Самостоятельная работа № 12.

Операции над массивами.

Задание 1:

Постановка задачи:

Найти произведения матриц $(AB) \cdot C$ ***и*** $A \cdot (BC)$***:***

$$A = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}, B = \begin{pmatrix} 2 & 0 \\ -3 & 1 \end{pmatrix}, C = \begin{pmatrix} 3 & -1 \\ 2 & 3 \end{pmatrix}$$

Мат.Модель:

$c_{ij} = a_{i1} \cdot b_{1j} + a_{i2} \cdot b_{2j} + ... + a_{in} \cdot b_{nj}$

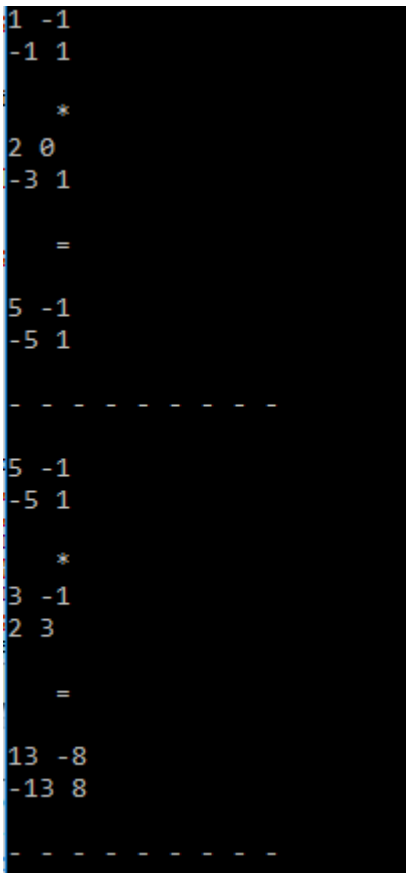
Список идентификаторов:

Название	**Тип**	**Функция**
A	int	Исходная матрица A 2x2
B	int	Исходная матрица B 2x2
C	int	Исходная матрица C 2x2
tmp	int	Промежуточный двумерный массив A*B 2x2
res	int	Итоговый двумерный массив A*B*C 2x2
p	int	Аргумент функции вывода массива
p1	int	1 перемножаемая матрица
p2	int	2 перемножаемая матрица
p3	int	результат перемножения матриц p1 и p2
i	int	Номер строки
j	int	Номер столбца
k	int	Промежуточная переменная

Код:

```c
#include <stdio.h>
#include <locale.h>
void seematrix (int p[2][2])
{
    int i,j;
    for (i = 0; i < 2; i++){
        for (j = 0; j < 2; j++)
            printf("%d ", p[i][j]);
        printf("\n");
    }
    printf("\n");
}
void multmatrix (int p1[2][2], int p2[2][2], int p3[2][2])
{
    seematrix(p1);
    printf("   *   \n");
    seematrix(p2);
    int i,j,k;
    for (i = 0; i < 2; i++){
        for (j = 0; j < 2; j++){
            for (k = 0; k < 2; k++)
                p3[i][j] += (p1[i][k] * p2[k][j]);
        }
    }
    printf("   =   \n");
    printf("\n");
    seematrix(p3);
    printf("- - - - - - - - -\n");
        printf("\n");
}
int main()
{
    char*locale = setlocale(LC_ALL, "");
    int tmp[2][2] = {{0,0},{0,0}};
    int res[2][2] = {{0,0},{0,0}};
    int A[2][2] = {{1,-1},{-1,1}};
    int B[2][2] = {{2,0},{-3,1}};
    int C[2][2] = {{3,-1},{2,3}};
    multmatrix(A,B,tmp);
    multmatrix(tmp,C,res);
    return 0;
}
```

Результат вывода:



Задание 2:

Постановка задачи:

Транспонировать матрицу $A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$

Мат.Модель:

$$A^T = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix}^T = \begin{pmatrix} a_{11} & \cdots & a_{m1} \\ \vdots & & \vdots \\ a_{1n} & \cdots & a_{mn} \end{pmatrix}$$

Список идентификаторов:

Название	Тип	Функция
p1	int	Матрица A переданная функции
p2	int	Транспонированная матрицы в функции
res	int	результат транспонирования
i	int	Номер строки
j	int	Номер столбца
A	int	Матрица A

Код:

```c
#include <stdio.h>
#include <locale.h>
void transpon (int p1[2][3], int n, int m, int p2[3][2])
{
    int i,j;
    for (j = 0; j < 3; j++){
        for (i = 0; i < 2; i++){
            p2[j][i] = p1[i][j];
            printf("%d ", p2[j][i]);
        }
        printf("\n");
    }
}

int main()
{
    char*locale = setlocale(LC_ALL, "");
    int A[2][3] = {{1,2,3},{4,5,6}};
    int res[3][2];
    transpon(A,2,3,res);
    return 0;
}
```

Результат вывода:

```
1 4
2 5
3 6
```

Задание 3:

Постановка задачи:

Вычислить произведения AA^T и A^TA при заданной матрице A:

$$A = \begin{pmatrix} 1 & 2 & 1 & 3 \\ 4 & -1 & 5 & -1 \end{pmatrix}$$

Мат.Модель:

$c_{ij} = a_{i1} \cdot b_{1j} + a_{i2} \cdot b_{2j} + ... + a_{in} \cdot b_{nj}$

$$A^T = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix}^T = \begin{pmatrix} a_{11} & \cdots & a_{m1} \\ \vdots & & \vdots \\ a_{1n} & \cdots & a_{mn} \end{pmatrix}$$

Список идентификаторов:

Название	Тип	Функция
A	int	Исходная матрица A
At	int	Транспонированная матрица A^T
N	int	Макрос хранящий кол-во строк массива A
M	int	Макрос хранящий кол-во столбцов массива A
size_x	int	Аргумент функции хранящий кол-во строк выводимого массива

size_y	int	Аргумент функции хранящий кол-во столбцов выводимого массива
*ptr	int	Указатель хранящий в себе адрес первого элемента массива
p2	int	2 перемножаемая матрица
p3	int	результат перемножения матриц p1 и p2
arr	int	Первый массив, переданный в функцию
arr1	int	Второй массив, переданный функцию
arr2	int	Третий массив, переданный в функцию
i	int	Переменная, отвечающая за номер строки
j	int	Переменная, отвечающая за номер столбца
k	int	Переменная, отвечающая за вычисления элемента

Код:

```c
#include <stdio.h>
#include <stdlib.h>
#define N 2
#define M 4
void print_matr(int size_x, int size_y, int *ptr) {
    for (int i = 0; i < size_x; i++) {
        for (int j = 0; j < size_y; j++) {
            printf("%d ", *(ptr++));
        }
        printf("\n");
    }
    printf("\n");
}
void tr_matr(int arr[], int arr1[], int size_x, int size_y){
    printf("A =   \n");
    print_matr(size_x,size_y,arr);
    for (int j = 0; j < size_y; j++) {
        for (int i = 0; i < size_x; i++)
            arr1[j*size_x + i] = arr[i*size_y + j];
    }
    printf("At =  \n");
    print_matr(size_y,size_x,arr1);
}
void mul_matr(int size_x, int size_y, int arr[], int size_x1, int size_y1, int arr1[], int arr2[]) {
    if (size_x != size_y1){
        printf("Умножение невозможно!!!");
    } else {
        print_matr(size_x, size_y, arr);
        printf("   *   \n");
        print_matr(size_x1, size_y1, arr1);
        for (int i = 0; i < size_x; i++) {
            for (int j = 0; j < size_y1; j++) {
                arr2[i * size_x + j] = 0;
                for (int k = 0; k < size_y; k++) {
                    arr2[i * size_x + j]  += arr[i * size_y + k] * arr1[k * size_y1 + j];
                }
            }
        }
        printf("   =   \n");
        print_matr(size_x, size_y1, arr2);
    }
}
```

```c
int main(){
    int A[N][M] = {{1,2,1,3},{4,-1,5,-1}};
    int At[M][N];
    int res[N][N];
    int resl[M][M];
    tr_matr(A,At,N,M);
    printf("---------------\n");
    mul_matr(N,M,A,M,N,At,res);
    printf("---------------\n");
    mul_matr(M,N,At,N,M,A,resl);
    return 0;
}
```

Результат вывода:

```
 A =
 1 2 1 3
 4 -1 5 -1

 At =
 1 4
 2 -1
 1 5
 3 -1

 ---------------
 1 2 1 3
 4 -1 5 -1

    *
 1 4
 2 -1
 1 5
 3 -1

    =
 15 4
 4 43

 ---------------
 1 4
 2 -1
 1 5
 3 -1

    *
 1 2 1 3
 4 -1 5 -1

    =
 17 -2 21 -1
 -2 5 -3 7
 21 -3 26 -2
 -1 7 -2 10
```

Задание 4:

Постановка задачи:

Преобразовать исходную матрицу так, чтобы первый элемент каждой строки был заменен средним арифметическим элементов этой строки.

Мат.Модель:

Список идентификаторов:

Название	Тип	Функция
p	float	Матрица переданная функции для вывода
i	int	Номер строки
j	int	Номер столбца
A	float	Матрица A

Код:

```c
#include <stdio.h>
#include <locale.h>
void seematrix (float p[2][4])
{
    int i,j;
    for (i = 0; i < 2; i++){
        for (j = 0; j < 4; j++)
            printf("%f ", p[i][j]);
        printf("\n");
    }
    printf(" - - - - - - - - \n");
}
int main()
{
    char*locale = setlocale(LC_ALL, "");
    float A[2][4] = {{1.0,2,1,3},{4,-1,5,-1}};
    int i,j;
    seematrix(A);
    for (i = 0; i < 2; i++){
        for (j = 0; j < 3; j++){
            A[i][0] += A[i][j+1];
        }
        A[i][0] /= 4;
    }
    seematrix(A);
    return 0;
}
```

Результат вывода:

```
1,000000 2,000000 1,000000 3,000000
4,000000 -1,000000 5,000000 -1,000000
 - - - - - - - - -
1,750000 2,000000 1,000000 3,000000
1,750000 -1,000000 5,000000 -1,000000
 - - - - - - - - -
```