**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное автономное образовательное учреждение высшего профессионального образования

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Инженерная школа информационных технологий и робототехники

Направление 15.03.04 «Автоматизация технологических процессов и производств»

Отделение автоматизации и робототехники

Работа с различными двумерными массивами

Наименование лабораторной работы

Отчет по лабораторной работе № 3

по дисциплине «Основы программирования и алгоритмизации»

Наименование учебной дисциплины

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Выполнил студент гр. 8Т32 |  | C:\Users\pavlu\OneDrive\Рабочий стол\Подпись.jpg |  |  |  | П. П. Гаврильев |
|  |  | Подпись |  | Дата |  | И.О. Фамилия |
|  |  |  |  |  |  |  |
| Проверил ассистент ОАР |  |  |  |  |  | А. В. Павловский |
| Должность |  | Подпись |  | Дата |  | И.О. Фамилия |

Томск – 2024 г.

**Цель работы:**

Ознакомиться и реализовать решение для двумерных массивов на практике на языке программирования С++.

**Ход работы:**

Таблица 1 – Формулировка заданий

|  |  |
| --- | --- |
| **№** | Формулировка |
| **1** | Даны целые положительные числа M, N, число D и набор из M чисел. Сформировать матрицу размера M × N, у которой первый столбец совпадает с исходным набором чисел, а элементы каждого следующего столбца равны сумме соответствующего элемента предыдущего столбца и числа D (в результате каждая строка матрицы будет содержать элементы арифметической прогрессии). |
| **2** | Дана квадратная матрица A порядка M. Найти среднее арифметическое элементов каждой ее диагонали, параллельной главной (начиная с одноэлементной диагонали). |
| **3** | Дана матрица размера M × N. Поменять местами столбец с номером N и первый из столбцов, содержащих только отрицательные элементы. Если требуемых столбцов нет, то вывести матрицу без изменений. |

Задание 1.

В данном задании поставлена задача сформировать матрицу m × n, у которой первый столбец совпадает с набором чисел m, а последующие столбцы больше предыдущего на d. Для реализации этой задачи были спроектированы блок схемы, представленные на рисунках 1.1, 1.2 и 1.3. По данным блок схемам были написаны программа, которая представлены в листингах 1.1 и 1.2. Результат выполнения программы представлен на рисунке 1.4.

Программа запрашивает размер массива, после которого вводятся числа набора m и число d на которое будут увеличиваться последующие столбцы. Для реализации поставленной задачи была реализована функция masssum, также для проверки ввода чисел была реализована функция incorrectioninput, которая дальше будет использоваться в следующих заданиях. После вызова функции masssum на консоль пользователя выводится конечный массива.

|  |  |
| --- | --- |
|  |  |
| a) | б) |

Рисунок 1.1 – Блок схема программы

|  |  |
| --- | --- |
|  |  |
| a) | б) |

Рисунок 1.2 – Блок схема функции incorrectioninput (a, b)

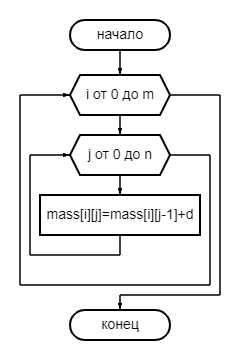


Рисунок 1.3 – Блок схема функции masssum(mass, d, m, n)

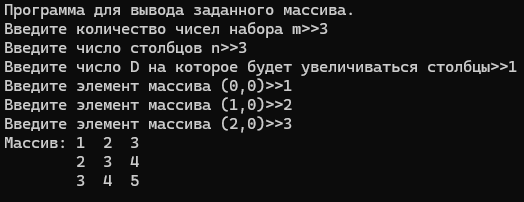


Рисунок 1.4 – Итог запуска программы

Листинг 1.1 – Код программы

|  |
| --- |
| #include <iostream>  #include "incorrectinput.h"  #include "masssum.h"  int main()  {  int m = 0, n = 0, d=0;  setlocale(LC\_ALL, "ru");  std::cout << "Программа для вывода заданного массива."<<std::endl;  std::cout << "Введите количество чисел набора m>>";  std::cin >> m;  m=incorrectinput(m,1);  std::cout << "Введите число столбцов n>>";  std::cin >> n;  n = incorrectinput(n,1);  std::cout << "Введите число D на которое будет увеличиваться столбцы>>";  std::cin >> d;  d = incorrectinput(d, 1);  int\*\* mass = new int\* [m];  for (int i = 0; i < m; i++) {  mass[i] = new int[n];  }  for (int i = 0; i < m; i++) {  std::cout << "Введите элемент массива " << "(" << i<<",0)>>";  std::cin >> mass[i][0];  \*mass[i] = incorrectinput(\*mass[i], 2);  }  masssum(mass, d, m, n);  std::cout << "Массив: ";  for (int i = 0; i < m; i++) {  for (int j = 0; j < n; j++) {  std::cout << mass[i][j]<<" ";  }  std::cout << '\n'<<'\t';  }  for (int i = 0; i < m; i++) {  delete[] mass[i];  }  delete[] mass;  } |

