

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«Казанский национальный исследовательский технический университет
им. А.Н. Туполева – КАИ»

Институт компьютерных технологий и защиты информации
Отделение СПО ИКТЗИ (Колледж информационных технологий)

ЛАБОРАТОРНАЯ РАБОТА №5

по дисциплине

Основы алгоритмизации и программирования

Тема: «Программирование алгоритмов с использованием динамического
выделения памяти»

Работу выполнил

Студент гр.4238

Бусов В.Р.

Принял

Преподаватель Шмидт. И.Р.

Казань 2023

ВАРИАНТ 4

Цель работы

Приобрести умения и практические навыки для программирования алгоритмов с использованием механизма динамического выделения памяти.

Задание на лабораторную работу

Задание 1. Написать программу, вычисляющую значения a^b и b^a . Расчет должен производиться в функции, которая получает данные по указателю и возвращает результат по указателю.

Задание 2. Написать программу, заменяющую положительные элементы массива на их квадраты. При вводе/выводе элементов использовать индексы, а обработку элементов массива осуществлять с помощью указателей. Расчет должен производиться в функции, в которую массив передается по указателю.

Задание 3. Написать программу, выполняющую расчет суммы двух неквадратных матриц. Матрицы должны храниться в памяти в виде двумерного динамического массива, размерности матриц вводятся пользователем с клавиатуры. Предусмотреть генерацию значений матриц как случайных вещественных чисел в диапазоне от -1.0 до 1.0.

Результат выполнения работы

Задание 1

Описание: Программа выводит приветственное сообщение, после чего принимает на вход два числа: a и b . Результат: два новых числа a в степени b и b в степени a .

```
Введите число a: 3
Введите число b: 5
243
125
D:\Documents\KIT\2 курс\лабы\ОАИП\5 лаба\ConsoleApplication1\x64\Debug\ConsoleApplication1.exe (процесс 3636) завершил работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" ->"Параметры" ->"Отладка" -> "Автоматически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно: _
```

Рисунок 1 - Результат выполнения задания 1

Задание 2

Описание: Программа выводит приветственное сообщение, после чего принимает на вход длину массива L. Далее происходит ввод L чисел. Числа ≤ 0 останутся неизменными, а числа > 0 возведутся в квадрат

```
Программа получает на вход длину массива L, после чего вводятся L элементов.
На выходе получается массив, где все элементы > 0 возведены в квадрат
Введите длину массива: 5
Введите 1 элемент массива: 0
Введите 2 элемент массива: -3
Введите 3 элемент массива: -1029
Введите 4 элемент массива: 2
Введите 5 элемент массива: 50
Результат: 0 -3 -1029 4 2500
D:\Documents\KIT\2 курс\лабы\ОАИП\5 лаба\ConsoleApplication2\x64\Debug\ConsoleApplication2.exe (процесс 9436) завершил работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" ->"Параметры" ->"Отладка" -> "Автоматически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно: _
```

Рисунок 2 - Результат выполнения задания 2

Задание 3

Описание: Программа выполняет сложение двух случайно сгенерированных матриц со значениями элементов от -1 до 1. Сначала вводится размер матрицы (количество строк и столбцов), после чего программа выводит исходные матрицы и результирующую.

```
Введите количество строк: 4
Введите количество столбцов: 5
Матрица A:
-0.3  0.2  -0.3  -0.8  0.2
0.4  0  0  0.1  0.3
-0.8  -0.6  0.3  -0.2  0.1
0.6  0.1  -0.6  0.4  0.1
Матрица B:
0.4  -0.2  -0.4  0.2  0.2
0.1  0.2  -0.2  -0.4  -0.5
-0.4  0.2  0.4  -0.2  -0.9
0.3  0.4  -0.1  0.1  0.1
Сложение матриц...
Результат:
0.1  0  -0.7  -0.6  0.4
0.5  0.2  -0.2  -0.3  -0.2
-1.2  -0.4  0.7  -0.4  -0.8
0.9  0.5  -0.7  0.5  0.2

D:\Documents\КИТ\2 курс\лабы\ОАИП\5 лаба\ConsoleApplication3\x64\Debug\ConsoleApplication3.exe (процесс 13404) завершил
работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" ->"Параметры" ->"Отладка" -> "Ав
томатически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно: █
```

