ГУАП

КАФЕДРА № 43

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

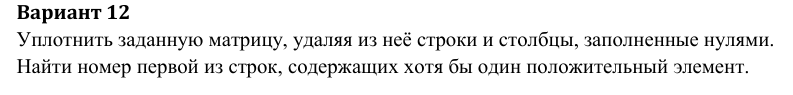
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ассистент |  |  |  | М. А. Мурашова |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ |
| Обработка числовых матриц |
| по курсу: ОСНОВЫ ПРОГРАММИРОВАНИЯ |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4135К |  |  |  | Столяров Н.С. |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2022

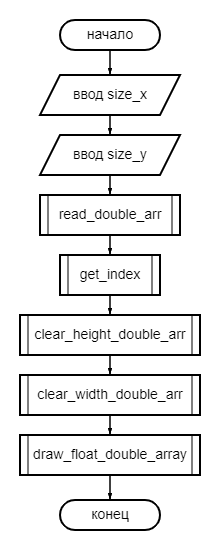
**1.Цель работы:** Целью работы является изучение структуры данных двумерный массив.  
**2.Задачи работы:**  
  
**3.Описание функций:  
1)Имя: main**   
**Назначение:** Вычисление номера первой строки содержащей положительный элемент. Удаление столбцов и строк заполненными нулями.  
**Входные данные:** Высота, ширина и массив.   
**Выходные данные:** Индекс и массив.  
**Побочный эффект:** отсутствует.   
**Тестовые данные:**

|  |  |  |  |
| --- | --- | --- | --- |
| Ширина | Высота | Массив | Ответ |
| 4 | 4 | 1 0 3 4 0 0 0 0 4 0 2 1 5 0 7 8 | Индекс первой строки содержащей положительный элемент: 0 1       3       4  4 2 1  5 7 8 |

**Прототип:** int main()  
**Псевдокод:**  
Ввод ширины и высоты массивами  
Ввод массива  
Нахождение номера строки содержащей хотя бы 1 положительный элемент  
Вывод номера строки  
Удаление из массива строк состоящих из нулей  
Удаление из массива столбцов состоящих из нулей

Вывод очищенного от нулей массива

**Блок-схема:**



**4. Имя: draw\_float\_double\_array**  
**Назначение:** Вывод двумерного массива в консоль.  
**Входные данные:** arr, size\_x, size\_y.   
**Выходные данные:** массив.  
**Побочный эффект:** отсутствует.   
**Прототип:** draw\_float\_double\_array ()

|  |  |
| --- | --- |
| **Псевдокод:** | **Блок-схема:** |
| Вывод массива в консоль | C:\Users\nikit\AppData\Local\Microsoft\Windows\INetCache\Content.Word\diagram.png |

**5. Имя: read\_size\_arr**  
**Назначение:** Проверка на размер массива.  
**Входные данные:** promt(комментарий).   
**Выходные данные:** size(размер массива).  
**Побочный эффект:** отсутствует.   
**Прототип:** read\_size\_arr ()

|  |  |
| --- | --- |
| **Псевдокод:** | **Блок-схема:** |
| Бесконечный цикл  Ввод значения  Если значение больше 1, то  Останавливаем цикл  Если нет  Заново вводим | C:\Users\nikit\Desktop\diagram (1).png |

**6. Имя: get\_index**  
**Назначение:** нахождение строки содержащей хотя бы 1 положительный элемент.  
**Входные данные:** arr, size\_x, size\_y.   
**Выходные данные:** y  
**Побочный эффект:** отсутствует.   
**Прототип:** int get\_index ()

|  |  |
| --- | --- |
| **Псевдокод:** | **Блок-схема:** |
| Нахождение первой строки с положительным элементом.  Если такой строки не существует, возвращаем -1  Если есть индекс данной строки | C:\Users\nikit\AppData\Local\Microsoft\Windows\INetCache\Content.Word\get_index.png |

**7. Имя: read\_double\_arr**  
**Назначение:** ввод элементов массива  
**Входные данные:** arr, size\_x, size\_y.   
**Выходные данные:** arr.  
**Побочный эффект:** отсутствует.   
**Прототип:** double read\_double\_arr ()

|  |  |
| --- | --- |
| **Псевдокод:** | **Блок-схема:** |
| Ввод элементов массива пока не остановятся циклы | C:\Users\nikit\AppData\Local\Microsoft\Windows\INetCache\Content.Word\read_double_arr.png |

**8. Имя: clear\_height\_double\_arr**  
**Назначение:** проверка переменной a на корректность ввода  
**Входные данные:** arr, size\_x, size\_y, new\_size\_y.   
**Выходные данные:** new\_arr\_y.  
**Побочный эффект:** отсутствует.   
**Прототип:** clear\_height\_double\_arr()

|  |  |
| --- | --- |
| **Псевдокод:** | **Блок-схема:** |
| Цикл по массиву  если в строке массива есть не 0, то  добавляем это строку в новый масив  после цикла возвращаем получившийся массив |  |

