

Лабораторная работа №3

Задание 3.1

Постановка задачи

3.1: Для некоторого числового вектора X , введённого с клавиатуры, вычислить значения вектора $Y = X \cdot X$ ($y_i = x_i \cdot x_i$ — поэлементно).

Математическая модель

$$Y = X \cdot X \quad (y_i = x_i \cdot x_i \text{ — поэлементно}).$$

Описание переменных

Переменная	Тип	Суть
N	Int const	Длина числового вектора X
x[N]	Array of int	Числовой вектор X
n	int	Текущий эл-т массива
i	int	Счётчик цикла

Код программы

```
#include <stdio.h>

void main() {
    int const N = 5;
    int x[N], n;

    for (int i = 0; i < N; ++i) {
        printf("X[%d] = ", i + 1);
        scanf("%d", &n);
        x[i] = n * n;
    }

    printf("Y = [");
    for (int i = 0; i < N; ++i) {
        printf(" %d", x[i]);
    }
    printf(" ]\n");
}
```

Вывод программы

```
X[1] = 1
X[2] = 2
X[3] = 3
X[4] = 4
X[5] = 5
Y = [ 1 4 9 16 25 ]
```

Задание 3.2

Постановка задачи

Для некоторого числового массива X , введённого с клавиатуры, изменить порядок элементов на обратный без привлечения вспомогательного массива и со вспомогательным массивом.

Описание переменных

Переменная	Тип	Суть
N	Int const	Длина числового вектора X
x[N]	Array of int	Числовой вектор X
n	int	Текущий эл-т массива
i	int	Счётчик цикла
y[N]	Array of int	Числовой вектор Y

Код программы

```
#include <stdio.h>

void main() {
    int const N = 5;
    int x[N], n;

    for (int i = 0; i < N; ++i) {
        printf("X[%d] = ", i + 1);
        scanf("%d", &x[i]);
    }

    // Без вспомогательного массива
    printf("[");
    for (int i = N - 1; i >= 0; --i) {
        printf(" %d", x[i]);
    }
    printf(" ]\n");

    // С привлечением вспомогательного массива
    int y[N];
```

```
for (int i = 0; i < N; ++i) {  
    y[N - i - 1] = x[i];  
}  
  
printf("[");  
for (int i = 0; i < N; ++i) {  
    printf(" %d", y[i]);  
}  
printf(" ]\n");  
}
```

Вывод программы

```
X[1] = 1  
X[2] = 2  
X[3] = 3  
X[4] = 4  
X[5] = 5  
[ 5 4 3 2 1 ]  
[ 5 4 3 2 1 ]
```

Задание 3.3

Постановка задачи

3.3: Транспонировать матрицу:

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \quad (1)$$

Описание переменных

Переменная	Тип	Суть
a	Array int	Двумерный массив (матрица)
i	int	Счётчик цикла
j	int	Счётчик цикла

Код программы

```
#include <stdio.h>

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

    for (int i = 0; i < 3; ++i) {
        for (int j = 0; j < 3; ++j) printf("%d ", a[j][i]);
        printf("\n");
    }
}
```

Вывод программы

```
1 4 7
2 5 8
3 6 9
```

Задание 3.4

Постановка задачи

3.4: Найти произведения матриц ABC и CBA :

$$A = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}, B = \begin{bmatrix} 2 & 0 \\ -3 & 1 \end{bmatrix}, C = \begin{bmatrix} 3 & -1 \\ 2 & 3 \end{bmatrix}. \quad (2)$$

Математическая модель

$$X_1 = A \times B \times C;$$

$$X_2 = C \times B \times A$$

Описание переменных

Переменная	Тип	Суть
a	2D array of int	Матрица A
b	2D array of int	Матрица B
c	2D array of int	Матрица C
ab	2D array of int	Матрица AB
bc	2D array of int	Матрица BC
abc	2D array of int	Матрица ABC
bca	2D array of int	Матрица BCA
i	int	i координата
j	int	j координата
p	int	p координата
N	const	Размер матрицы

