

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Алтайский государственный технический университет им. И. И. Ползунова»

Факультет информационных технологий
Кафедра прикладной математики

Отчет защищен с оценкой _____

Преподаватель _____
(подпись)
«___» _____ 2025 г.

Отчет
По лабораторной работе №8

«Динамические массивы»

по дисциплине «Программирование – 1 семестр»

Студент группы ПИ-54 Дворников М. С.

Преподаватель доцент, к.т.н. Егорова Е.В.

Барнаул 2025

Задание 1. Написать программу для ввода, обработки и вывода одномерного динамического массива.

№5.

Дан динамический символьный массив c_1, c_2, \dots, c_m . Подсчитать n - количество символов – латинских букв 'a' и 'z' в заданном массиве.

Создать новый динамический целочисленный массив из n элементов, каждый из которых последовательно равен номеру позиции, в которой в исходном массиве расположен символ 'a' или 'z'.

Все исходные данные, результаты и соответствующие адреса распечатать.

Программа 1:

```
#define _CRT_SECURE_NO_WARNINGS

#include <stdio.h> // Стандартный вход/выход
#include <locale.h> // Вывод на русском языке
#include <conio.h> // Задержка экрана
#include <string.h> // Работа со строками
#include <stdlib.h> // Создание динамических массивов
#include <math.h> // Математические функции

#include <windows.h> // Очистка консоли

#define RU setlocale(LC_ALL, "Russian") // Для вывода русских слов

int main() {
    RU;
    do {
        int m = 0; // Размер массива
        char buffer; // Сюда мы будем сохранять ввод
        int n; // Количество букв "a" и "z"
        char* array; // Массив на ввод
        char* array_ind; // Массив на индексы

        printf("Введите m, размер символьного массива, m > 0\n");
        do scanf("%d", &m);
        while (m < 1);

        // Выделяем память
        array = (char*)calloc(m, sizeof(char));
        if (array == NULL)
        {
            puts("Нет памяти. Конец работы\n");
            return 1;
        }

        // Пользователь вводит массив
        printf("\nВведите массив посимвольно:\n");
        getchar(); // Очищаем поток
        n = 0;
        for (int i = 0; i < m; i++) {
```

```

        buffer = getchar();
        *(array + i) = buffer;
        if (buffer == 'a' || buffer == 'z')
            n++;
    }

printf("\nПолучился вот такой массив:\n");
for (int i = 0; i < m - 1; i++)
    printf("%c, ", *(array + i));
printf("%c\n", *(array + m - 1));
printf("\nВ памяти лежат значения:\n");
for (int i = 0; i < m; i++)
    printf("%p - %c ~ %d\n", array + i, *(array + i), *(array + i));
printf("В нём %d элементов-символов \"a\" и \"z\"\n", n);

// Теперь создадим новый массив из индексов "a" и "z"
array_ind = (char*)calloc(n, sizeof(char));
for (int i = 0, j = 0; j < n; i++) {
    if (*(array + i) == 'a' || *(array + i) == 'z')
        *(array_ind + j++) = *(array + i);
}

printf("\nИз символов \"a\" и \"z\" получился вот такой массив:\n");
for (int i = 0; i < n - 1; i++)
    printf("%c, ", *(array_ind + i));
printf("%c\n", *(array_ind + n - 1));
printf("\nВ памяти лежат значения:\n");
for (int i = 0; i < n; i++)
    printf("%p - %c ~ %d\n", array_ind + i, *(array_ind + i),
*(array_ind + i));
free(array); free(array_ind); // Очищаем память
printf("Нажмите ESC, чтобы закончить.\n");
printf("Нажмите любую другую кнопку, чтобы продолжить...\n");
} while (_getch() != 27);
return 0;
}

```

Тесты программы 1:

Введите m , размер символьного массива, $m > 0$
10

Введите массив посимвольно:
abcazzappppppppppppp

Получился вот такой массив:
a, b, c, a, z, z, a, p, p, p

В памяти лежат значения:

0000028309004C90 - a ~ 97

0000028309004C92 = C ≈ 9

0000028309004C93 - a ≈ 97

0000028309004C94 - z ~ 131

0000028309004C94 - 2 x 122
0000038300004C8E - 3 x 131

0000028309004C93 - 2 - 12
0000028300004C06 - 2 - 03

0000028309004C96 - a ~ 9/
0000028300004C03 - a ~ 11/

0000028309004C97 - p ~ 112

0000028309004C98 - p ~ 112

0000028309004C99 - p ~ 112

В нём 5 элементов-символов

```

Из символов "а" и "z" получился вот такой массив:
1, 4, 5, 6, 7

В памяти лежат значения:
0000028309004CE0 - 1
0000028309004CE1 - 4
0000028309004CE2 - 5
0000028309004CE3 - 6
0000028309004CE4 - 7
Нажмите ESC, чтобы закончить.
Нажмите любую другую кнопку, чтобы продолжить...

```

Задание 2. Написать программу для ввода, обработки и вывода двумерного динамического массива..

№1.

Дана динамическая символьная матрица размером ($n \times m$) и динамическая целая переменная h . Если $h < 1$, то построить динамический одномерный массив, каждый элемент которого равен разности в одной строке числа элементов исходной матрицы, равных 'а' и равных 'б'; если $h \geq 1$, то вычислить количество элементов, равных 'а' и 'б' в исходной матрице, и записать это число как динамическую переменную.

Освободить память от матрицы.

Все исходные данные, результаты и соответствующие адреса распечатать.

