

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 №7**

по дисциплине: Основы программирования

тема: «Использование рекурсивных процедур и функций»

Выполнил: ст. группы ПВ-201  
Машуров Дмитрий Русланович

Проверил:  
Притчин Иван Сергеевич

Белгород 2020 г.

## **Лабораторная работа №7**

### **Использование рекурсивных процедур и функций**

**Цель работы:** получение навыков описания и использования рекурсивных подпрограмм

#### **Задания для подготовки к работе**

1. Изучите правила организации рекурсивных процедур и функций.
2. Опишите математическое решение задачи, если необходимо.
3. Опишите блок-схему алгоритма решения задачи в укрупненных блоках
4. Опишите используемые структуры данных, если необходимо.
5. Опишите спецификацию и блок-схему итеративной подпрограммы.
6. Опишите спецификацию и блок-схему рекурсивной подпрограммы,
7. Если список параметров рекурсивной подпрограммы отличается от списка параметров итеративной подпрограммы, то опишите подпрограмму с таким же заголовком, как и у итеративной подпрограммы, которая вызывает рекурсивную с необходимыми ей параметрами. При этом сама рекурсивная подпрограмма может быть описана в основной подпрограмме или вне ее.
8. Закодируйте алгоритмы
9. Подберите наборы тестовых данных с обоснованием их выбора.

#### **Задания к работе**

1. Наберите программы, отладьте их, протестируйте.
2. Выполните анализ ошибок, выявленных при отладке программы

#### **Задание варианта №17**

Найти номер первого вхождения максимального значения в последовательности длины  $n$ .

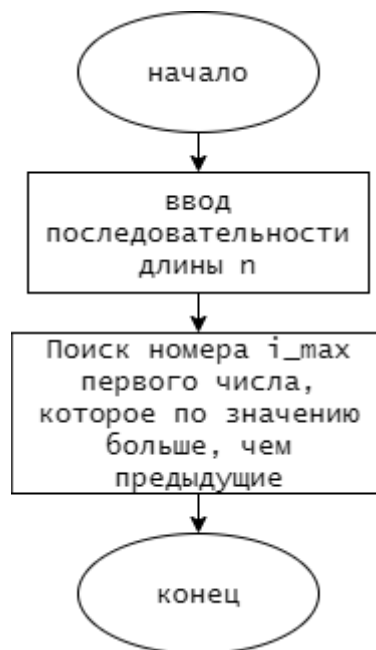
## Выполнение работы:

### 1. Выделение подзадач

Выделим следующие подзадачи:

- 1) Ввод последовательности длины  $n$
- 2) Поиск номера  $i_{\max}$  первого числа, которое по значению больше предыдущих

### 2. Блок-схема алгоритма с укрупнёнными блоками



### 3. Описание структур данных

`SIZE` — константа, описывающая максимальный размер последовательности

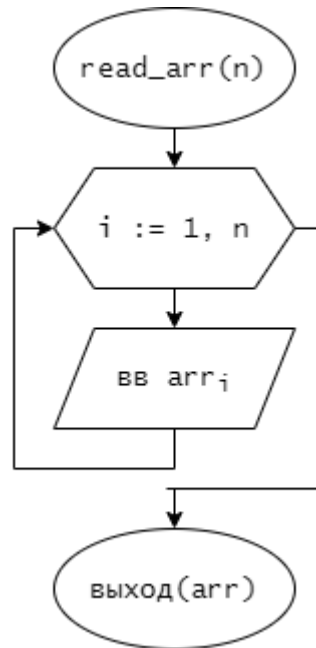
`t_arr` — тип, описывающий последовательность размера `SIZE`