Код программы

```
#include <stdio.h>
#define N 2

void main() {
    int
        a[N][N] = {{1, -1}, {-1, 1}},
```

```
b[N][N] = {{2, 0}, {-3, 1}},  
c[N][N] = {{3, -1}, {2, 3}},  
ab[N][N] = {0}, cb[N][N] = {0}, abc[N][N] = {0}, cba[N][N] = {0};
```

```
// умножить A на B, C на A  
for (int i = 0; i < N; ++i) {  
    for (int j = 0; j < N; ++j) {  
        for (int p = 0; p < N; ++p) {  
            ab[i][j] += a[i][p] * b[p][j];  
            cb[i][j] += c[i][p] * b[p][j];  
        }  
    }  
}  
  
// домножить на C, на A  
for (int i = 0; i < N; ++i) {  
    for (int j = 0; j < N; ++j) {  
        for (int p = 0; p < N; ++p) {  
            abc[i][j] += ab[i][p] * c[p][j];  
            cba[i][j] += cb[i][p] * a[p][j];  
        }  
    }  
}
```

```
// вывести результат  
printf("A*B*C = \n");  
for (int i = 0; i < N; ++i) {  
    for (int j = 0; j < N; ++j)  
        printf("%3d ", abc[i][j]);  
    printf("\n");  
}
```

```
printf("\nC*B*A = \n");  
for (int i = 0; i < N; ++i) {  
    for (int j = 0; j < N; ++j)  
        printf("%3d ", cba[i][j]);  
    printf("\n");  
}
```

Вывод программы

$A * B * C =$
13 -8
-13 8

$C * B * A =$
10 -10
-8 8

Задание 3.5

Постановка задачи

3.5: Найти значения матричного многочлена $f(A)$, где

$$f(A) = -2A^2 + 5A + 9. \quad (3)$$

Математическая модель

$$f(A) = -2A^2 + 5A + 9.$$

Описание переменных

Переменная	Тип	Суть
a	2D array of int	Матрица A
a2	2D array of int	Матрица A^2
i	int	i координата
j	int	j координата
p	int	p координата
N	const	Размер матрицы

Код программы

```
#include <stdio.h>
#define N 2

void main() {
    int
        a[N][N] = {{2, 0}, {-3, 1}},
        a2[N][N] = {0};

    // умножить A на A
    for (int i = 0; i < N; ++i) {
        for (int j = 0; j < N; ++j) {
            for (int p = 0; p < N; ++p) {
                a2[i][j] += a[i][p] * a[p][j];
            }
        }
    }

    // вывести результат
    printf("A = \n");
    for (int i = 0; i < N; ++i) {
        for (int j = 0; j < N; ++j)
            printf("%3d ", a[i][j]);
        printf("\n");
    }

    printf("\nf(A) = \n");
    for (int i = 0; i < N; ++i) {
        for (int j = 0; j < N; ++j)
            printf("%3d ", -2 * a2[i][j] + 5 * a[i][j] + 9);
        printf("\n");
    }
}
```

Вывод программы

```
A =
  2   0
 -3   1

f(A) =
11   9
12  12
```

Задание 3.6

Постановка задачи

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

Описание переменных

Переменная	Тип	Суть
N	Int const	Число строк в матрице
M	Int const	Число столбцов в матрице
i	int	Счётчик внешнего цикла
j	int	Счётчик вложенного цикла
x[N][M]	2D array of int	Исходная матрица
y[N][M]	2D array of int	Результирующая матрица

Код программы

```
#include <stdio.h>

void main() {
    int const N = 3, M = 4;
    int x[N][M], y[N][M], s;

    for (int i = 0; i < N; ++i) {
        s = 0;
        for (int j = 0; j < M; ++j) {
            printf("X[%d][%d] = ", i + 1, j + 1);
            scanf("%d", &x[i][j]);
            s += x[i][j];
        }
        x[i][0] = s / M;
    }

    printf("\n");
    for (int i = 0; i < N; ++i) {
        for (int j = 0; j < M; ++j) {
            printf("%d ", x[i][j]);
        }
        printf("\n");
    }
}
```

Вывод программы

```
X[1][1] = 1
X[1][2] = 2
X[1][3] = 3
X[1][4] = 4
X[2][1] = 5
X[2][2] = 6
X[2][3] = 7
X[2][4] = 8
X[3][1] = 1
X[3][2] = 2
X[3][3] = 3
X[3][4] = 4
```

```
2 2 3 4
6 6 7 8
2 2 3 4
```

Задание 3.7

Постановка задачи

3.7: Реализовать самостоятельно алгоритм сортировки вставками.