Рисунок 3 - Результат выполнения задания 3

Листинг кода

Задание 1

```
#include <iostream>
using namespace std;

int calc_pow(int* a, int* b) {
    int i = pow(*a, *b);
    int j = pow(*b, *a);

    *a = i;
    *b = j;

    return 0;
}

int main() {
    setlocale(LC_ALL, "Russian");
    int a, b;

    cout << "Введите число a: ";
    cin >> a;
```

```

    cout << "Введите число b: ";
    cin >> b;

    int* ptr_a = &a;
    int* ptr_b = &b;

    calc_pow(ptr_a, ptr_b);
    cout << *ptr_a << endl << *ptr_b;

    return 0;
}

```

Задание 2

```

#include <iostream>
using namespace std;

auto calc(int *ind, int arr_len) {
    for (int i = 0; i < arr_len; i++) {
        if (*ind > 0) {
            *ind *= *ind;
        }
        ind++;
    }
    return 0;
}

int main() {
    setlocale(LC_ALL, "Russian");
    cout << "Программа получает на вход длину массива L, после чего вводятся L
элементов.\nНа выходе получается массив, где все элементы > 0 возведены в
квадрат";

    int arr_len;
    cout << endl << "Введите длину массива: ";
    cin >> arr_len;

    int* arr = new int[arr_len];

    for (int i = 0; i < arr_len; i++) {
        cout << "Введите " << i + 1 << " элемент массива: ";
        cin >> arr[i];
    }

    calc(&arr[0], arr_len);
    cout << "Результат: ";
    for (int i = 0; i < arr_len; i++) {
        cout << arr[i] << " ";
    }

    cout << "\n\n\n" << endl;
    return 0;
}

```

Задание 3

```
#include <iostream>
using namespace std;

int main()
{
    srand(time(NULL));

    setlocale(LC_ALL, "Russian");

    int n, m;
    cout << "Введите количество строк: ";
    cin >> n;
    cout << "Введите количество столбцов: ";
    cin >> m;

    double** arr1 = new double* [n];
    for (int i = 0; i < n; ++i)
        arr1[i] = new double[m];

    double** arr2 = new double* [n];
    for (int i = 0; i < n; ++i)
        arr2[i] = new double[m];

    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < m; j++)
        {
            double r1 = rand() % 10 - rand() % 10;
            double r2 = rand() % 10 - rand() % 10;
            arr1[i][j] = r1 / 10;
            arr2[i][j] = r2 / 10;
        }
    }

    cout << "Матрица A: \n";
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < m; j++)
        {
            cout << arr1[i][j] << " \t";
        }
        cout << "\n";
    }

    cout << "Матрица B: \n";
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < m; j++)
        {
            cout << arr2[i][j] << " \t";
```

```

    }
    cout << "\n";
}

cout << "Сложение матриц..." << endl;
cout << "Результат: \n";
for (int i = 0; i < n; i++) {
    for (int j = 0; j < m; j++) {
        cout << arr1[i][j] + arr2[i][j] << " \t ";
    }
    cout << endl;
}
}

```

Контрольные вопросы

- 1) *Указатель* – особый тип переменной, содержащий адрес ячейки памяти, в которой хранится значение переменной.
- 2) При статическом выделении памяти в массив невозможно добавить элементов больше, чем указано. Динамическое выделение памяти позволяет это сделать, так как при добавлении выделяет дополнительное место в памяти.
- 3) С помощью оператора разыменования и оператора получения адреса памяти (* и &);
- 4) Да, может. При объявлении функции необходимо рядом с типом возвращаемого значения поставить *.
- 5) Да, могут. В таком случае параметры передаются по ссылке.
- 6) Для создания динамических двумерных массивов, для итерации двумерных массивов.