Листинг 1.2 - Библиотека incorrectioninput(a, b)

|  |
| --- |
| #pragma once  void incorrectinput(float a, int b)  {  if (b==1){  if (std::cin.fail()) {  std::cout << "Ошибка ввода. Повторите ввод>>";  std::cin.clear();  std::cin.ignore(100, '\n');  std::cin >> a;  }  while (a < 0) {  std::cout << "Количество элементов не может быть отрицательным. Повторите ввод>>";  std::cin >> a;  }  while (a == 0) {  std::cout << "Количество элементов не может быть равным 0. Повторите ввод>>";  std::cin >> a;  }    }  if (b == 2) {  if (std::cin.fail()) {  std::cout << "Ошибка ввода. Повторите ввод>>";  std::cin.clear();  std::cin.ignore(100, '\n');  std::cin >> a;  }  }  } |

Листинг 1.3 - Библиотека masssum(mass, d, m, n)

|  |
| --- |
| #pragma once  void masssum(int\*\* mass, int d, int m, int n)  {  for (int i = 0; i < m; i++) {  for (int j = 1; j < n; j++) {  mass[i][j] = mass[i][j - 1] + d;  }  }    } |

Задание 2.

В данном задании поставлена задача найти среднее арифметическое диагоналей параллельных главной. Для реализации этой задачи были спроектированы блок схемы, представленные на рисунках 2.1 и 2.2. По данным блок схемам были написаны программа, которая представлены в листингах 2.1 и 2.2. Результат выполнения программы представлен на рисунке 2.3.

Программа запрашивает размер n квадратного массива, после которого вводятся элементы массива. Для нахождения средней арифметической диагоналей реализована функция diagonalaverage. Также для проверки введенных данных использована функция incorrectioninput из первого задания. После вызова функции на консоль пользователя выводится массив из средних арифметических диагоналей.