Описание переменных

Переменная	Тип	Суть
N	Int const	Длина массива
x	Arr of int	Массив для сортировки
i	int	Счетчик внешнего цикла
j	int	Счетчик вложенного цикла
tmp	int	Буфер обмена

Код программы

```
#include <stdio.h>

void main() {
    int const N = 5;
    int x[N];

    //Ввод
    for (int i = 0; i < N; ++i) {
```

```
printf("X[%d] = ", i + 1);  
scanf("%d", &x[i]);  
}  
  
//Сортировка  
for(int i = 1; i < N; ++i) {  
    for(int j = i; j > 0 && x[j - 1] > x[j]; j--) {  
        int tmp = x[j - 1];  
        x[j - 1] = x[j];  
        x[j] = tmp;  
    }  
}  
  
//Вывод  
printf("\nОтсортированный массив: [");  
for (int i = 0; i < N; ++i) {  
    printf(" %d", x[i]);  
}  
printf(" ]\n");  
}
```

Вывод программы

```
X[1] = 3  
X[2] = 4  
X[3] = 2  
X[4] = 5  
X[5] = 1
```

Отсортированный массив: [1 2 3 4 5]

Задание 4.1

Постановка задачи

4.1: Создать динамический одномерный массив целых чисел F размерности 12 и заполнить его положительными и отрицательными числами.

$$T = \frac{R + Q + S}{R \cdot Q \cdot S + 2}, \quad (4)$$

где R - сумма отрицательных элементов F , Q - отрицательный элемент массива, S — произведение положительных элементов массива F .

Описание переменных

Переменная	Тип	Суть
*f	Int pointer	Указатель на массив
N	Int const	Длина массива
i	int	Счётчик цикла
t	double	T (см. условие)
r	double	R (см. условие)
q	double	Q (см. условие)
s	double	S (см. условие)

Код программы

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#define N 12

void main() {
    srand(time(NULL));
    int *f;
    f = (int *) malloc(12 * sizeof(int));

    printf("Массив F: [ ");
    for(int i = 0; i < N; ++i) {
        f[i] = rand() % 21 - 10;
        printf("%3d ", f[i]);
    }
    printf("]");
```

```
// Вычислить T
double t, r = 0, q, s = 1;
for(int i = 0; i < N; ++i) {
    if (f[i] > 0) s *= f[i];
    if (f[i] < 0) {
        r += f[i];
        q = f[i];
    }
}

t = (r + q + s) / (r * q * s + 2);
printf("\nT(%lg) = %lg\n", q, t);
}
```

Вывод программы

Массив F: [9 7 4 0 -10 -4 -8 -9 3 1 -9 6]
T(-9) = 0.00274777

Задание 4.2

Постановка задачи

4.2: Дан одномерный массив целых чисел A размера 12. Вычислить:

$$Y = (U + T) \cdot (S + 2), \quad (5)$$

где S - количество элементов массива A с нечётными индексами, T - наибольший по модулю элемент массива A , U - сумма отрицательных элементов массива A .

Математическая модель

$$Y = (U + T) \cdot (S + 2), \quad (5)$$

где S - количество элементов массива A с нечётными индексами, T - наибольший по модулю элемент массива A , U - сумма отрицательных элементов массива A .

Описание переменных

Переменная	Тип	Суть
*a	Int pointer	Указатель на массив
N	const	Длина массива
i	int	Счётчик цикла
y	double	Y (см. условие)
u	double	U (см. условие)
t	double	T (см. условие)
s	int	S (см. условие)

Код программы

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <math.h>
#define N 12

void main() {
    srand(time(NULL));
    int *a;
    a = (int *) malloc(12 * sizeof(int));

    printf("Массив A: [ ");
    for(int i = 0; i < N; ++i) {
        a[i] = rand() % 21 - 10;
        printf("%d ", a[i]);
    }
    printf("]");

    // Вычислить y
    double y, u = 0, t;
    int s = N / 2;
    t = abs(a[0]);
    for(int i = 0; i < N; ++i) {
        if (abs(a[i]) > t) t = a[i];
        if (a[i] < 0) u += a[i];
    }

    y = (u + t) * (s + 2);
    printf("\nY = %lg\n", y);
}
```

