

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 №4

по дисциплине: Основы программирования

тема: «Обработка одномерных массивов с использованием
подпрограмм»

Выполнил: ст. группы ПВ-202
Буйвало Анастасия Андреевна

Проверил:
Притчин Иван Сергеевич

Белгород 2020 г.

Лабораторная работа №4¹ **«Обработка одномерных массивов с использованием подпрограмм»**

Цель работы: получение навыков работы с массивами и подпрограммами.

Задания для подготовки к работе²:

1. Изучите способы описания и использования массивов, базовые алгоритмы обработки массивов.
2. Изучите виды и назначение подпрограмм, правила их описания и вызова.
3. Разбейте задачу соответствующего варианта на подзадачи, таким образом, чтобы решение каждой подзадачи описывалось подпрограммой, а основная программа состояла бы в основном из вызовов подпрограмм.
4. Опишите блок-схему алгоритма решения задачи в укрупненных блоках.
5. Опишите используемые структуры данных, спецификации и блок-схемы подпрограмм, соответствующих укрупненным блокам. Спецификация содержит: заголовок подпрограммы, назначение, входные и выходные параметры.
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 №7:

Даны две последовательности. Получить упорядоченную по не возрастанию последовательность, состоящую из тех членов первой последовательности, которых нет во второй³.

¹ Тема лабораторной работы по центру страницы, **полужирным** шрифтом.

² Внимательно **ознакомьтесь с заданиями для подготовки. Убедитесь**, что все они выполнены.

³ **Используйте выравнивание по ширине**, если это улучшает визуальную составляющую вашей работы.

Выполнение работы⁴:

1. Выделение подзадач⁵

Выделим следующие подзадачи:

- a. Ввод массивов a размера n и b размера m ;
- b. Получение последовательности на массиве c ⁶ из элементов массива a , значения которых отсутствуют в массиве b ;
 $k :=$ количество элементов в массиве c ;
- c. Сортировка k элементов массива c по не возрастанию;
- d. Вывод элементов массива c .

Опишем алгоритм в укрупненных блоках в терминах выделенных подзадач⁷.

⁴ **Перенесите выполнение работы на новую страницу.** Отделите решаемые задачи и выполнение.

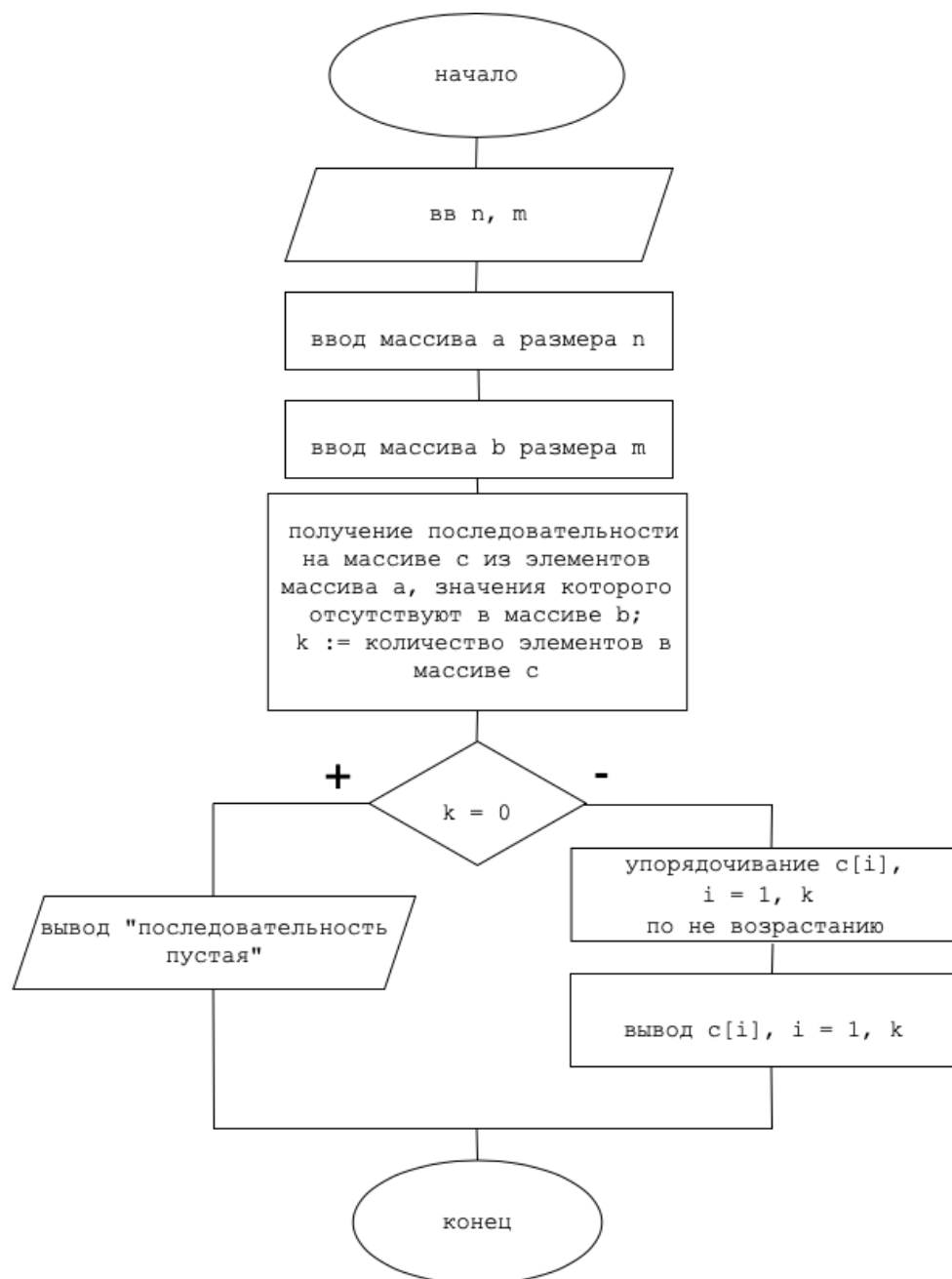
⁵ Выделение подзадач должно происходить **до** проектирования алгоритма и написания кода. В данном разделе подумайте, какие действия необходимо выполнить, чтобы решить задачу. Пока что не обязательно думать о последовательности выполнения этих действий, и как между собой они будут связаны. Цель – просто выделить подзадачи.