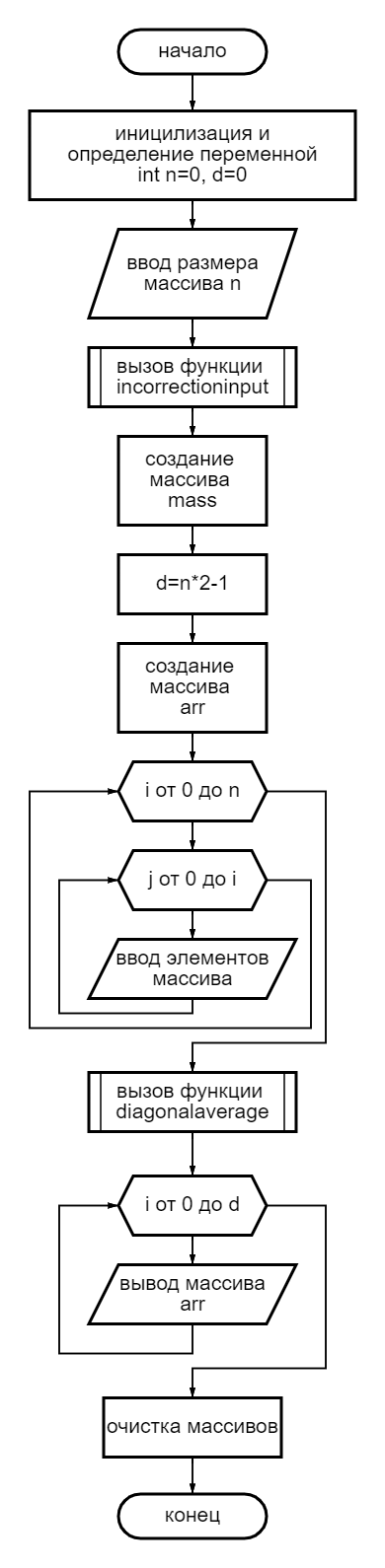


Рисунок 2.1 – Блок схема программы

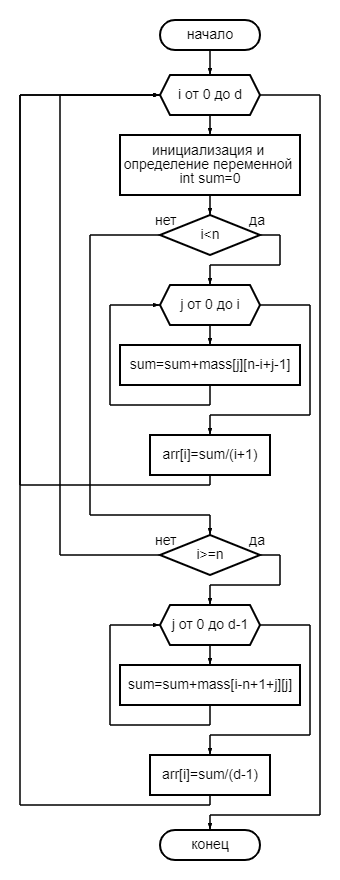


Рисунок 2.2-Блок схема функции diagonaaverage(mass, arr, d, n)

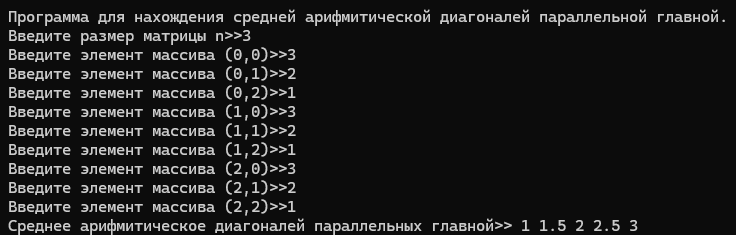


Рисунок 2.2 – Итог запуска программы

Листинг 2.1 – Код программы

|  |
| --- |
| #include <iostream>  #include "incorrectioninput.h"  #include "diagonalaverage.h"  int main()  {  setlocale(LC\_ALL, "ru");  int n = 0, d=0;  std::cout << "Программа для нахождения средней арифмитической диагоналей параллельной главной." << std::endl;  std::cout << "Введите размер матрицы n>>";  std::cin >> n; //ввод размерности массива  n = incorrectioninput(n, 1); //проверка  d = n \* 2 - 1;//количество диагоналей  float\* arr = new float [d]; //создание массивов  float\*\* mass = new float\* [n];  for (int i = 0; i < n; i++) {  mass[i] = new float[n];  }  for (int i = 0; i < n; i++) { //заполнение массива  for (int j = 0; j < n; j++) {  std::cout << "Введите элемент массива " << "(" << i<<","<<j << ")>>";  std::cin >> mass[i][j];  mass[i][j] = incorrectioninput(mass[i][j], 2);  }  }    diagonalaverage(mass, arr, d, n); //вычисление среднего арифмитического диагоналей    std::cout << "Среднее арифмитическое диагоналей параллельных главной>> "; //вывод ср. арифм. диагоналей  for (int i = 0; i < d; i++) {  std::cout << arr[i]<<" ";  }  for (int i = 0; i < n; i++) { //очистка массивов  delete[] mass[i];  }  delete[] mass;  delete[] arr;  } |

Листинг 2.2-библиотека diagonalaverage

|  |
| --- |
| #pragma once  void diagonalaverage(float\*\* mass, float\* arr, int d, int n)  {  for (int i = 0; i < d; i++) { //вычисление среднего арифмитического диагоналей  float sum = 0;  if (i < n) {  for (int j = 0; j <= i; j++) {  sum = sum + mass[j][n - i + j - 1];  }  arr[i] = sum / (i + 1);  }  if (i >= n) {  for (int j = 0; j < d - i; j++) {  sum = sum + mass[i - n + 1 + j][j];  }  arr[i] = sum / (d - i);  }  }    } |

Задание 3.

В данном задании поставлена задача реализовать замену выбранного пользователем столбца на первый отрицательный столбец. Для реализации этой задачи были спроектированы блок схемы, представленные на рисунках: 3.1, 3.2. По данным блок схемам были написаны программа, которая представлена в листингах 3.1 и 3.2. Результат выполнения программы представлен на рисунке 3.3.

Программа запрашивает размер массива, после которого вводятся элементы массива. Для замены столбцов реализована функция columnreplace. Также для проверки введенных данных использована функция incorrectioninput из первого задания. После вызова функции на консоль пользователя выводится массив с замененными столбцами.