Вывод программы

Массив A: [-7 -5 5 5 -6 4 -7 2 10 4 -10 2]
Y = -200

Задание 4.3

Постановка задачи

4.3: Написать программу, которая вычисляет некоторый вектор b , как результат умножения некоторой матрицы M на вектор a :

$$b = M \times a \quad (6)$$

Математическая модель

$$b = M \times a$$

Описание переменных

Переменная	Тип	Суть
N	Int const	Число строк в матрице
M	Int const	Число столбцов в матрице
m	int	Счётчик внешнего цикла
n	int	Счётчик вложенного цикла
A[N][M]	2D array of int	Исходная матрица
V[N]	array of int	Вектор-множитель
s	int	Эл-т результирующей м-цы

Код программы

```
#include <stdio.h>
#include <time.h>
#include <stdlib.h>

void main() {
    int const M = 5, N = 5;
    int A[M][N], V[N];
    srand(time(NULL));

    printf("a = ");
    for (int m = 0; m < M; ++m) {
        V[m] = rand() % 10;
        printf("\n%3d ", V[m]);
    }
}
```

```
printf("\n\nA = \n");
for (int m = 0; m < M; ++m) {
    for (int n = 0; n < N; ++n) {
        A[m][n] = rand() % 10;
        printf(" %2d", A[m][n]);
    }
    printf("\n");
}

printf("\nA * a =");
for (int m = 0; m < M; ++m) {
    int s = 0;
    for (int n = 0; n < N; ++n) {
        s += A[m][n] * V[n];
    }
    printf("\n%d", s);
}
}
```

Вывод программы

a =
0
2
0
5
5

A =
9 2 4 6 3
0 8 6 6 4
9 2 7 4 8
7 8 8 9 4
7 2 1 9 5

A * a =
49
66
64
81

Задание 4.4

Постановка задачи

4.4: Выделить динамически память под некоторую матрицу A размерности $M \times N$ и заполнить её произвольными числами. Сократить правильно размер этой матрицы, удалив из неё одну выбранную строку, освободив от неё также и память, используя указатели. После удаления строки в матрице $A_{M \times N}$ должна быть возможность обхода всех элементов «новой» матрицы $A_{M-1 \times N}$ таким же способом, что и изначальной матрицы. Распечатать (используя циклы) матрицу до удаления строки и после удаления.

Описание переменных

Переменная	Тип	Суть
m	Int	Число строк
n	int	Число столбцов
**a	Int pointer	Двумерный массив
i	int	Счетчик внешнего цикла
j	int	Счетчик вложенного цикла

Код программы

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <math.h>

void main() {
    srand(time(NULL));
    int n, m;
    n = rand() % 10 + 3;
    m = rand() % 10 + 3;

    // объявить матрицу
    int **a = NULL;
    a = (int **) malloc(m * sizeof(int *));
    for (int i = 0; i < m; ++i) {
```

```
a[i] = (int *) malloc(n * sizeof(int));
}

// задать и вывести матрицу
printf("Матрица A[%d x %d]:\n", m, n);
for(int i = 0; i < m; ++i) {
    for(int j = 0; j < n; ++j) {
        a[i][j] = rand() % 21 - 10;
        printf("%3d ", a[i][j]);
    }
    printf("\n");
}

// удалить 2 строку
free(a[1]);
m -= 1;
for(int i = 1; i < m; ++i) {
    a[i] = a[i + 1];
}