Описать подзадачи можно и более подробно (в том числе и без использования нумерованных списков), если в этом есть необходимость.

⁶ Формулировка «получение последовательности на массиве» использована не случайно. Очень вероятно, что не все элементы массива c будут использованы. Именно этим и обусловлен данный оборот.

⁷ Перенос блок-схемы в укрупненных блоках на следующую страницу выполнен с целью **не разрывать блок-схему между страницами**. Если блок-схема умещается полностью – оставьте её на этой странице.

2. Блок-схема алгоритма в укрупненных блоках⁸:



3. Описание структур данных⁹:

```
const SIZE = 100;  
type t_arr = array [1..SIZE] of integer;
```

⁸ Для блок-схем используйте моноширинный шрифт.

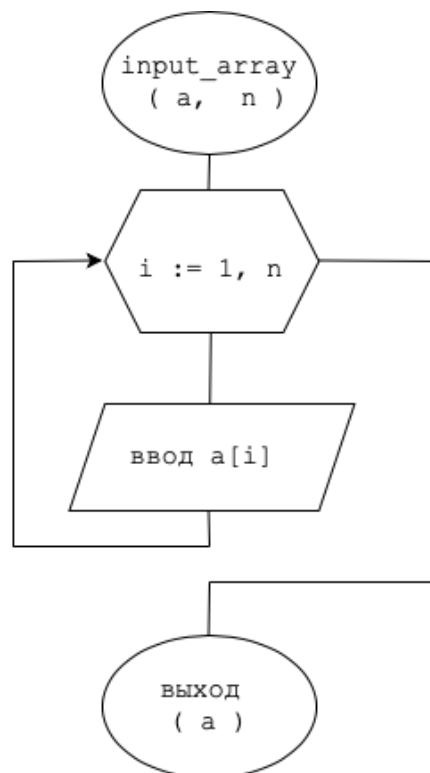
⁹ Описываются константы и все создаваемые типы данных.

4. Описание подпрограмм:

Процедура `input_array`¹⁰:

1. Заголовок: `procedure input_array(var a: t_arr; n11: integer;)`.
2. Назначение: ввод массива `a` размера `n`¹².
3. Входные параметры: `n`.
4. Выходные параметры: `a`.

