158.480300] [ 161.117300] [ 161.996300]
[ 23.515074] [ 36.700074] [ 48.127074] [ 57.796074] [ 65.707074] [ 71.860074] [ 76.255074] [ 78.892074] [ 79.771074]
[ -15.077888] [ -1.892888] [ 9.534112] [ 19.203112] [ 27.114112] [ 33.267112] [ 37.662112] [ 40.299112] [ 41.178112]
[ -37.619436] [ -24.434436] [ -13.007436] [ -3.338436] [ 4.572564] [ 10.725564] [ 15.120564] [ 17.757564] [ 18.636564]
[ -54.256000] [ -41.071000] [ -29.644000] [ -19.975000] [ -12.064000] [ -5.911000] [ -1.516000] [ 1.121000] [ 2.000000]
----|Vector X:
[ 829.801587] [ 842.986587] [ 854.413587] [ 864.082587] [ 871.993587] [ 878.146587] [ 882.541587] [ 885.178587] [ 886.057587]
----|Function U :
[ 329880785624294366380032.000000]
```

Матриця A :

```
[ 2.000000] [ 18.636564] [ 41.178112] [ 79.771074] [ 161.996300] [ 362.826318]
[ 1.121000] [ 17.757564] [ 40.299112] [ 78.892074] [ 161.117300] [ 361.947318]
[ -1.516000] [ 15.120564] [ 37.662112] [ 76.255074] [ 158.480300] [ 359.310318]
[ -5.911000] [ 10.725564] [ 33.267112] [ 71.860074] [ 154.085300] [ 354.915318]
[ -12.064000] [ 4.572564] [ 27.114112] [ 65.707074] [ 147.932300] [ 348.762318]
[ -19.975000] [ -3.338436] [ 19.203112] [ 57.796074] [ 140.021300] [ 340.851318]
```

Третій рядок транспонованої матриці

```
[ 57.796074] [ 65.707074] [ 71.860074] [ 76.255074] [ 78.892074] [ 79.771074]
```

Добуток елементів вектора X: {2268129693.101543}, що і є результатом.