

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г. ШУХОВА»
(БГТУ им. В.Г. Шухова)**

Кафедра программного обеспечения вычислительной техники и автоматизированных
систем

Лабораторная работа № 7
по дисциплине: Основы программирования
тема: «Массивы и указатели»

Выполнил: ст. группы ПВ202

Аладиб язан

Проверил:

Валентина Станиславовна

Притчин Иван Сергеевич

Лабораторная работа № 7

«Массивы и указатели»

Цель работы: Освоение работы с динамическими массивами в языке Си, осознание связи между массивами и указателями.

Задания варианта №2:

Дана матрица A . Найти произведение $A \cdot A^T$.

Тестовые данные:

Исходные данные				Результаты		
8	4	6	5	141	95	147
2	5	4	7	95	94	135
3	6	9	9	147	135	207

Текст программы:

```
//Дана матрица A. Найти произведение  $A \cdot A^T$ 

#include <stdlib.h>
#include <stdio.h>
#include <stdarg.h>

//Транспонирует матрицу
int** trm(int** mt, int rows, int cols) {
    int** At = (int**)calloc(cols, sizeof(int*)); //создаем новую матрицу
    for (int i = 0; i < cols; i++) {
        At[i] = (int*)calloc(rows, sizeof(int));
        for (int j = 0; j < rows; j++)
            At[i][j] = mt[j][i];
    }
    return At;
}

//Инициализирует строку матрицы
void initRow(int** mx, int row, int col, ...) {
    va_list arg;
    va_start(arg, col);
    for (int i = 0; i < col; i++)
        mx[row][i] = va_arg(arg, int);
}
```

```

        va_end(arg);
    }

//Умножает матрицы
int** mult(int** mt1, int rows1, int cols1, int** mt2, int rows2, int cols2) {
    int** Ar = (int**)calloc(rows1, sizeof(int*)); //новая матрица
    for (int i = 0, k = 0; i < rows1; k++) {
        if (!k)Ar[i] = (int*)calloc(cols2, sizeof(int));
        Ar[i][k] = 0;
        for (int j = 0; j < cols1; j++)
            Ar[i][k] += mt1[i][j] * mt2[j][k]; //вычисляем значения новой
матрицы
        if (k + 1 == cols2) {
            i++;
            k = -1;
        }
    }
    return Ar;
}

//Печатает матрицу
void print(int** mx, int rows, int cols) {
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++)
            printf("%d ", mx[i][j]);
        printf("\n");
    }
}

//Удаляет матрицу
void free_mx(int*** mx, int rows) {
    for (int i = 0; i < rows; i++)
        free((*mx)[i]);
    free(*mx);
    *mx = NULL;
}

int main() {

    const int row = 3;
    const int col = 4;
    int** A = (int**)calloc(row, sizeof(int*));
    for (int i = 0; i < row; i++)
        A[i] = (int*)calloc(col, sizeof(int));

    initRow(A, 0, col, /*matrix - */ 8, 4, 6, 5);
    initRow(A, 1, col, /*matrix - */ 2, 5, 4, 7);
    initRow(A, 2, col, /*matrix - */ 3, 6, 9, 9);
    int** At = trm(A, row, col);
    int** Ar = mult(A, row, col, At, col, row);
    print(Ar, row, row);

    free_mx(&A, row);
    free_mx(&At, col);
}

```

```
    free_mx(&Ar, row);  
    return 0;  
}
```



```
main.c  
60 int main() {  
61  
62     const int row = 3;  
63     const int col = 4;  
64     int** A = (int**)calloc(row, sizeof(int*));  
65     for (int i = 0; i < row; i++)  
66         A[i] = (int*)calloc(col, sizeof(int));  
67  
68     initRow(A, 0, col, /*matrix - */ 8, 4, 6, 5);  
69     initRow(A, 1, col, /*matrix - */ 2, 5, 4, 7);  
70     initRow(A, 2, col, /*matrix - */ 3, 6, 9, 9);  
71     int** At = trm(A, row, col);  
72     int** Ar = mult(A, row, col, At, col, row);  
73     print(Ar, row, row);  
74  
75     free_mx(&A, row);  
76     free_mx(&At, col);  
77     free_mx(&Ar, row);  
78     return 0;  
79 }  
80
```

input

```
141 95 147  
95 94 135  
147 135 207  
  
...Program finished with exit code 0  
Press ENTER to exit console.
```