Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования «Тульский государственный университет»

КАФЕДРА ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

**ДИНАМИЧЕСКИЕ ПЕРЕМЕННЫЕ В ЯЗЫКЕ ПРОГРАММИРОВАНИЯ C**

отчет о лабораторной работе № 6

по дисциплине

*ТЕХНОЛОГИИ И МЕТОДЫ ПРОГРАММИРОВАНИЯ*

***ВАРИАНТ 13***

|  |  |  |
| --- | --- | --- |
| Выполнила: | ст. гр. 230711 | Павлова В.С. |
| Проверил: | асс. каф. ИБ | Курбаков М.Ю. |

Тула, 2023 г.

**ЦЕЛЬ И ЗАДАЧА РАБОТЫ**

**Цель:** научиться использовать динамические переменные.

**Задача:** в данной работе требуется написать программу с использованием динамических переменных, в которой данные сначала из файла считываются в память, потом обрабатываются и записываются в файл

**ЗАДАНИЕ НА РАБОТУ**

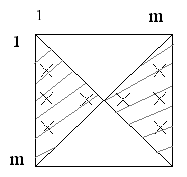
Выполнить задание по работе с двумерными массивами из лабораторной работы № 4 «Работа с массивами в языке программирования C», взяв в качестве номера варианта значение (30-NumInList), где NumInList – номер по списку, и используя динамическое создание массивов. Размер массива должен задаваться пользователем с клавиатуры. Результат работы программы должен выводиться на экран и в файл.

**Задание №1 на тему «Одномерные массивы»**

**Вариант №17**. Задана последовательность из N вещественных чисел. Вычислить значение функции: .

**Задание №2 на тему «Двумерные массивы»**

**Вариант №17.** Вычислить сумму элементов квадратной матрицы, помеченных символом х.



# **СХЕМА ПРОГРАММЫ ДЛЯ ЗАДАНИЯ №1**

Схема алгоритма программы, реализующей обработку одномерного массива (входной последовательности) по заданию варианта представлена на рисунке 1.

  
Рисунок 1 – Схема алгоритма программы, реализующей обработку одномерного массива (входной последовательности)

# **ТЕКСТ ПРОГРАММЫ ДЛЯ ЗАДАНИЯ №1**

Текст программы на языке программирования С/С++, реализующей обработку одномерного массива (входной последовательности) по заданию варианта представлен в листинге 1.

## **Листинг 1. Текст программы**

#include <iostream>

#include <cmath>

int main()

{

setlocale(LC\_ALL, "RUSSIAN");

int n; double res = 1;

std::cout << "Введите длину последовательности: "; std::cin >> n;

double\* a = new double[n];

for (int i = 0; i < n; i++){

std::cin >> a[i];

res \*= a[i];

}

std::cout << "\nРезультат вычислений: " << pow(abs(res), 1.0 / n);

return 0;   
}

**ИНСТРУКЦИЯ ПРОГРАММИСТА ДЛЯ ЗАДАНИЯ №1**

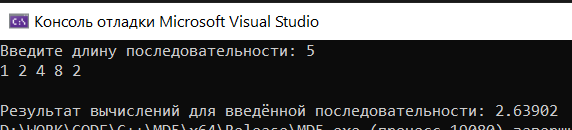
Структуры данных, используемые в программе, приведены в таблице 1.

Таблица 1 – Структуры данных в программе

|  |  |  |
| --- | --- | --- |
| **Имя** | **Тип** | **Предназначение** |
| n | int | Длина последовательности |
| a | double\* | Ссылка на динамический массив |
| res | double | Результат расчётов |

# **ДЕМОНСТРАЦИОННЫЙ ПРИМЕР ДЛЯ ЗАДАНИЯ №1**

Пусть дана последовательность длиной N = 5 и вида: {1, 2, 4, 8, 2}, тогда результат вычислений для неё равен .Результат работы программы изображен на рисунке 2.

