Бюджетное учреждение высшего образования   
Ханты-Мансийского автономного округа   
«Сургутский государственный университет»

Политехнический институт

Кафедра автоматики и компьютерных систем

**Отчет**

по лабораторной работе № 5  
«Многомерные массивы»

Выполнил: Фаттахов М.Р

студент группы 609-31

Проверил: Гришмановский П. В.

доцент кафедры автоматики и компьютерных систем

Сургут

2023 г.

Цель работы: закрепление теоретических знаний и  
приобретение практических навыков использования многомерных массивов на примере двухмерных..

Задание: Требуется решить задачу в соответствии с вариантом индивидуального задания, выданным преподавателем, используя многомерные массивы.

Вариант 27. Задано количество строк и столбцов матрицы, действительные значения ее элементов и размеры прямоугольной области. Найти координаты начала той области, сумма значений элементов которой  
максимальна, и значение этой суммы.

**Вариант 27**

**Формальное описание задачи.**

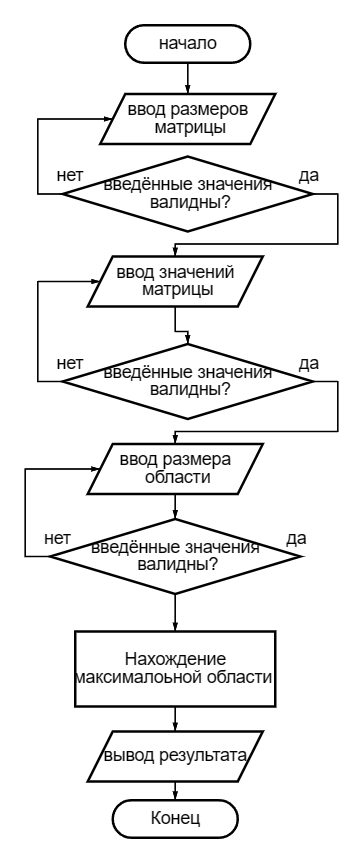
Пройтись по всем возможным вариантам области матрицы(от 0 координаты до количество строк/столбцов матрицы - количество строк/столбцов области), вычислить сумму чисел каждой, сравнить сохранённую сумму с каждой и если новая сумма больше предыдущих записать её как максимальную, сохранить координаты и после прохождения по всем вариантам вывести максимальную сумму и опираясь на сохранённые координаты вывести область на экран.

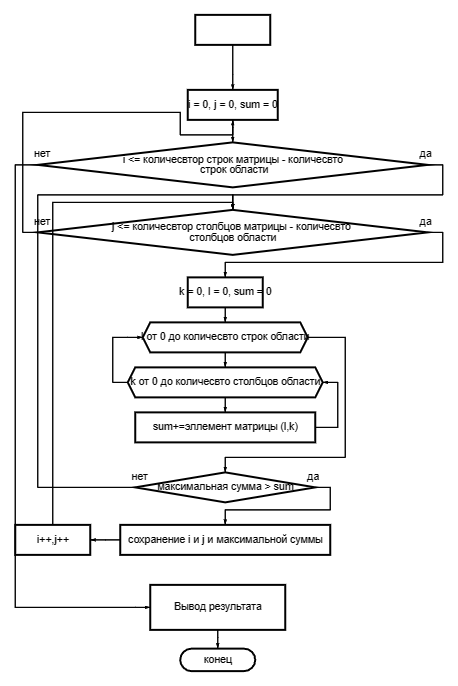
**Алгоритм программы.**

Для решения данной задачи необходимо:

1. Получить введённые данные.
2. Проверить валидность введённых данных.
3. Пройтись по всем вариантам области матрицы
4. Вычислить сумму чисел области
5. Найти максимальную сумму
6. Сохранить позицию области с максимальной суммой
7. Вывести область.

**Блок-схема алгоритма программы.**



****

**Листинг программы.**

#include <stdio.h>

#include <stdlib.h>

long double input\_arr(long double \*arr, int \*str\_count, int \*stolb\_count);

long double input\_size\_oblast(int str\_count, int stolb\_count, int \*size\_str\_o,

                      int \*size\_stolb\_o);

long double find\_max\_oblast(long double \*arr, int str\_count, int stolb\_count, int size\_str\_o,

                    int size\_stolb\_o, int \*max\_i, int \*max\_j, long double \*max\_sum);

long double calc\_sum(long double \*arr, int size\_stolb\_o, int size\_str\_o, int i, int j, long double \*sum,

             int stolb\_count);

long double output(long double \*arr, int size\_str\_o, int size\_stolb\_o, int max\_i, int max\_j,

           long double max\_sum, int stolb\_count);

int main() {

  long double max\_sum = 0, arr[225];

  int str\_count = 0, stolb\_count = 0, size\_str\_o, size\_stolb\_o, i = 0, j = 0, max\_i = 0, max\_j = 0;

  system("chcp 1251 > NUL");

  input\_arr(arr, &str\_count, &stolb\_count);

  input\_size\_oblast(str\_count, stolb\_count, &size\_str\_o, &size\_stolb\_o);

  find\_max\_oblast(arr, str\_count, stolb\_count, size\_str\_o, size\_stolb\_o, &max\_i,

                  &max\_j, &max\_sum);

  printf("Для матрицы:\n");

  for (i = 0; i < str\_count; i++) { /\*Вывод начальной матрицы\*/

    for (j = 0; j < stolb\_count; j++) {

      printf("[%Lg] ", \*((arr + i \* stolb\_count) + j));

    }

    printf("\n");