**9. Имя: clear\_width\_double\_arr**  
**Назначение:** проверка переменной a на корректность ввода  
**Входные данные:** arr, size\_x, size\_y, new\_size\_x.   
**Выходные данные:** new\_arr\_x.  
**Побочный эффект:** отсутствует.   
**Прототип:** clear\_width\_double\_arr()

|  |  |
| --- | --- |
| **Псевдокод:** | **Блок-схема:** |
| Цикл по массиву  если в столбце массива есть не 0, то  добавляем это столбец в новый масив  после цикла возвращаем получившийся массив |  |

**10.ЛИСТИНГ КОДА:**

**main.cpp  
/\***

**Задания на лабораторную работу приводятся в каждом варианте. При написании программ**

**можно использовать как динамические, так и нединамические массивы. Размерность последних**

**задаѐтся именованной константой**

**Вариант 12**

**Уплотнить заданную матрицу, удаляя из неѐ строки и столбцы, заполненные нулями.**

**Найти номер первой из строк, содержащих хотя бы один положительный элемент.**

**\*/**

**#define RANDOM\_NUMS false**

**#include <iostream>**

**using namespace std;**

**#include "libs/lib.h"**

**#include <cmath>**

**#include <time.h>**

**// проверка ввода**

**#include "libs/simple\_char.h"**

**#include "libs/input\_validation.h"**

**// работа с массивами**

**#include "libs/array.h"**

**#include "clear\_arr.h"**

**int read\_size\_arr(const char \*promt = "") {**

**int size;**

**while (true) {**

**size = read\_value(promt, false, false, false);**

**if (size > 0) {**

**break;**

**} else {**

**cout << "Размер должен быть больше 1." << endl;**

**}**

**}**

**return size;**

**}**

**int main() {**

**// смена кодировки**

**system("chcp 65001");**

**// очистка терминала**

**clear\_scr();**

**// рандом**

**srand(time(NULL));**

**int x, y;**

**int size\_x, size\_y;**

**// ввод размеров массива**

**while (true) {**

**size\_x = read\_size\_arr("Ширина массива: ");**

**size\_y = read\_size\_arr("Высота массива: ");**

**if (size\_x < 1 || size\_y < 0) {**

**cout << "Массив не может содержать 0 элементов." << endl;**

**} else break;**

**}**

**// создаём новый массив**

**double \*\*arr = (double\*\*)malloc(size\_y \* sizeof(double\*));**

**for(int i = 0; i < size\_y; i++) {**

**arr[i] = (double\*)malloc(size\_x \* sizeof(double));**

**}**

**// ввод значений массива**

**arr = read\_double\_arr(arr, size\_x, size\_y, RANDOM\_NUMS);**

**draw\_line(20);**

**draw\_float\_double\_array(arr, size\_x, size\_y);**

**draw\_line(20);**

**// Найти номер первой из строк, содержащих хотя бы один положительный элемент.**

**get\_index(arr, size\_x, size\_y);**

**draw\_line(20);**

**// Уплотнить заданную матрицу, удаляя из неѐ строки и столбцы, заполненные нулями.**

**int size\_y\_new\_arr;**

**// уплотняем по высоте**

**arr = clear\_height\_double\_arr(arr, size\_x, size\_y, &size\_y\_new\_arr);**

**int size\_x\_new\_arr;**

**// уплотняем по ширине**

**arr = clear\_width\_double\_arr(arr, size\_x, size\_y\_new\_arr, &size\_x\_new\_arr);**

**// выводим новый массив**

**draw\_float\_double\_array(arr, size\_x\_new\_arr, size\_y\_new\_arr);**

**// очистка памяти**

**for(y = 0; y < size\_y; y++) {**

**free(arr[y]);**

**}**

**free(arr);**

**return 0;**

**}**

**clear\_arr.h**

#include <iostream>

#include <cmath>

using namespace std;

double \*\*clear\_height\_double\_arr(double \*\*arr, int size\_x, int size\_y, int \*new\_size\_y) {

int x, y;

double \*\*new\_arr\_y = (double\*\*)malloc(sizeof(double\*));

(\*new\_size\_y) = 0;

bool ok = false;

for (y = 0; y < size\_y; y++) {

ok = false;

for (x = 0; x < size\_x; x++) {

if (arr[y][x] != 0) {

ok = true;

break;

}

}

if (ok) {

new\_arr\_y = (double\*\*) realloc(new\_arr\_y, (++(\*new\_size\_y)) \* sizeof(double\*));

new\_arr\_y[(\*new\_size\_y)-1] = (double\*)malloc(size\_x \* sizeof(double));

for (int j = 0; j < size\_x; j++) {

new\_arr\_y[(\*new\_size\_y)-1][j] = arr[y][j];

}

}

}

return new\_arr\_y;

}

double \*\*clear\_width\_double\_arr(double \*\*arr, int size\_x, int size\_y, int \*new\_size\_x) {

int x, y;

bool ok;

double \*\*new\_arr\_x = (double\*\*)malloc(size\_y \* sizeof(double\*));

