

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:
Брусенцева Валентина
Станиславовна

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 $a_{i,n-i+1}$, где n – 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 *обратной диагональю*), содержащей элементы $a_{i,n-i+1}$, (выделены красным), и найти среди каждой диагонали максимальное значение, которое потом мы сложим с остальными максимальными значениями:

3	4	2
54	2	1
0	7	54

3	4	2
54	2	1
0	7	54

3	4	2
54	2	1
0	7	54

3	4	2
54	2	1
0	7	54

3	4	2
54	2	1
0	7	54

2. Разбиение задачи на подзадачи

- 1) Ввод матрицы определённого порядка
- 2) Поиск и суммирование максимального элемента каждой псевдодиагонали, располагающейся выше обратной диагонали (и включая саму обратную диагональ)
- 3) Поиск и суммирование максимального элемента каждой псевдодиагонали, располагающейся ниже обратной диагонали (не включая обратную диагональ)
- 4) Вывод суммы

3. Описание структур данных

M – константа, определяющая максимальный размер матрицы

N – константа, определяющая максимальный размер столбцов матрицы

$t_iterator$ – тип, определяющий различные переменные-итераторы в циклах

t_matrix – тип, описывающий квадратную матрицу размера 99×99

const

$M = 99;$

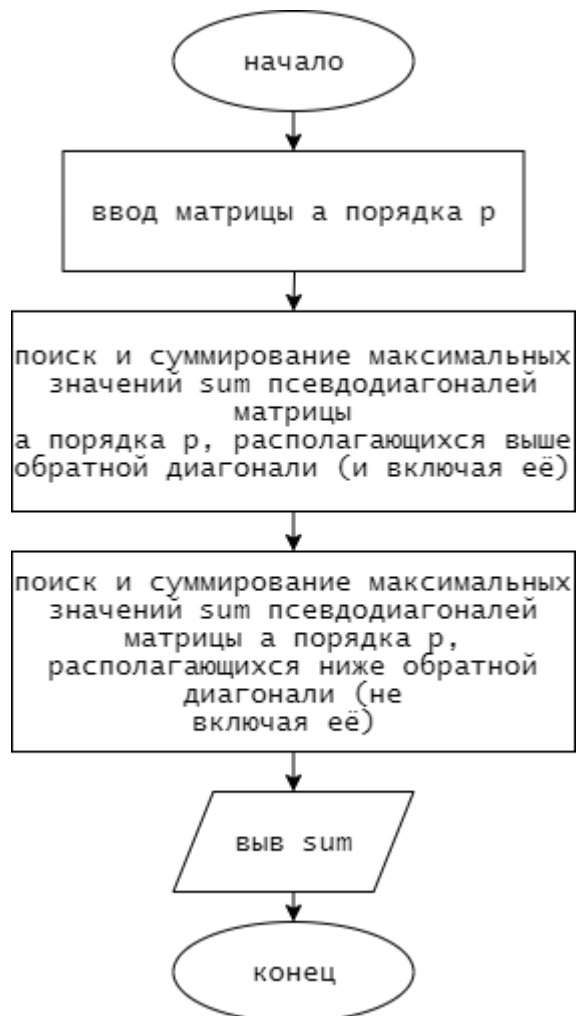
$N = 99;$

type

$t_matrix = \text{array}[0..m, 0..n] \text{ of integer};$

$t_iterator = \text{integer};$

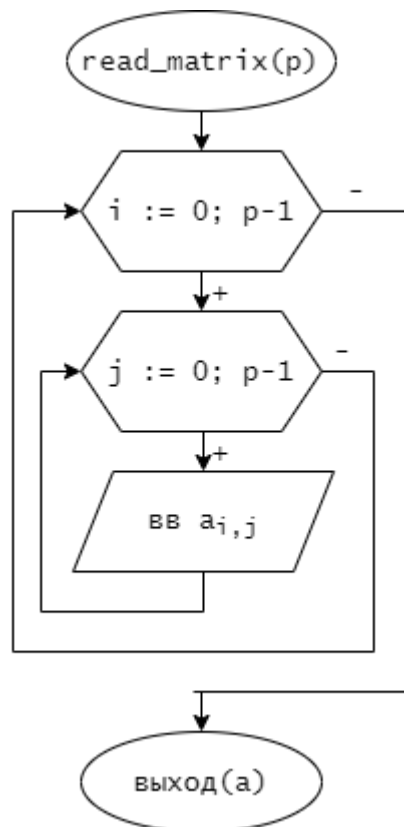
4. Блок-схема с укрупнёнными блоками



5. Описание подпрограмм

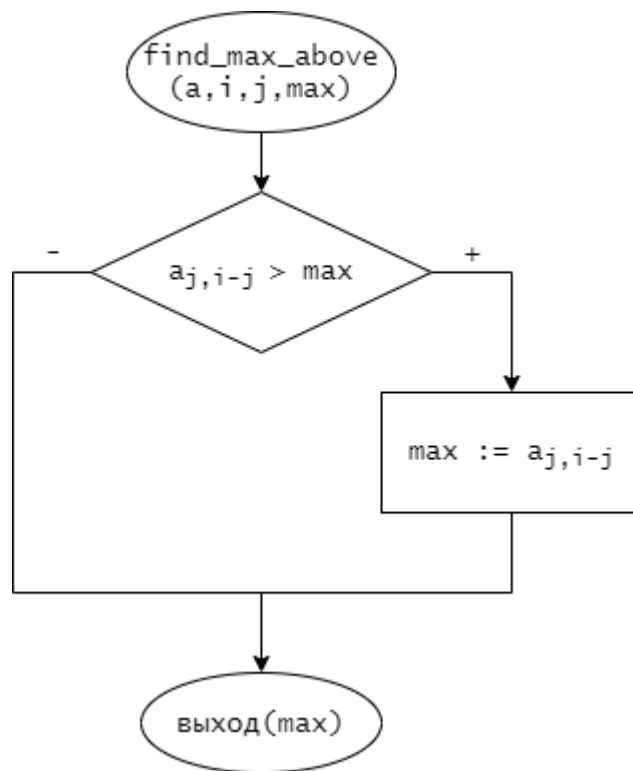
Спецификация процедуры `read_matrix`

- 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 find_max_above

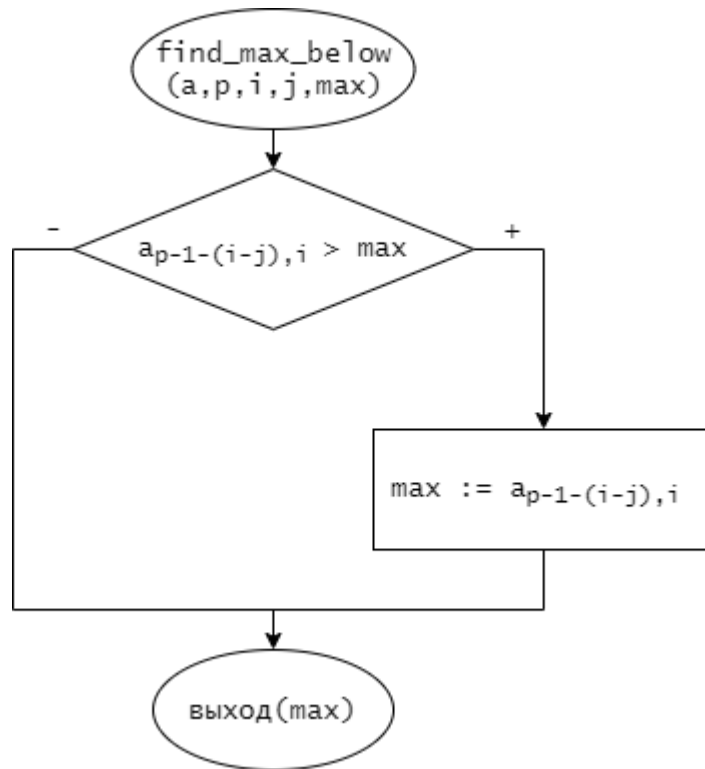
- 1) Заголовок: `procedure find_max_above(a: t_matrix; i: integer; j: integer; var max: integer)`
 - 2) Назначение: сравнение максимального значения `max` и значения `aj,i-j` с целью найти максимальное из них
 - 3) Входные параметры: `a`, `i`, `j`, `max`
 - 4) Выходные параметры: `max`
- Блок-схема:



Спецификация процедуры find_max_below

- 1) Заголовок: `procedure find_max_below(a: t_matrix; p: integer; i: integer; j: integer; var max: integer)`
- 2) Назначение: сравнение максимального значения `max` и значения $a_{p-1-(i-j)}$ с целью найти максимальное из них
- 3) Входные параметры: `a`, `p`, `i`, `j`, `max`
- 4) Выходные параметры: `max`