Блок-схема:



Процедура `get_array`:

1. Заголовок: `procedure get_array(a: t_arr; n: integer; b: t_arr; m: integer; var c: t_arr; var k: integer)`¹³;
2. Назначение: получение последовательности в массиве `c` размера `k` из элементов массива `a` размера `n`, которых нет в массиве `b` размера `n`.
3. Входные параметры: `a`, `n`, `b`, `m`.
4. Выходные параметры: `c`, `k`.

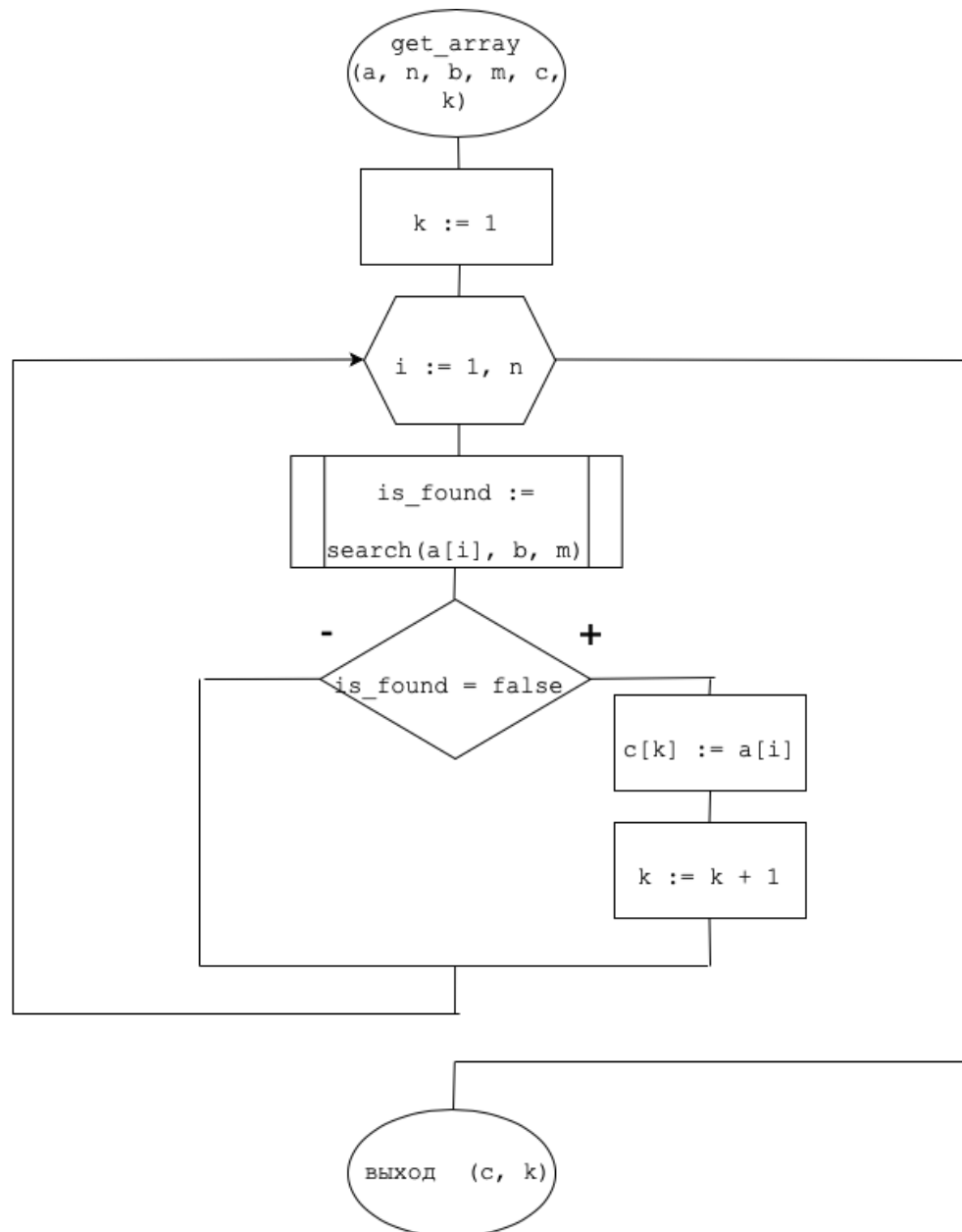
¹⁰ Настоятельно рекомендуется выделять ключевые слова языков программирования, имена идентификаторов и т. п. моноширинным шрифтом (например, Consolas).

