МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ   
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ**

**ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г. ШУХОВА»**

**(БГТУ им. В.Г. Шухова)**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

**Лабораторная работа №5**

по дисциплине: Основы программирования

тема: «Использование подпрограмм при работе с двумерными массивами»

Выполнил: ст. группы ПВ-201

Машуров Дмитрий Русланович

Проверил:

Притчин Иван Сергеевич

Белгород 2020 г.

**Лабораторная работа №5**

**«Использование подпрограмм при работе с двумерными массивами»**

**Цель работы**: получение навыков работы с двумерными массивами и закрепление навыков использования подпрограмм.

**Задания для подготовки к работе:**

1. Изучите способы описания и использования многомерных массивов.
2. Разбейте задачу соответствующего варианта на подзадачи, таким образом, чтобы решение каждой подзадачи описывалось подпрограммой, а основная программа состояла бы в основном из вызовов подпрограмм.
3. Опишите математическое решение задачи с выводом необходимых формул, если необходимо.
4. Опишите блок-схему алгоритма решения задачи в укрупненных блоках.
5. Для каждой подзадачи опишите используемые структуры данных, спецификацию и блок-схему алгоритма
6. Опишите блок-схему алгоритма решения задачи с использованием блоков «предопределенный процесс».
7. Закодируйте алгоритм.
8. Подберите наборы тестовых данных с обоснованием их выбора.

**Задания к работе:**

1. Наберите программу, отладьте ее и протестируйте.
2. Выполните анализ ошибок, выявленных при отладке программы

**Задание варианта №17**

Дана квадратная матрица, все элементы которой различны. Назовем псевдодиагональю множество элементов этой матрицы, лежащих на прямой, параллельной прямой, содержащей элементы , где – порядок матрицы. Найти сумму максимальных элементов псевдодиагоналей данной матрицы

**Выполнение работы:**

1. **Общее решение**

По условию задачи нам нужно перебрать элементы, которые стоят на диагоналях, параллельных диагонали (которую в дальнейшем обозначим *побочной диагональю*), содержащей элементы , (выделены красным), и найти среди каждой диагонали максимальное значение, которое потом мы сложим с остальными максимальными значениями:

1. **Разбиение задачи на подзадачи**
2. Ввод матрицы определённого порядка
3. Поиск и суммирование максимального элемента каждой псевдодиагонали, располагающейся выше обратной диагонали (и включая саму обратную диагональ)
4. Поиск и суммирование максимального элемента каждой псевдодиагонали, располагающейся ниже обратной диагонали (не включая обратную диагональ)
5. Вывод суммы
6. **Описание структур данных**

M – константа, определяющая максимальный размер матрицы

N – константа, определяющая максимальный размер столбцов матрицы

t\_matrix – тип, описывающий квадратную матрицу размера 99х99

const

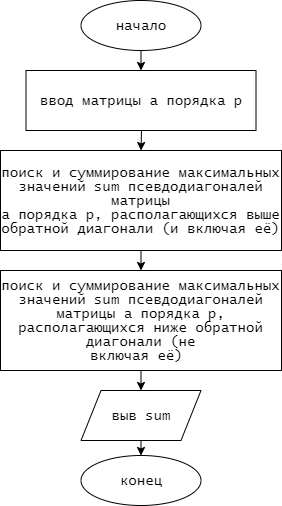
M = 99;

N = 99;

type

t\_matrix = array[0..m,0..n] of integer;

1. **Блок-схема с укрупнёнными блоками**

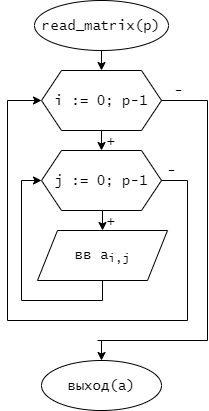
****

1. **Описание подпрограмм**

Спецификация процедуры read\_matrix

1. Заголовок: procedure read\_matrix(var a: t\_matrix; p: integer)
2. Назначение: ввод матрицы a порядка p
3. Входные параметры: p
4. Выходные параметры: a

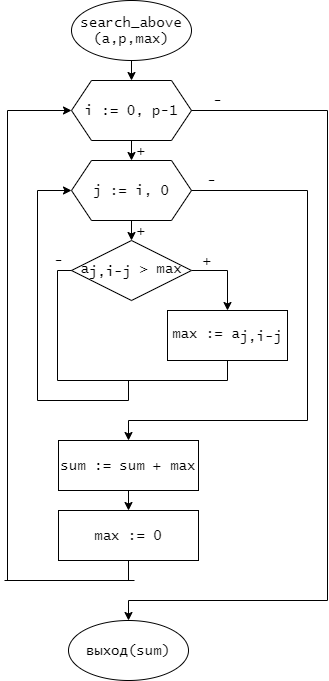
Блок-схема:



Спецификация процедуры search\_above

1. Заголовок: procedure search\_above(a: t\_matrix; p: integer; max: integer; var sum: integer;)
2. Назначение: поиск и суммирование максимальных значений в sum псевдодиагоналей матрицы a порядка p, располагающихся выше обратной диагонали (и включая её)
3. Входные параметры: a, p, max
4. Выходные параметры: sum

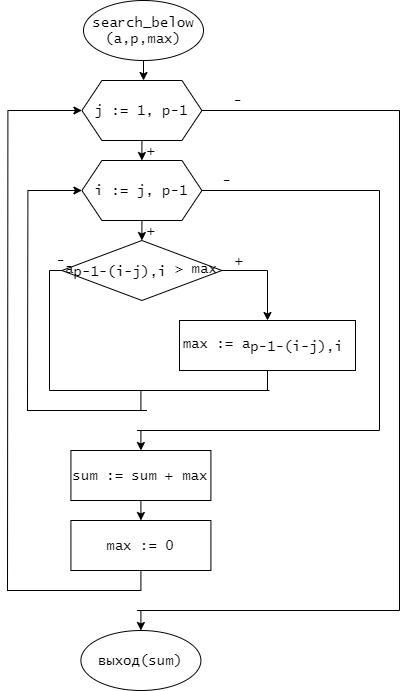
Блок-схема:



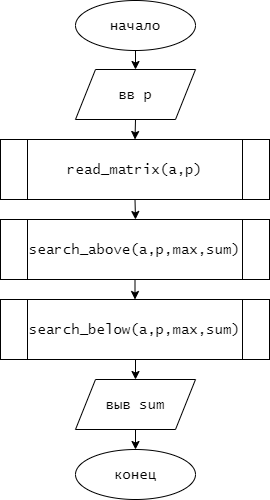
Спецификация процедуры search\_below

1. Заголовок: procedure search\_below(a: t\_matrix; p: integer; max: integer; var sum: integer)
2. Назначение: поиск и суммирование максимальных значений в sum псевдодиагоналей матрицы a порядка p, располагающихся ниже обратной диагонали (не включая её)
3. Входные параметры: a, p, max
4. Выходные: sum

Блок-схема:



1. **Блок-схема с блоками «предопределённый процесс»**

****

1. **Тестовые данные:**

|  |  |  |
| --- | --- | --- |
| № п/п | Входные значения | Результат |
| 1 |  | 8 |
| 2 |  | 24 |
| 3 |  | 5 |

1. **Текст программы**

**const**

M = 99;

N = 99;

**type**

t\_matrix = **array**[0..m,0..n] **of** integer;

t\_iterator = integer;

{процедура ввода матрицы matrix размера m строк и n столбцов}

**procedure** read\_matrix(**var** a: t\_matrix; p: integer);

**var** i,j: integer;

**begin**

**for** i := 0 **to** p-1 **do**

**for** j := 0 **to** p-1 **do**

read(a[i][j]);

**end**;

{процедура поиска выше обратной диагонали}

**procedure** search\_above(a: t\_matrix; p: integer; max: integer; **var** sum: integer);

**var** i,j: integer;

**begin**

**for** i := 0 **to** p-1 **do**

**begin**

**for** j := i **downto** 0 **do**

**if** (a[j][i-j] > max) **then**

max := a[j][i-j];

sum := sum + max;

max := 0;

**end**;

**end**;

{процедура поиска ниже обратной диагонали}

**procedure** search\_below(a: t\_matrix; p:integer; max: integer; **var** sum: integer);

**var** i,j: integer;

**begin**

**begin**

**for** j := 1 **to** p-1 **do**

**begin**

**for** i := j **to** p-1 **do**

**if** (a[p - 1 - (i - j)][i] > max) **then**

max := a[p - 1 - (i - j)][i];

sum := sum + max;

max := 0;

**end**

**end**;

**end**;

**var**

a: t\_matrix;

p, max, sum: integer;

**begin**

max := 0;

sum := 0;

writeln('Ввод квадратной матрицы');

write('Ввод порядка матрицы: ');

read(p);

writeln('Вводите значения матрицы по строками');

read\_matrix(a,p);

search\_above(a,p,max,sum);

search\_below(a,p,max,sum);

write('Сумма максимальных значений псевдодиагоналей равна = ');

write(sum);

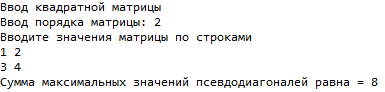
**end**.

1. **Анализ допущенных ошибок**

* вывод max вместо sum в конце
* добавлены лишние буквы в процедуре read\_matrix

1. **Результаты работы программы**

*Пример №1:*



*Пример №2:*