|  |  |
| --- | --- |
|  |  |
| а) | б) |

Рисунок 3.1 – Блок схема программы

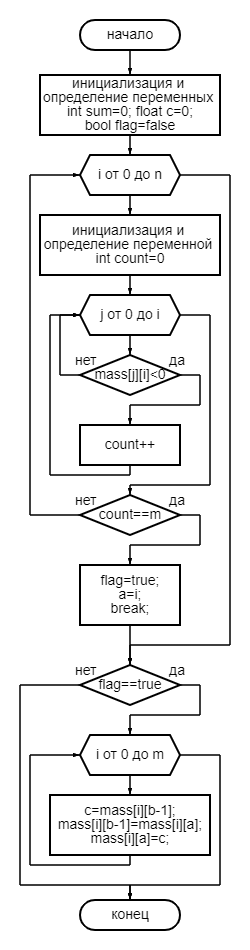


Рисунок 3.2 – Блок схема функции columnreplace(mass, m, n, b)

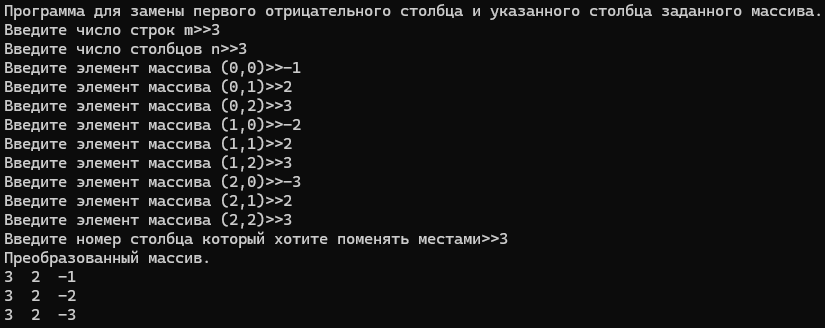


Рисунок 3.4 – Итог запуска программы

Листинг 3.1 – Код программы 3

|  |
| --- |
| #include <iostream>  #include "incorrectioninput.h"  #include "columnreplace.h"  int main()  {  setlocale(LC\_ALL, "ru");  int m = 0, n = 0, a=0, b=0;    std::cout << "Программа для замены первого отрицательного столбца и указанного столбца заданного массива." << std::endl;  std::cout << "Введите число строк m>>";  std::cin >> m;  m = incorrectioninput(m, 1); //проверка m  std::cout << "Введите число столбцов n>>";  std::cin >> n;  n = incorrectioninput(n, 1); //проверка n  float\*\* mass = new float\* [m];  for (int i = 0; i < m; i++) {  mass[i] = new float[n];  }  for (int i = 0; i < m; i++) { //заполнение массива  for (int j = 0; j < n; j++) {  std::cout << "Введите элемент массива " << "(" << i << "," << j << ")>>";  std::cin >> mass[i][j];  mass[i][j] = incorrectioninput(mass[i][j], 2);  }  }  std::cout << "Введите номер столбца который хотите поменять местами>>";//ввод номера столбца  std::cin >> b;  while (b <= 0) {  std::cout << "Номер столбца не может быть отрицательным или равным 0. Введите еще раз>>";  std::cin >> b; b = incorrectioninput(b, 2);  }  while (b > n) {  std::cout << "Номер столбца не может быть больше количества столбцов. Введите еще раз>>";  std::cin >> b; b = incorrectioninput(b, 2);  }  columnreplace(mass, m, n, b); |

Продолжение листинга 3.1

|  |
| --- |
| std::cout << "Преобразованный массив." << std::endl; //вывод преобразованного массива  for (int i = 0; i < m; i++) {  for (int j = 0; j < n; j++) {  std::cout << mass[i][j] << " ";  }  std::cout << '\n';  }  for (int i = 0; i < m; i++) { //очистка массива  delete[] mass[i];  }  delete[] mass;  } |

Листинг 3.2 – библиотека columnreplace

|  |
| --- |
| #pragma once  void columnreplace(float\*\* mass, int m, int n, int b)  {  int a = 0;  bool flag = false;  float c = 0;  for (int i = 0; i < n; i++) { //нахождение первого отрицательного столбца  int count = 0;  for (int j = 0; j < m; j++) {  if (mass[j][i] < 0) {  count++;  }  }  if (count == m) {  flag = true;  a = i;  break;  }  }  if (flag == true) { //замена стобцов  for (int i = 0; i < m; i++) {  c = mass[i][b - 1];  mass[i][b - 1] = mass[i][a];  mass[i][a] = c;  }  }    } |

**Вывод**

В ходе лабораторной работы была изучены информация по работе с двумерными массивами. На основе этой информации была проделана работа по написанию кода, представленных на листингах (1.1, 1.2, 1.3, 2.1, 2.2, 3.1, 3.2), для заданий. Также были сделаны блок схемы, которые показаны на рисунках (1.1, 1.2, 1.3, 2.1, 2.2, 3.1, 3.2). В итоге благодаря этой информации можно реализовать различные задания с обработкой двумерных массивов.