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**

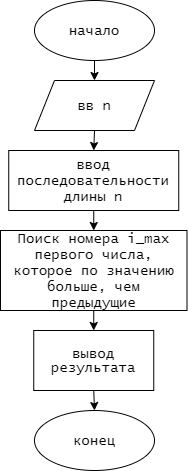
Найти номер первого вхождения максимального значения в последовательности длины n.

**Выполнение работы:**

1. **Выделение подзадач**

Выделим следующие подзадачи:

1. Ввод n
2. Ввод последовательности arr длины n
3. Поиск номера i\_max первого числа, которое по значению больше предыдущих
4. Вывод результата
5. **Блок-схема алгоритма с укрупнёнными блоками**



1. **Описание структур данных**

SIZE – константа, описывающая максимальный размер последовательности

t\_arr – тип, описывающий последовательность размера SIZE

**const**

SIZE = 100;

**type**

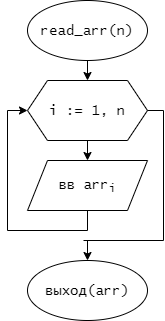
t\_arr = **array**[1..SIZE] **of** integer;

1. **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

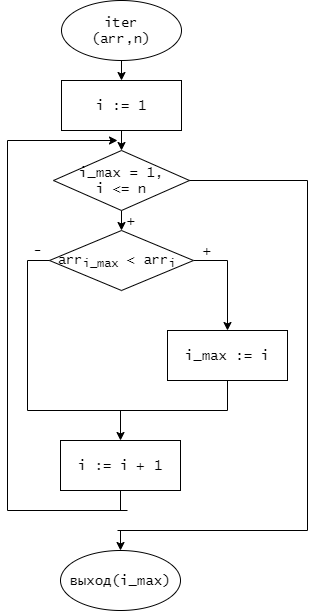
Блок-схема:



Спецификация процедуры iter

1. Заголовок: procedure iter(arr: t\_arr; n: integer; var i\_max: 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

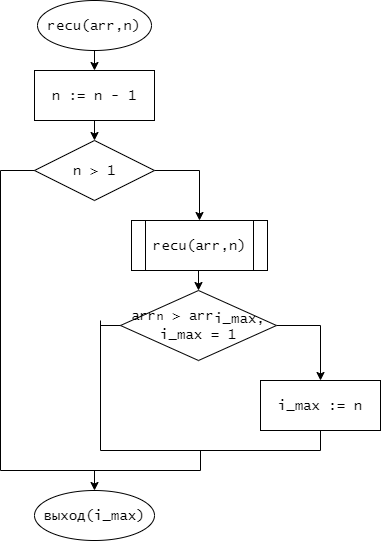
Блок-схема



Спецификация процедуры recu

1. Заголовок: procedure recu(arr: t\_arr; n: integer; var i\_max: 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:



1. **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 |

1. **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] > arr[i\_max]) **then**

i\_max := i;

i := i + 1;

**end**;

**end**;

**var**

n, i\_max: integer;

arr: t\_arr;

**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 > 1) **then**

**begin**

recu(arr,n,i\_max);

**if** (arr[n] > arr[i\_max]) **and** (i\_max = 1) **then**

i\_max := n;

**end**;

**end**;

**var**

n, i\_max: integer;

arr: t\_arr;

**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);

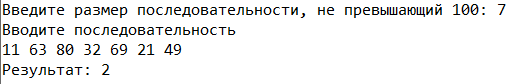
writeln('Результат:',i\_max);

**end**.

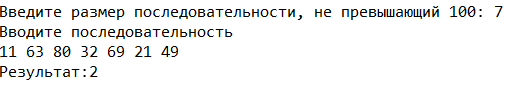
1. **Результаты работы программы**

*Пример №1:*

*Итеративная:*

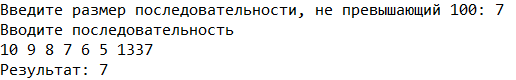


*Рекурсивная*

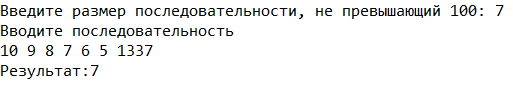


*Пример №2:*

*Итеративная:*

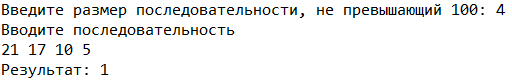


*Рекурсивная:*

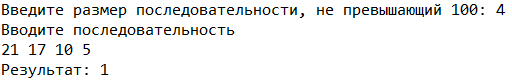


*Пример №3*

*Итеративная:*



*Рекурсивная:*



1. **Анализ ошибок**

* Изначально в рекурсивной функции возникала «ошибка переполнения стека»
* Пропустил точку с запятой после i\_max