  }

  output(arr, size\_str\_o, size\_stolb\_o, max\_i, max\_j, max\_sum, stolb\_count);

  system("pause");

  return 0;

}

long double input\_arr(long double \*arr, int \*str\_count, int \*stolb\_count) {

  int i = 0, j = 0;

  printf("Введите количество строк и столбцов матрицы от 1 до 15 каждый: ");

  while ((scanf("%d%d", str\_count, stolb\_count) != 2) || \*str\_count < 1 ||

         \*str\_count > 15 || \*str\_count < 1 || \*str\_count > 15) {

    printf("Неверный ввод чисел!\n");

    printf("Введите количество строк и столбцов матрицы от 1 до 15 каждый:");

    fflush(stdin);

    scanf("%\*[^\r\n]");

  }

  for (i = 0; i < \*str\_count;

       i++) { /\*присвоение  значений эллементам матрицы \*/

    for (j = 0; j < \*stolb\_count; j++) {

      scanf("%\*[^\r\n]");

      printf("Введите значение для эллемента arr[%d][%d]: ", i, j);

      while (scanf("%Lf", (arr + i \* \*stolb\_count) + j) != 1) {

        scanf("%\*[^\r\n]");

        printf("Неверный ввод чисел!\n");

        printf("Введите значение для эллемента ar[%d][%d]: ", i, j);

        fflush(stdin);

      }

    }

  }

  return 0;

}

long double input\_size\_oblast(int str\_count, int stolb\_count, int \*size\_str\_o,

                      int \*size\_stolb\_o) {

  printf("Введите размер прямоугольной области: ");

  scanf("%\*[^\r\n]");

  while ((scanf("%d%d", size\_str\_o, size\_stolb\_o) != 2 ||

          \*size\_str\_o > str\_count || \*size\_stolb\_o > stolb\_count ||

          \*size\_str\_o < 0 || \*size\_stolb\_o < 0)) {

    printf("Неверный ввод чисел!\n");

    printf("Введите размер группы в рамках матрицы: ");

    fflush(stdin);

    scanf("%\*[^\r\n]");

  }

  return 0;

}

long double find\_max\_oblast(long double \*arr, int str\_count, int stolb\_count, int size\_str\_o,

                    int size\_stolb\_o, int \*max\_i, int \*max\_j, long double \*max\_sum) {

  long double i = 0, j = 0, sum = 0;

  while (i <= str\_count - size\_str\_o) { /\*Прохождение по каждой обасти\*/

    j = 0;

    while (j <= stolb\_count - size\_stolb\_o) {

      sum = 0;

      calc\_sum(arr, size\_stolb\_o, size\_str\_o, i, j, &sum, stolb\_count);

      if (\*max\_sum < sum) { /\*Сравнение новой суммы и максимальной\*/

        \*max\_sum = sum;

        \*max\_i = i, \*max\_j = j; /\*Сохранение координат\*/

      }

      j++;

    }

    i++;

  }

  return 0;

}

long double output(long double \*arr, int size\_str\_o, int size\_stolb\_o, int max\_i, int max\_j,

           long double max\_sum, int stolb\_count) {

  printf(

      "максимальная сумма = %Lg\nкоординаты начала:(%d;%d)\n", max\_sum,

      max\_i + 1,

      max\_j + 1); /\* Вывод координатов начала области с максимальным значением

                     суммы эллементов со свигом на 1 для НЕпрограммистов\*/

  int i = 0, j = 0;

  for (i = max\_i; i < max\_i + size\_str\_o;

       i++) { /\*Вывод области с максимальным значением суммы эллементов\*/

    for (j = max\_j; j < max\_j + size\_stolb\_o; j++) {

      printf("[%Lg] ", \*(arr + i \* stolb\_count + j));

    }

    printf("\n");

  }

  return 0;

}

long double calc\_sum(long double \*arr, int size\_stolb\_o, int size\_str\_o, int i, int j, long double \*sum,

             int stolb\_count) {

  int k = 0, l = 0;

  for (; l < size\_str\_o; l++) { /\*Вычисление суммы эллементов области\*/

    k = 0;

    for (; k < size\_stolb\_o; k++) {

      \*sum += arr[(i + l) \* stolb\_count + j + k];

    }

  }

  return 0;

}

**Пояснения к программе.**

Для корректного ввода был использован цикл, который не запускает основную программу пока не введены корректные числа, для очистки буфера scanf использовано регулярное выражение и scanf которые считывают все символы до переноса на новую строку и команда fflush(stdin) для очистки буфера ввода stdin.

Для нахождения области с максимальной суммой чисел реализовано 2 цикла с проходом по строкам и столбцам матрицы с условием что размер области влезет с этой точки. Начиная с

той точки проходим по строкам и столбцам области, и складываем сумму цифр в переменную. После прохода по области сравнить сумму с заранее сохранённой, в случае если новая сумма больше, перезаписываем сохранённую сумму и начальную позицию области в матрице.

Для организации дружественного интерфейса пользователя перед вводом данных выводится сообщение для пользователя. Ввод и вывод организован при помощи стандартных функций scanf() и printf(), для этого подключен заголовочный файл stdio.h.

Заголовочный файл stdlib.h необходим для использования функции system(), чтобы приостановить выполнение программы перед завершением (выполняется команда «pause» операционной системы).

**Обоснование эффективности разработанного алгоритма с точки зрения минимального количества итераций и эффективного использования оперативной памяти:** В программе не используются лишние переменные, циклы не проходят по лишним ячейкам памяти 2 цикла используются для прохода по матрице и ещё 2 для прохода по области и подсчёта суммы. Используются условные операторы.