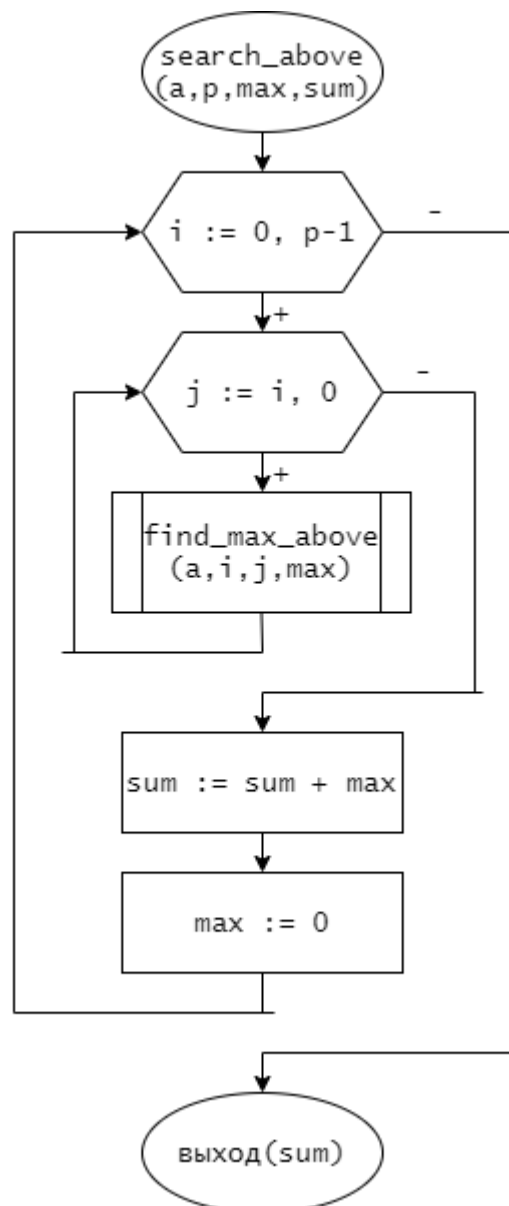
Блок-схема:



Спецификация процедуры search_above

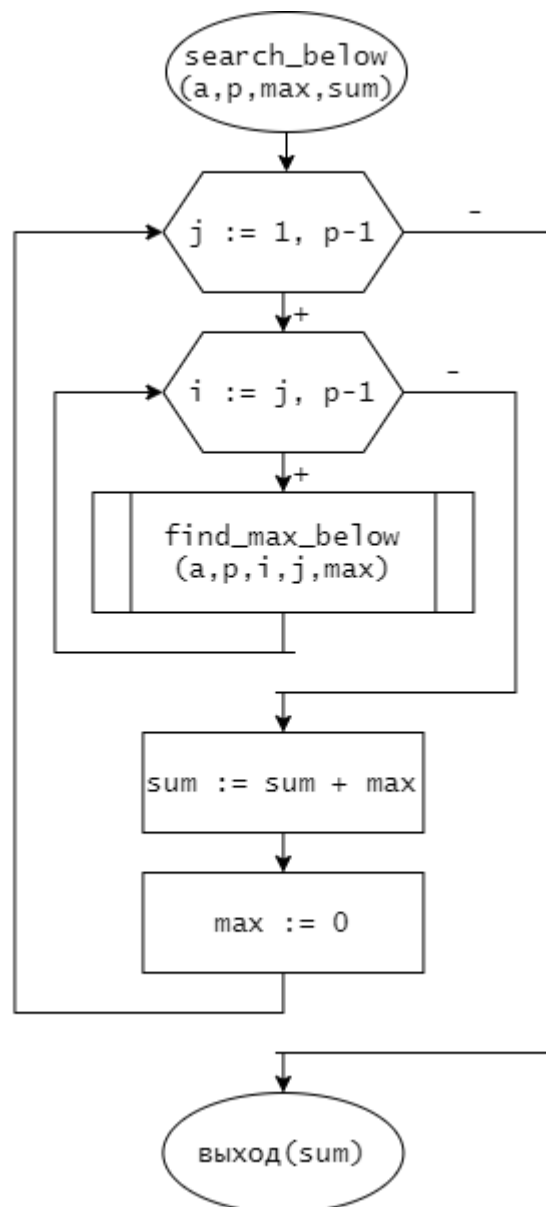
- 1) Заголовок: `procedure search_above(a: t_matrix; p: integer; max: integer; var sum: integer;)`
- 2) Назначение: поиск и суммирование максимальных значений в `sum` псевдодиагоналей матрицы `a` порядка `p`, располагающихся выше обратной диагонали (и включая её)
- 3) Входные параметры: `a`, `p`, `max`, `sum`
- 4) Выходные параметры: `sum`