¹¹ Вы действительно можете указать вместо `integer` какой-нибудь другой целочисленный тип, который поддерживает только положительные значения. Ведь количество элементов в массиве не может быть отрицательным.

¹² Все параметры функции **обязаны** прозвучать в назначении.

¹³ **Обратите внимание на группировку параметров.** В начале собраны параметры, характеризующие первый массив, потом – параметры характеризующие второй массив и т. д.

Блок схема:

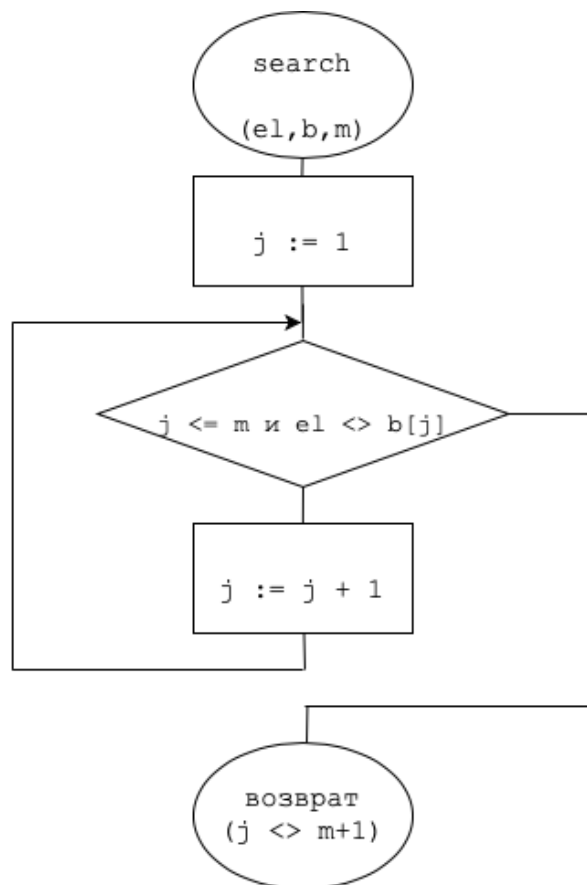


Функция search:

1. Заголовок: `search (el: integer, b: t_arr; m: integer)`
2. Назначение: возвращает значение истина, если `el` является элементом массива `b` размера `m`, иначе – ложь¹⁴.
3. Входные параметры: `el`, `b`, `m`.
4. Выходные параметры: `fl`.

¹⁴ Назначение функции в большинстве случаев должно начинаться со слова «возвращает».

Блок-схема¹⁵:



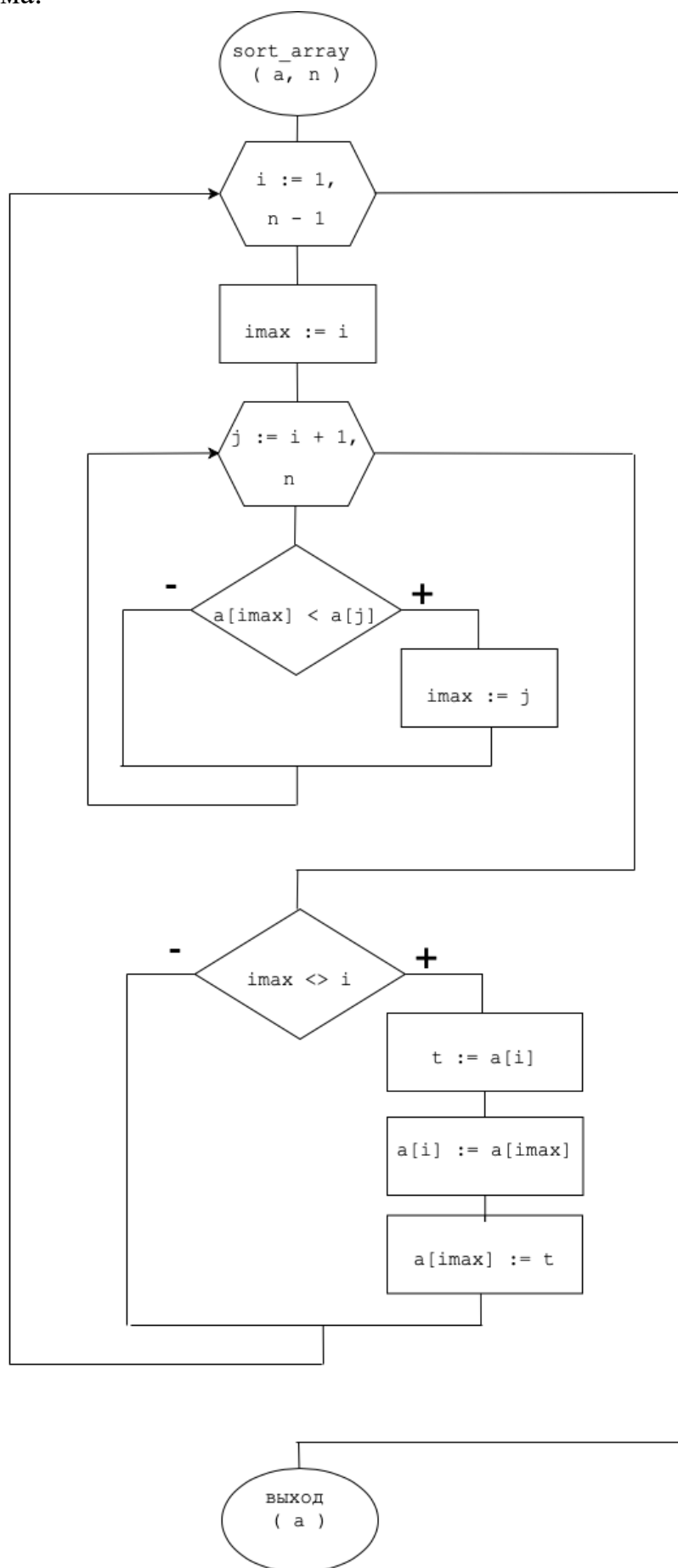
Процедура sort_array:

1. Заголовок: procedure sort_array (var a: t_arr; n: integer;).
2. Назначение: сортировка массива a размера n по не возрастанию¹⁶.
3. Входные параметры: n.
4. Выходные параметры: a.

¹⁵ При оформлении блок-схемы избегайте использования обозначений, присущих языкам программирования. В данном случае, неплохо заменить \neq на \neq .

¹⁶ **Описания** назначений процедур и функций должны быть конкретными. В них не должно быть «неопределенности». В данном случае указано каким образом производится сортировка.

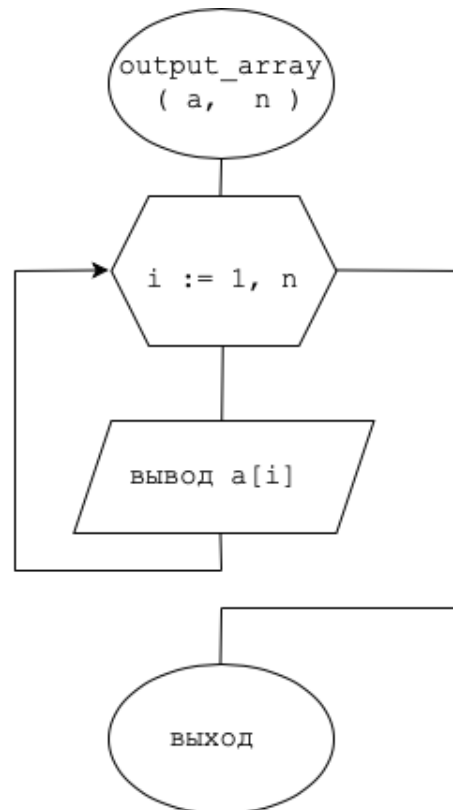
Блок-схема:



Процедура output_array:

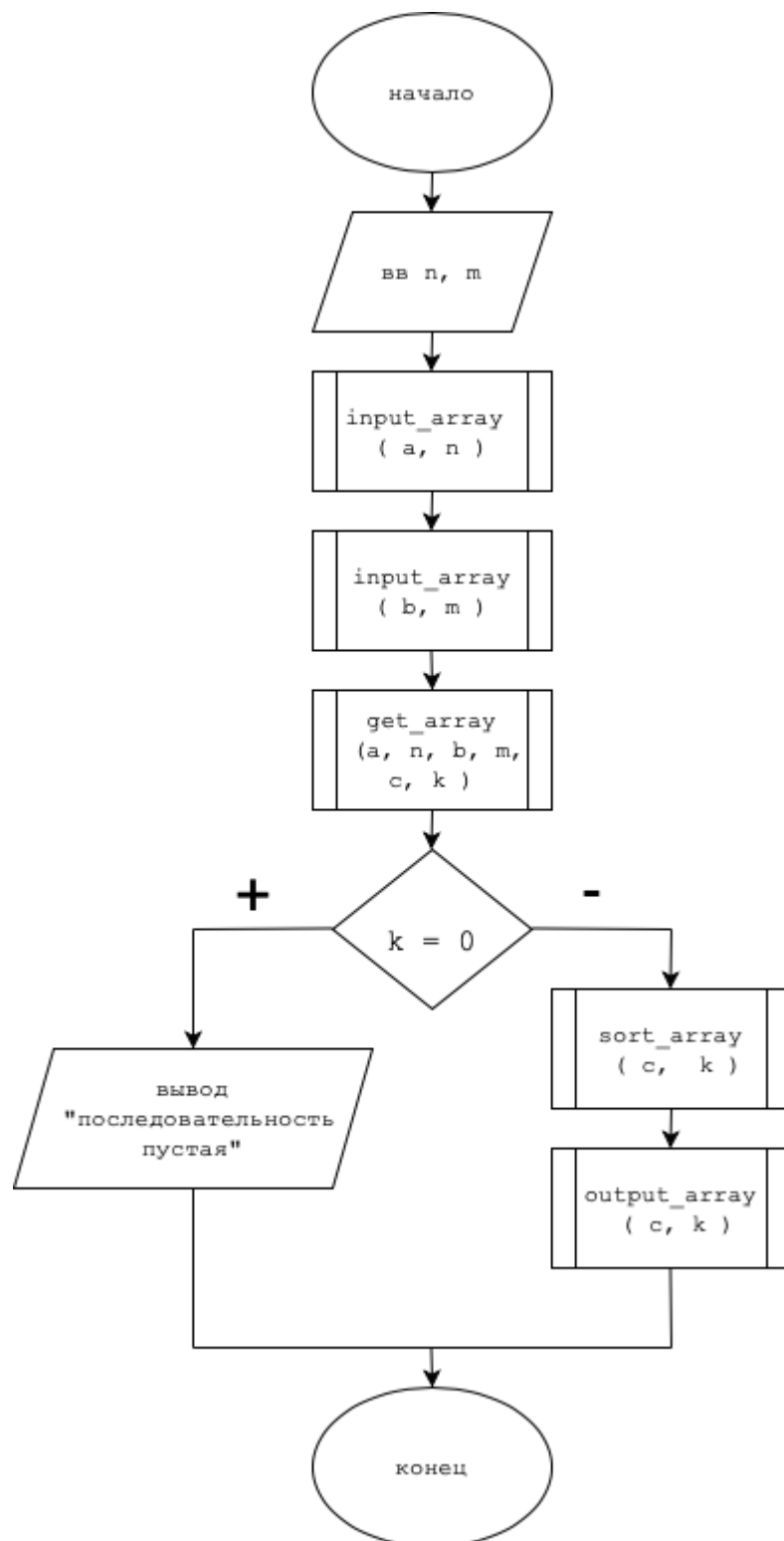
1. Заголовок: procedure output_array (var a: t_arr; n: integer).
2. Назначение: вывод массива a^{17} размера n.
3. Входные параметры: n.
4. Выходные параметры: нет.

Блок-схема:



¹⁷ В данной работе назначение изначально звучало следующим образом: «вывод упорядоченного массива...». Но данная функция же выводит произвольный массив. Да, по условию задачи он будет отсортирован. Но **в назначении указывать нужно не то, что конкретно в нашей программе делает функция, а что делает функция в принципе.**

5. Блок-схема алгоритма решения задачи с блоками «предопределенный процесс»:



6. Тестовые данные¹⁸:

№	Исходные данные				Результаты
	<i>n</i>	<i>m</i>	<i>a</i>	<i>b</i>	<i>c[i]</i>
1	1	1	1	2	1
2	1	1	2	2	Последовательность пустая
3	2	3	1 2	2 3 4	1
4	3	2	1 2 3	3 2	1
5	1	3	1 2 3	3 4 5	2 1
6	3	3	1 2 2	4 5 6	2 2 1
7	3	4	1 2 3	1 2 3 4	Последовательность пустая

