Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

Специальность: Программная инженерия

Курс: 1

Группа: 10

Дисциплина: Основы алгоритмизации и программирования

 Реферат

На тему «n-мерный массив»

Выполнил:

Студент: Сегренёв Кирилл Сергеевич

Проверил:

Преподаватель: Белодед Н.И

2023, Минск

Содержание

[**ВВЕДЕНИЕ** 3](#_Toc152716295)

[**АЛГОРИТМ** 4](#_Toc152716296)

[**N-МЕРНЫЙ МАССИВ** 11](#_Toc152716297)

[**ВЫВОД** 13](#_Toc152716298)

**ВВЕДЕНИЕ**

На лекции 05.12 преподаватель озвучил следующее задание: Из двумерного массива сделать одномерный и наоборот. Реализовать алгоритм перевода n-мерного массива в одномерный.

**АЛГОРИТМ**

**Псевдокод:**

НАЧАЛО

ИНИЦИАЛИЗИРУЕМ константы для двумерного массива

ОБЪЯВЛЯЕМ массивы

НАЧАЛО цикла

Для каждого i равного 0 до m

НАЧАЛО цикла

Для каждого g равного 0 до n

ПРИСВАИВАЕМ двумерному массиву число от 0 до 99

ВЫВОДИМ данный элемент

КОНЕЦ цикла

КОНЕЦ цикла

НАЧАЛО цикла

Для каждого i равного 0 до m

НАЧАЛО цикла

Для каждого g равного 0 до n

ПРИСВАИВАЕМ одномерному массиву значение двумерного

ДОБАВЛЯЕМ к индексу одномерного массива 1

КОНЕЦ цикла

КОНЕЦ цикла

ОБНУЛЯЕМ переменную, отвечающую за индексы одном. Массива

НАЧАЛО цикла

Для каждого i равного 0 до m \* n

ВЫВОДИМ каждый элемент одном.массива

КОНЕЦ цикла

НАЧАЛО цикла

Для каждого i равного 0 до m

НАЧАЛО цикла

Для каждого g равного 0 до n

ПРИСВАИВАЕМ двум. массиву значение одном.

ДОБАВЛЯЕМ к индексу одномерного массива 1

КОНЕЦ цикла

КОНЕЦ цикла

НАЧАЛО цикла

Для каждого i равного 0 до m

НАЧАЛО цикла

Для каждого g равного 0 до n

ВЫВОДИМ каждый элемент двум.массива

КОНЕЦ цикла

КОНЕЦ цикла

КОНЕЦ

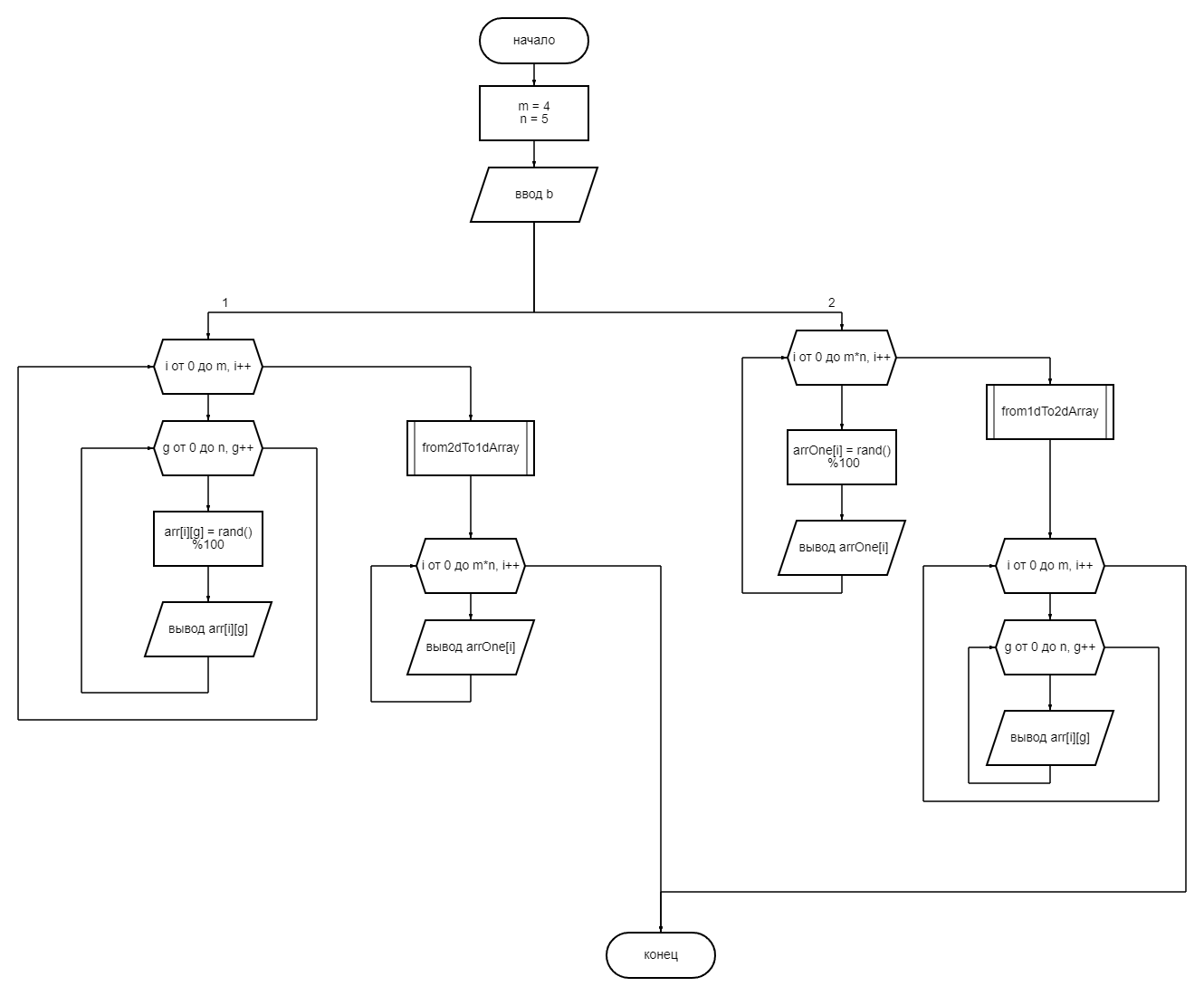
Данный псевдокод показывает, как можно перевести из двумерного массива в одномерный и наоборот с выводом на экран. Будет выводится изначальный двумерный массив, в который мы записали наши случайные значения, а уже после будет выводится одномерный массив с теми же значениями. В конце производится перевод одномерного массива в двумерным, поэтому те значения, которые были в самом начале программы, будут и в конце.