Блок-схема:

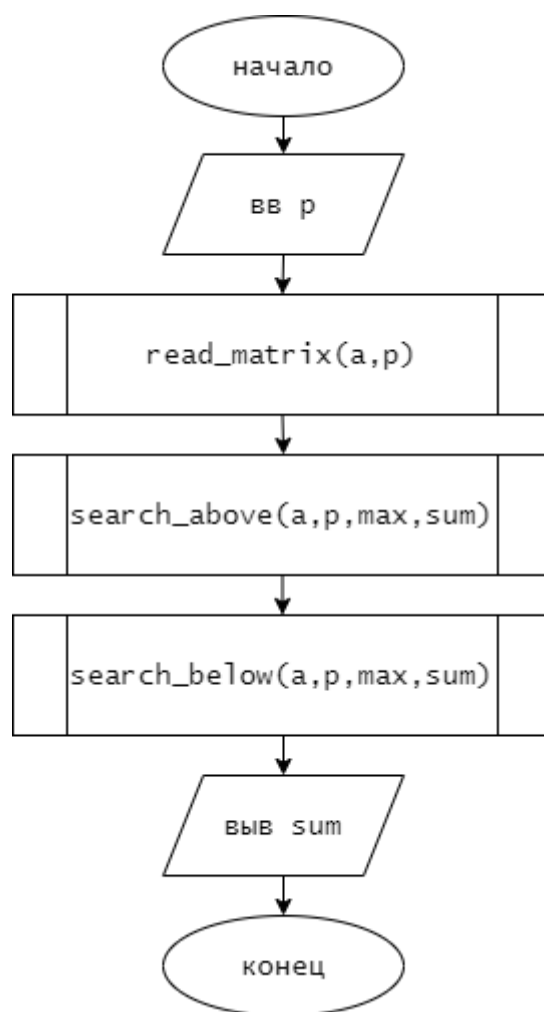


Спецификация процедуры search_below

- 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`, `sum`
 - 4) Выходные: `sum`
- Блок-схема:



6. Блок-схема с блоками «предопределённый процесс»



7. Тестовые данные:

№ п/п	Входные значения	Результат
1	1 2 3 4	8
2	5 2 1 4 3 2 7 5 3	24
3	0 0 5 0 5 0 5 0 0	5

8. 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: t_iterator;
begin
  for i := 0 to p-1 do
    for j := 0 to p-1 do
      read(a[i][j]);
    end;
  end;

{поиск максимального выше обратной диагонали}
procedure find_max_above(a: t_matrix; i: integer; j: integer; var max: integer);
begin
  if (a[j][i-j] > max) then
    max := a[j][i-j]
  end;
end;

{поиск максимального ниже обратной диагонали}
procedure find_max_below(a: t_matrix; p: integer; i: integer; j: integer; var
max: integer);
begin
  if (a[p - 1 - (i - j)][i] > max) then
    max := a[p - 1 - (i - j)][i];
  end;
end;

{процедура поиска выше обратной диагонали}
procedure search_above(a: t_matrix; p: integer; max: integer; var sum: integer);
var i,j: t_iterator;
begin
  for i := 0 to p-1 do
    begin
      for j := i downto 0 do
        begin
          find_max_above(a,i,j,max);
        end;
      end;
    end;
  end;
```

```

        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: t_iterator;
begin
    begin
        for j := 1 to p-1 do
            begin
                for i := j to p-1 do
                    begin
                        find_max_below(a, p, i, j, max);
                    end;
                    sum := sum + max;
                    max := 0;
                end
            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.

```

9. Анализ допущенных ошибок

- вывод `max` вместо `sum` в конце
- добавлены лишние буквы в процедуре `read_matrix`

10. Результаты работы программы

Пример №1:

```
Ввод квадратной матрицы
Ввод порядка матрицы: 2
Вводите значения матрицы по строками
1 2
3 4
Сумма максимальных значений псевдодиагоналей равна = 8
```

Пример №2:

```
Ввод квадратной матрицы
Ввод порядка матрицы: 3
Вводите значения матрицы по строками
5 2 1
4 3 2
7 5 3
Сумма максимальных значений псевдодиагоналей равна = 24
```