// вывести результат
printf("\nМатрица A[%d x %d] (удалена 2 строка):\n", m, n);
for(int i = 0; i < m; ++i) {
    for(int j = 0; j < n; ++j) {
        printf("%3d ", a[i][j]);
    }
    printf("\n");
}
}
```

Вывод программы

Матрица A[10 x 9]:

-10	-10	-2	7	-2	-1	-7	1	-10
-5	4	-5	8	-6	-4	-2	4	1
-4	0	7	6	-1	10	10	7	6
0	3	-4	5	1	-6	-10	-2	1
-1	-1	10	-3	4	1	0	-1	5
6	7	-3	-6	-9	5	-10	5	-7
-10	5	8	6	-6	0	10	9	-9
-9	9	-3	2	-5	6	0	0	-3
-10	-10	7	4	4	1	-10	9	2
5	9	-3	8	7	10	5	0	-7

Матрица A[9 x 9] (удалена 2 строка):

-10	-10	-2	7	-2	-1	-7	1	-10
-----	-----	----	---	----	----	----	---	-----

-4	0	7	6	-1	10	10	7	6
0	3	-4	5	1	-6	-10	-2	1
-1	-1	10	-3	4	1	0	-1	5
6	7	-3	-6	-9	5	-10	5	-7
-10	5	8	6	-6	0	10	9	-9
-9	9	-3	2	-5	6	0	0	-3
-10	-10	7	4	4	1	-10	9	2
5	9	-3	8	7	10	5	0	-7

Задание 4.5

Постановка задачи

4.5: Напишите программу для вычисления пересечения двух конечных множеств (наборов) А и В целых чисел одинакового размера с использованием динамических массивов. В качестве множества А можно взять первые 12 чисел ряда Фибоначчи: 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144. В качестве множества В можно взять первые 12 чисел последовательности Падована: 1, 1, 1, 2, 2, 3, 4, 5, 7, 9, 12, 16. Дублирующиеся значения можно исключать.

Описание переменных

Переменная	Тип	Суть
N	const	Длина массива
a	Arr of int	Массив А
b	Arr of int	Массив В
*c	Int pointer	Массив пересечения
c_len	int	Длина массива пересечения
i	int	Счетчик внешнего цикла
j	int	Счетчик вложенного цикла

Код программы

```
#include <stdio.h>
#include <stdlib.h>
#define N 12

void main() {
    int a[N] = {1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144},
        b[N] = {1, 1, 1, 2, 2, 3, 4, 5, 7, 9, 12, 16};

    printf("Массив A: [ ");
    for(int i = 0; i < N; ++i) {
        printf("%d ", a[i]);
    }
    printf("\n");

    printf("Массив B: [ ");
    for(int i = 0; i < N; ++i) {
        printf("%d ", b[i]);
    }
    printf("]");

    int *c, c_len = 0;
    c = (int *) malloc(N * sizeof(int));

    for(int i = 0; i < N; ++i) {
        for(int j = 0; j < N; ++j) {
            if (a[i] == b[j]) {
                c[c_len] = a[i];
                c_len += 1;
                break;
            }
        }
    }

    printf("\nПересечение: [ ");
    for(int i = 0; i < c_len; ++i) {
        printf("%d ", c[i]);
    }
    printf("]");
}
```

Вывод программы

Массив А: [1 1 2 3 5 8 13 21 34 55 89 144]

Массив В: [1 1 1 2 2 3 4 5 7 9 12 16]

Пересечение: [1 1 2 3 5]

Задание 4.6

Постановка задачи

4.6: Напишите программу, в которой создаётся квадратная матрица, заполненная нулями и единицами. Единичные значения у тех элементов, для которых сумма индексов является нечётным числом. Нулевые значения у тех элементов, для которых сумма индексов является чётным числом.

Описание переменных

Переменная	Тип	Суть
m	Int	Число строк
n	int	Число столбцов
**a	Int pointer	Двумерный массив
i	int	Счетчик внешнего цикла
j	int	Счетчик вложенного цикла

Код программы

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <math.h>

void main() {
    srand(time(NULL));
    int n, m;
    n = rand() % 10 + 3;
    m = rand() % 10 + 3;

    // объявить матрицу
    int **a = NULL;
```

```
a = (int **) malloc(m * sizeof(int *));
for (int i = 0; i < m; ++i) {
    a[i] = (int *) malloc(n * sizeof(int));
}

// задать и вывести матрицу
printf("Матрица A[%d x %d]:\n", m, n);
for(int i = 0; i < m; ++i) {
    for(int j = 0; j < n; ++j) {
        a[i][j] = (i + j) % 2;
        printf("%3d ", a[i][j]);
    }
    printf("\n");
}
}
```

Вывод программы

Матрица A[6 x 6]:

0	1	0	1	0	1
1	0	1	0	1	0
0	1	0	1	0	1
1	0	1	0	1	0
0	1	0	1	0	1
1	0	1	0	1	0