Программа 2:

```

#define _CRT_SECURE_NO_WARNINGS

#include <stdio.h> // Стандартный вход/выход
#include <locale.h> // Вывод на русском языке
#include <conio.h> // Задержка экрана
#include <string.h> // Работа со строками
#include <stdlib.h> // Создание динамических массивов
#include <math.h> // Математические функции

#include <windows.h> // Очистка консоли

#define RU setlocale(LC_ALL, "Russian") // Для вывода русских слов

int main() {
    RU;
    do {
        int n, m; // Размеры массива
        char* array; // Массив на ввод
        char* array_ind; // Массив на индексы

```

```

int* h;

printf("Введите n, количество строчек двумерного символьного массива, n
> 0\n");
do scanf("%d", &n);
while (n < 1);

printf("Введите m, количество столбцов двумерного символьного массива,
m > 0\n");
do scanf("%d", &m);
while (m < 1);

// Выделяем память
array = (char*)calloc(n * m, sizeof(char));
if (array == NULL)
{
    puts("Нет памяти на массив. Конец работы\n");
    return 1;
}

// Пользователь вводит массив
printf("\nВведите массив посимвольно:\n");
getchar(); // Очищаем поток

for (int i = 0;i < n;i++)
    for (int j = 0;j < m;j++)
        *(array + i * m + j) = getchar();

printf("\nПолучился вот такой массив:\n");
for (int i = 0;i < n;i++) {
    for (int j = 0;j < m;j++)
        printf("%c", *(array + i * m + j));
    printf("\n");
}
printf("\nВ памяти лежат значения:\n");
for (int i = 0; i < m * n; i++)
    printf("%p - %c ~ %d\n", array + i, *(array + i), *(array + i));

h = (int*)malloc(sizeof(int));
if (h == NULL)
{
    puts("Нет памяти на h. Конец работы\n");
    return 1;
}

printf("\nВведите h:\n");
scanf("%d", h);

printf("\nВ памяти лежит значение:\n");
printf("%p - %d\n", h, *h);

// Обрабатываем данные в зависимости от h
if (*h < 1) {
    int* diff = (int*)calloc(n, sizeof(int));
    if (diff == NULL)
    {
        puts("Нет памяти на массив. Конец работы\n");
        return 1;
    }
    for (int i = 0; i < n; i++) {
        *(diff + i) = 0;
        for (int j = 0; j < m; j++) {
            if (*(array + i * m + j) == 'a')
                *(diff + i) += 1;
            else if (*(array + i * m + j) == 'b')

```

```

                *(diff + i) -= 1;
            }
            if (*(diff + i) < 0)
                *(diff + i) *= -1;
        }
        printf("\nПолучился вот такой массив из разниц количеств \"a\" и
\"b\":\n");
        for (int i = 0; i < n - 1; i++)
            printf("%d, ", *(diff + i));
        printf("%d\n", *(diff + n - 1));

        printf("\nВ памяти лежат значения:\n");
        for (int i = 0; i < n; i++)
            printf("%p - %d\n", diff + i, *(diff + i));
    }
    else {
        int* diff = (int*)malloc(sizeof(int));
        if (diff == NULL)
        {
            puts("Нет памяти на количество а и б. Конец работы\n");
            return 1;
        }
        *diff = 0;
        for (int i = 0; i < n; i++)
            for (int j = 0; j < m; j++)
                if (*(array + i * m + j) == 'a' || *(array + i * m +
j) == 'b')
                    *diff += 1;
        printf("\nПолучается, всего в массиве %d элементов \"a\" и
\"b\"\n", *diff);
        printf("\nВ памяти лежит значение:\n");
        printf("%p - %d\n", diff, *diff);
    }
    free(array); free(h); // Очищаем память
    printf("\nНажмите ESC, чтобы закончить.\n");
    printf("Нажмите любую другую кнопку, чтобы продолжить...\n");
} while (_getch() != 27);
return 0;
}

```

Тесты программы 2:

$h < 1$

Введите n , количество строчек двумерного символьного массива, $n > 0$

3

Введите m , количество столбцов двумерного символьного массива, $m > 0$

2

Введите массив посимвольно:

adbdab

Получился вот такой массив:

ad

bd

ab

В памяти лежат значения:

000001170A3C9770	- a ~ 97
000001170A3C9771	- d ~ 100
000001170A3C9772	- b ~ 98
000001170A3C9773	- d ~ 100
000001170A3C9774	- a ~ 97
000001170A3C9775	- b ~ 98

Введите h:

-5

В памяти лежит значение:

000001170A3CA520 - -5

Получился вот такой массив из разниц количеств "a" и "b":
1, 1, 0

В памяти лежат значения:

000001170A3C97C0 - 1

000001170A3C97C4 - 1

000001170A3C97C8 - 0

Нажмите ESC, чтобы закончить.

Нажмите любую другую кнопку, чтобы продолжить...

h >= 1

Введите n, количество строчек двумерного символьного массива, n > 0

2

Введите m, количество столбцов двумерного символьного массива, m > 0

3

Введите массив посимвольно:

abcdbb

Получился вот такой массив:

abc

dbb

В памяти лежат значения:

000001170A3C95E0 - a ~ 97

000001170A3C95E1 - b ~ 98

000001170A3C95E2 - c ~ 99

000001170A3C95E3 - d ~ 100

000001170A3C95E4 - b ~ 98

000001170A3C95E5 - b ~ 98

Введите h:

120

В памяти лежит значение:

000001170A3CAEA0 - 120

Получается, всего в массиве 4 элементов "a" и "b"

В памяти лежит значение:

000001170A3CA560 - 4

Нажмите ESC, чтобы закончить.

Нажмите любую другую кнопку, чтобы продолжить...