**Блок-схема:**

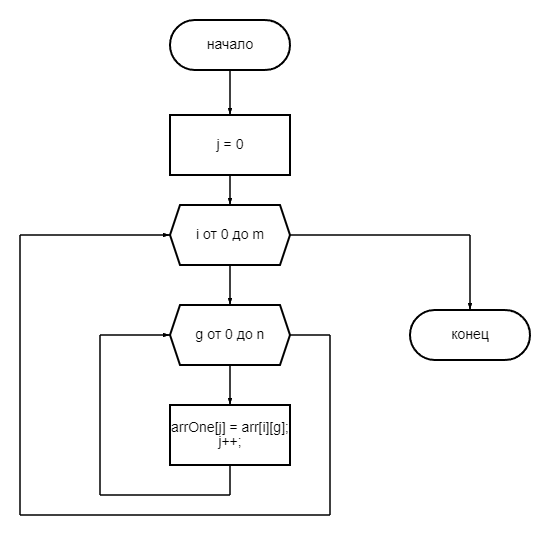
Псевдокод я писал, чтобы понимать, в каком направлении двигаться, поэтому он полностью не соответствует коду. Но он все еще остается правильным, если брать то, что я не буду использовать те функции, которые я использовал в блок-схеме и коде.

Блок-схема же полностью отображает мой код, который будет описан дальше.

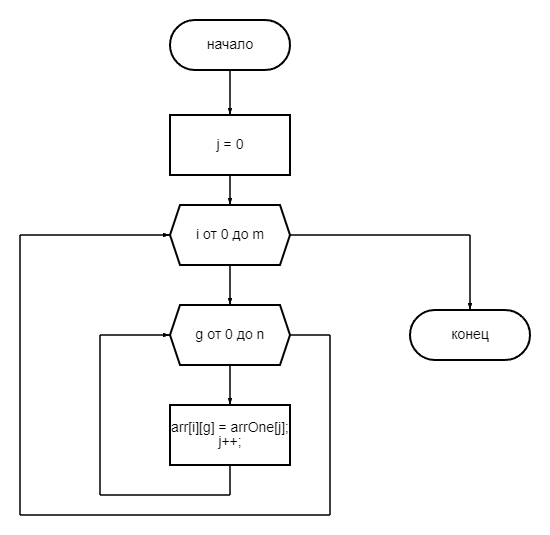
**Main()**



**from2dTo1dArray()**

****

**from1dTo2dArray()**

****

**Код программы:**

#include <iostream>

#include <stdio.h>

#include <ctime>

#include <iomanip>

void from2dTo1dArray(int arr[4][5], int\* arrOne, int m, int n) { // Функция для перевода двумерного массива в одномерный

int j = 0; // Переменная для определения индекса одномерного массива

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

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

arrOne[j] = arr[i][g]; // Каждому элементу одномерного массива присваиваем значение элемента двумерного массива

j++;

}

}

}

void from1dTo2dArray(int arr[4][5], int\* arrOne, int m, int n) { // Функция для перевода одномерного массива в двумерный

int j = 0;

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

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

arr[i][g] = arrOne[j]; // Каждому элементу двумерного массива присваиваем значение элемента одномерного массива

j++;