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

#### 4. Описание подпрограмм

Спецификация процедуры read\_arr

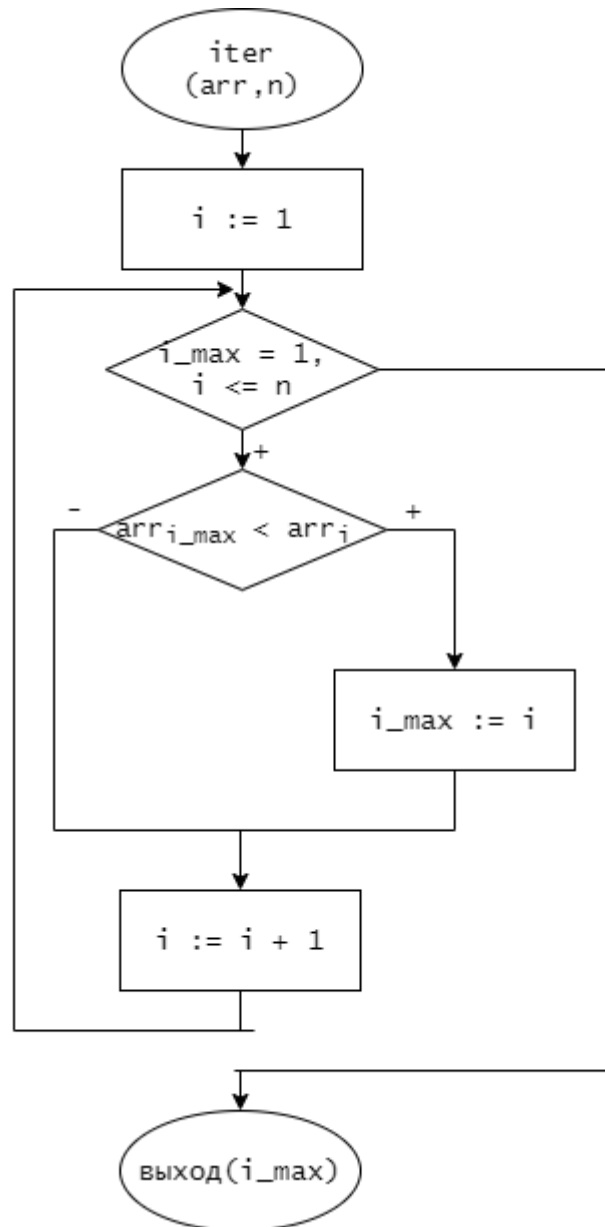
- 1) Заголовок: procedure read\_arr(var arr: t\_arr; n: integer)
  - 2) Назначение: ввод последовательности arr размера n
  - 3) Входные параметры: n
  - 4) Выходные параметры: arr
- Блок-схема:



### Спецификация процедуры `iter`

- 1) Заголовок: `procedure iter(arr: t_arr; n: integer;)`
- 2) Назначение: поиск номер `i_max` первого вхождения максимального значения последовательности `arr` размера `n`
- 3) Входные параметры: `arr`, `n`
- 4) Выходные параметры: `i_max`