  
Рисунок 2 – Пример работы программы

# **СХЕМА ПРОГРАММЫ ДЛЯ ЗАДАНИЯ №2**

Схема алгоритма программы, реализующей обработку двумерного массива по заданию варианта представлена на рисунке 3.

  
Рисунок 3 – Схема алгоритма программы, реализующей обработку двумерного массива

# **ТЕКСТ ПРОГРАММЫ ДЛЯ ЗАДАНИЯ №2**

Текст программы на языке программирования С/С++, реализующей обработку двумерного массива (квадратной матрицы) по заданию варианта представлен в листинге 2.

## **Листинг 2. Текст программы**

#include <iostream>

int main()

{

setlocale(LC\_ALL, "RUSSIAN");

int n;

int res = 0;

std::cout << "Введите размерность квадратной матрицы: "; std::cin >> n;

int\*\* a = new int\*[n];

for (int i = 0; i < n; i++){

a[i] = new int[n];

for (int j = 0; j < n; j++){

std::cin >> a[i][j];

}

## **Листинг 2. Текст программы (продолжение)**

if ((n %2 != 0 && i <= n / 2 ) || ( n%2 == 0 && i < n/2)) {

for (int k = 0; k < i + 1; k++){

res += a[i][k];

}

for (int k = n - 1; k >= n - i - 1; k--){

res += a[i][k];

}

}

else {

for (int k = 0; k < n - i; k++){

res += a[i][k];

}

for (int k = n - 1; k >= i; k--){

res += a[i][k];

}

}

}

if (n % 2 != 0) {

res -= a[n / 2][n / 2];

}

std::cout << "\nРезультат вычислений для введённой матрицы: " << res;

return 0;

}

**ИНСТРУКЦИЯ ПРОГРАММИСТА ДЛЯ ЗАДАНИЯ №2**

Структуры данных, используемые в программе, приведены в таблице 2.

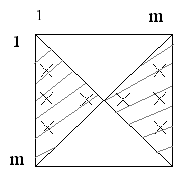
Таблица 2 – Структуры данных в программе

|  |  |  |
| --- | --- | --- |
| **Имя** | **Тип** | **Предназначение** |
| n | int | Размерность матрицы |
| a | int\*\* | Ссылка на динамический массив |
| res | int | Результат расчётов |

# **ДЕМОНСТРАЦИОННЫЙ ПРИМЕР ДЛЯ ЗАДАНИЯ №2**

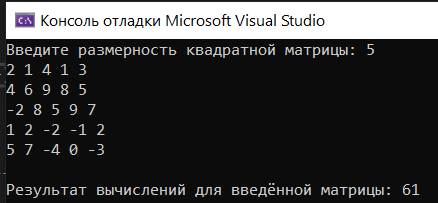
Пусть дана следующая квадратная матрица размером N = 5:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **2** | 1 | 4 | 1 | **3** |
| **4** | **6** | 9 | **8** | **5** |
| **-2** | **8** | **5** | **9** | **7** |
| **1** | **2** | -2 | **-1** | **2** |
| **5** | 7 | -4 | 0 | **-3** |



Для неё результат вычислений – сумма элементов, помеченных «х» как на картинке слева, будет равна 2 + 4 + 6 + (-2) + 8 + 5 + 1 + 2 + 5 + 3 + 8 + 5 + 9 + 7 + (-1) + 2 + (-3) = 61.

Результат работы программы изображен на рисунке 4.

  
Рисунок 4 – Пример работы программы

# **ВЫВОДЫ**

В ходе выполнения работы я ознакомилась с динамическими переменными языков С и С++. Так, переменные, созданные с помощью операции new, называются динамическими, потому что они создаются и уничтожаются в процессе выполнения программы. Динамические переменные размещаются в куче– динамической памяти. Для доступа к таким переменным используются только указатели (косвенная адресация), так как ячейкам динамической памяти нельзя присвоить имя.