}

}

}

int main() {

setlocale(LC\_ALL, "ru");

srand(time(0));

const int m = 4, n = 5;

int arr[m][n];

int arrOne[m \* n];// Важно заметить, что размером одномерного массива будет являться произведение двух размеров двумерного массива

// Данное произведение используется не только в инициализации массива, но и в реализации цикла(чтобы не залезать за пределы выделенной памяти)

int b;

std::cout << "1 - Перевод двумерного массива в одномерный.\n2 - Перевод одномерного массива в двумерный.\n";

std::cin >> b;

system("cls"); // После принятого предложения стираем из консоли данные строки(чтобы не мешали)

switch (b) {

case 1:

std::cout << "Двумерный массив:\n";

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

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

arr[i][g] = rand()% 100; // Записываем случайные значения от 0 до 99 для наглядности своих действий

std::cout << std::setw(3) << arr[i][g] << " "; // Выводим начальный массив

}

std::cout << "\n";

}

from2dTo1dArray(arr, arrOne, m, n); // Заходим в функцию, в которой преобразуем двумерный массив в одномерный

std::cout << "\nОдномерный массив, получившийся из двумерного:\n"; // Выводим одномерный массив, который получили из двумерного

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

std::cout << arrOne[i] << " ";

}

std::cout << "\n";

break;

case 2:

std::cout << "Одномерный массив:\n"; // Записываем в одномерный массив случайные значения от 0 до 99 и выводим данные значения на экран

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

arrOne[i] = rand() % 100;

std::cout << arrOne[i] << " ";

}

std::cout << "\n";

from1dTo2dArray(arr, arrOne, m, n); // Заходим в функцию, в которой преобразуем одномерный массив в двумерный

std::cout << "\nДвумерный массив, получившийся из одномерного:\n"; // Выводим преобразованный двумерный массив

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

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

std::cout << std::setw(3) << arr[i][g] << " ";

}

std::cout << "\n";

}

break;

default:

std::cout << "Вы выбрали не тот пункт((\n";

return(EXIT\_FAILURE);

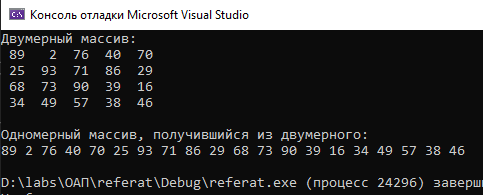
}

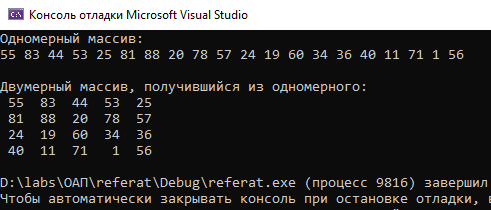
return 0;

}

Код программы я решил реализовать через выбор(Из одномерного в двумерный, либо из двумерного в одномерный)

**Output:**

Перевод из двумерного в одномерный:  


Перевод из одномерного в двумерный:  


**N-МЕРНЫЙ МАССИВ**

Чтобы понять, как из n-мерного массива сделать массив, нужно найти закономерность. Нужно узнать, что будет происходить с увеличением мерности массива. Попробуем написать функцию для перевода из трехмерного массива в одномерный.

Я возьму только одну функцию, но это нужно применять и для вывода на экран.

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

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

for(int k = 0; k < a; k++)

{

arrOne[j] = arr[i][g][k];

j++;

}

}

} // Мы допустили, что ранее объявили массивы и переменные m, j, n и a

Помним, что размер одномерного массива определяется произведением размеров многомерного массива(m\*n\*a)

Алгоритм преобразования трехмерного массива в одномерный теперь использует 3 цикла, в то время как в двухмерном массиве использовалось 2.  
Из этого можно сделать вывод, что при увеличении мерности массива увеличивается и количество вложенных циклов.

Размерность одномерного массива при переводе из n-мерного:

arrOne[размер1 \* размер2 \*. . . \*размерN)

Можем написать код в общем виде:

for(переменная1 = 0; переменная1 < размер1; переменная1++){

for(переменная2 = 0; переменная2 < размер2; переменная2++){

.

.

.

for(переменнаяN = 0; переменнаяN < размерN; переменнаяN++){

arrOne[j] = arr[размер1][размер2]. . . [размерN];

j++;

}

}

}

Перевод в n-мерный будет выглядеть точно так же по строению цикла, но в самом теле цикла элементы одномерного массива будут присваиваться многомерному массиву.

arr[размер1][размер2]. . . [размерN] = arrOne[j];

j++;

**ВЫВОД**

Нужно помнить, что если идет перевод из многомерного массива в одномерный или наоборот, то длина массива равна произведению длин многомерного массива.

Так же с увеличение мерности массива(двумерный, трехмерный и тд.) при переводе в одномерный будет и увеличиваться количество вложенных циклов.

Чтобы не заходить за пределы памяти при присвоении переменных одномерному массиву, нужно ввести отдельный счетчик индексов, который будет отличаться от индексов n-мерного массива.