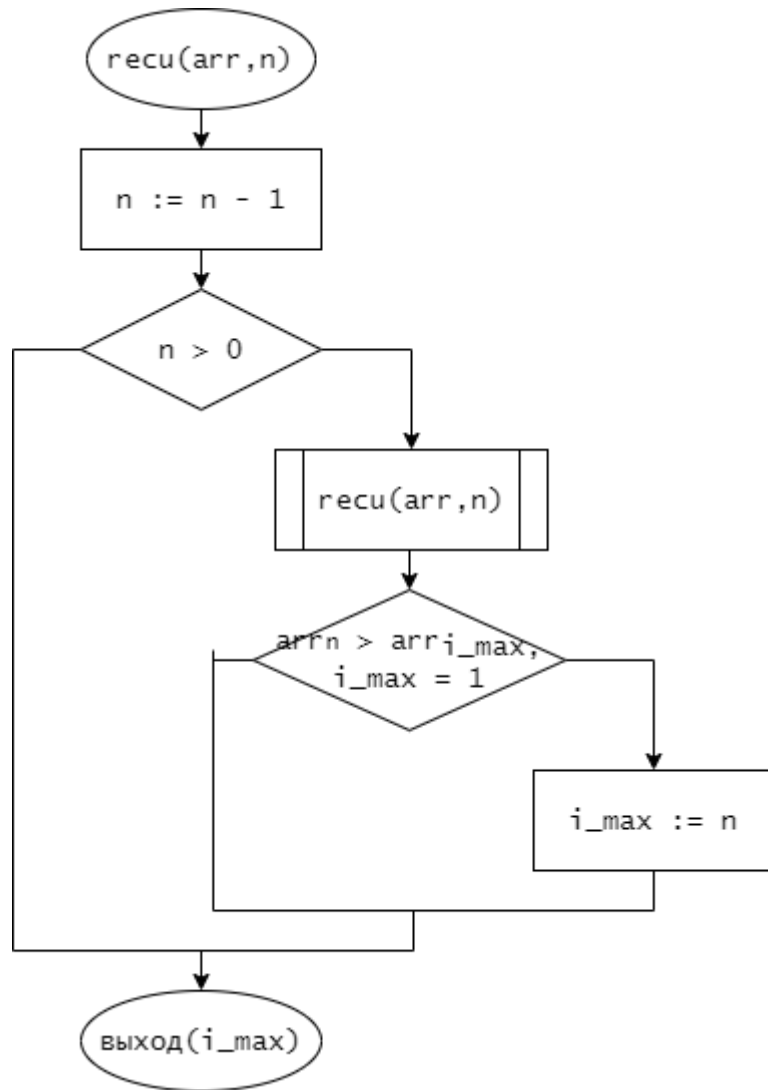
Блок-схема



### Спецификация процедуры `recu`

- 1) Заголовок: `procedure recu(arr: t_arr; n: integer)`
- 2) Назначение: поиск номера `i_max` первого вхождения максимального значения последовательности `arr` размера `n`
- 3) Входные параметры: `arr`, `n`
- 4) Выходные параметры: `i_max`

Блок-схема:



## 5. Тестовые данные

№	Последовательность	Результат
1	11 63 80 32 69 21 49	2
2	10 9 8 7 6 5 1337	7
3	21 17 10 5	1

## 6. Тексты программ

*Текст итеративной программы:*

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

procedure read_arr(var arr: t_arr; n: integer);
var i: integer;
begin
  for i := 1 to n do
    read(arr[i]);
end;

procedure iter(arr: t_arr; n: integer; var i_max: integer);
var i: integer;
begin
  i := 1;
  while (i_max = 1) and (i <= n) do
    begin
      if (arr[i_max] < arr[i]) then
        i_max := i;
      i := i + 1;
    end;
end;

var
  n: integer;
  arr: t_arr;
  i_max, res: integer;

begin
  write('Введите размер последовательности, не превышающий 100: ');
  read(n);

  writeln('Вводите последовательность');
  read_arr(arr, n);

  i_max := 1;

  iter(arr, n, i_max);
  writeln('Результат: ', i_max);
end.
```

*Текст рекурсивной программы:*

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

procedure read_arr(var arr: t_arr; n: integer);
var i: integer;
begin
    for i := 1 to n do
        read(arr[i]);
    end;

procedure recu(arr: t_arr; n: integer; var i_max: integer);
begin
    n := n - 1;
    if (n > 0) then
        begin
            recu(arr,n,i_max);
            if (arr[n] > arr[i_max]) and (i_max = 1) then
                i_max := n;
            end;
        end;
end;

var
    n: integer;
    arr: t_arr;
    i_max,res: integer;

begin
    write('Введите размер последовательности, не превышающий 100: ');
    read(n);

    writeln('Вводите последовательность');
    read_arr(arr,n);

    n := n + 1;
    i_max := 1;

    recu(arr,n,i_max);
    writeln('Результат:',i_max);
end.
```



## 7. Результаты работы программы

*Пример №1:*

```
Введите размер последовательности, не превышающий 100: 7
Вводите последовательность
11 63 80 32 69 21 49
Результат итеративной подпрограммы: 2
Результат рекурсивной подпрограммы: 2
```

*Пример №2:*

```
Введите размер последовательности, не превышающий 100: 7
Вводите последовательность
10 9 8 7 6 5 1337
Результат итеративной подпрограммы: 7
Результат рекурсивной подпрограммы: 7
```

## 8. Анализ ошибок

- Изначально в рекурсивной функции возникала «ошибка переполнения стека»
- Пропустил точку с запятой после `i_max`