(\*new\_size\_x) = 0;

for (x = 0; x < size\_x; x++) {

ok = false;

for (y = 0; y < size\_y; y++) {

if (arr[y][x] != 0) {

ok = true;

break;

}

}

if (ok) {

(\*new\_size\_x)++;

for (int j = 0; j < size\_y; j++) {

new\_arr\_x[j] = (double\*) realloc(new\_arr\_x[j], ((\*new\_size\_x)) \* sizeof(double));

new\_arr\_x[j][(\*new\_size\_x)-1] = arr[j][x];

}

}

}

return new\_arr\_x;

}

**arrray.h**

#include <iostream>

#include <cmath>

using namespace std;

void draw\_float\_array(double \*arr, int size) {

for (int i = 0; i < size; i++)

cout << arr[i] << "\t";

cout << endl;

}

void draw\_float\_double\_array(double \*\*arr, int size\_x, int size\_y) {

for (int y = 0; y < size\_y; y++) {

for (int x = 0; x < size\_x; x++) {

cout << arr[y][x] << "\t";

}

cout << endl;

}

}

// Найти номер первой из строк, содержащих хотя бы один положительный элемент.

int get\_index(double \*\*arr, int size\_x, int size\_y) {

int x, y;

bool find = false;

for (y = 0; y < size\_y; y++) {

for (x = 0; x < size\_x; x++) {

if (arr[y][x] > 0) {

cout << "Индекс первой строки содержащей положительный элемент: " << y << endl;

find = true;

break;

}

}

if (find) break;

}

if (!find) {

cout << "Строк с положительными элементами не существует." << endl;

return -1;

} else {

return y;

}

}

double \*\*read\_double\_arr(double \*\*arr, int size\_x, int size\_y, bool random = false) {

int x, y;

for (y = 0; y < size\_y; y++)

for (x = 0; x < size\_x; x++) {

if (random) {

arr[y][x] = random\_int(-10, 10);

} else {

cout << "Array[" << y << "][" << x << "] = ";

arr[y][x] = read\_value("", true, true, false);

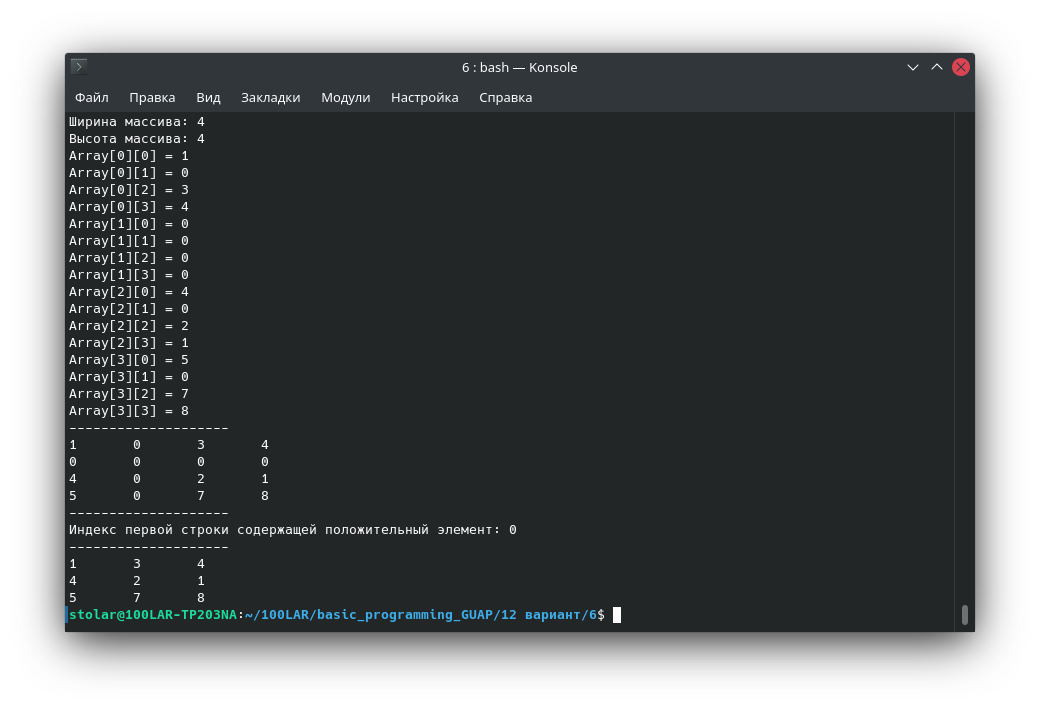
}

}

return arr;

}

**10.Пример выполнения программы:**



Видно, что результаты расчётов совпадают с тестовыми данными.

**11.Анализ результатов и выводы:**

В ходе этой лабораторной работы я устал перебирать динамические массивы.  
Из плюсов можно отметить её работоспособность.

Из минусов ужасный ввод элементов массива. Пока её протестируешь, уже можно стать дедом.