7. Текст программы¹⁹:

```
program lab4;
const SIZE20 = 100;
type t_arr = array [1..SIZE] of integer;
var a, b, c: t_arr;
    i, j, n, k, m, t: integer;
    fl : boolean;

{ввод массива a размера n}
procedure input_array(var a: t_arr; n: integer);
var i:integer;
begin
    for i := 1 to n do
        read(a[i])
    end;

{возвращает значение истина, если el входит в массив b размера m, иначе ложь}
function search (el : integer; b : t_arr; m: integer): boolean;
var j : integer;
    fl : boolean;
begin
    j := 1;
    while (j <= m) and (el <> b[j]) do
        j := j + 1;
    if j = m + 1 then
        fl := false
    else
        fl := true;
    search := fl;
end;

{получение последовательности в массиве c размера k из элементов массива a размера n,
которых нет в массиве b размера n}
procedure get_array(a : t_arr; n : integer; b : t_arr; m: integer; var c: t_arr; var k:
integer);
var i, j : integer;
    is_found : boolean;
begin
    k:=0;
    for i := 1 to n do
```

¹⁸ Тестовые данные определяются до написания программы. Тестовые случаи должны **репрезентативно** отображать классы возможных входных данных. Рекомендуется перечислять тестовые случаи в порядке их усложнения.

Кегль шрифта для таблиц должен быть меньше, чем для основного текста. Возьмите 12 или меньше.

¹⁹ **Стиль оформления кода! Если вашу программу сложно читать, считайте, что она не работает.**

Кегль шрифта для кода должен быть меньше, чем для основного текста. Возьмите 10 или меньше.

²⁰ Имена констант должны быть написаны в верхнем регистре.

```

begin
is_found := search(a[i], b, m);
if is_found = false then
begin
k := k + 1;
c[k] := a[i];
end;
end;
end;

{сортировка массива a размера n по не возрастанию}
procedure sort_array(var a: t_arr; n: integer);
var i, j, t, imax: integer;
begin
for i := 1 to n-1 do
begin
imax := i;
for j := i+1 to n do
if a[imax] < a[j] then
imax := j;
if imax <> i then
begin
t := a[i];
a[i] := a[imax];
a[imax] := t;
end;
end;
end;

{вывод массива a размера n}
procedure output_array(var a: t_arr; n: integer);
var i: integer;
begin
for i := 1 to n do
write (a[i], ' ');
end;

begin
writeln ('введите размер первого массива');
read (n);
writeln ('введите размер второго массива');
read (m);
input_array(a, n);
input_array(b, m);
get_array(a, n, b, m, c, k);
if k = 0 then
writeln ('последовательность пустая')
else
begin
sort_array (c, k);
output_array (c, k);
end;
end.

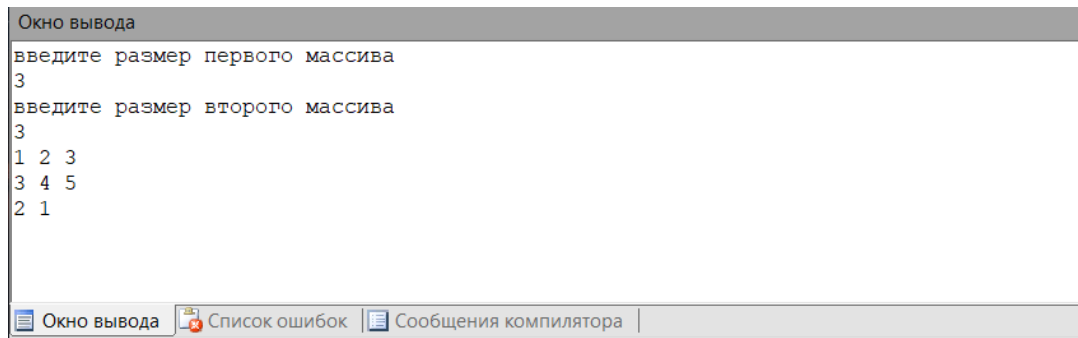
```

8. Анализ допущенных ошибок²¹:

- Пропущено k в sort (c, k);

²¹ Указываются ошибки при наборе кода и проектирования алгоритма. Это позволяет улучшить качество набора программ в будущем. Было много ошибок – напишите много. Не стесняйтесь.

9. Результаты работы программы:



The screenshot shows a window titled "Окно вывода" (Output Window) with a light gray background. The text inside the window is as follows:

```
введите размер первого массива
3
введите размер второго массива
3
1 2 3
3 4 5
2 1
```

At the bottom of the window, there is a tab bar with three tabs: "Окно вывода" (selected), "Список ошибок" (List of errors), and "Сообщения компилятора" (Compiler messages).