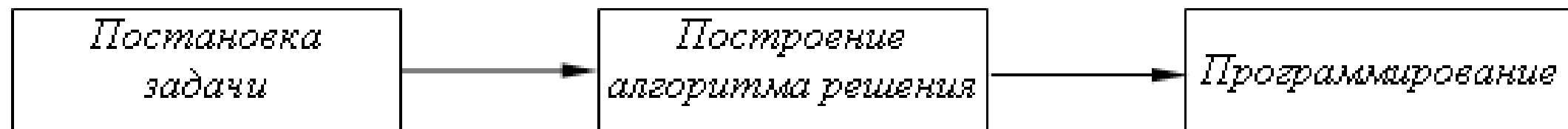


КУРС «ИНФОРМАТИКА»

Часть 1. Построение схем
алгоритмов

2021 – 2022 УЧЕБНЫЙ ГОД

Процесс решения задач на ПК



- **Постановка задачи** – это точная формулировка решения задачи на компьютере с описанием входной и выходной информации рассматриваемой предметной области.

Постановка задачи связана с конкретизацией основных параметров ее реализации, определением источников и структурой входной и выходной информации, востребованной пользователем.




Построение алгоритма решения задачи

- *Алгоритм* – последовательность действий, приводящая к заданному результату за конечное число шагов.


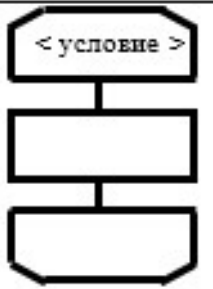

Алгоритм должен обладать следующими свойствами:

- *дискретностью* – разбиением процесса обработки информации на более простые этапы (шаги), выполнение которых компьютером или человеком не вызывает затруднений;
- *определенностью* (детерминированностью) – однозначностью получаемого результата при одних и тех же исходных данных;
- *результативностью* – обязательным получением желаемого результата за конечное число шагов при допустимых исходных данных;
- *массовостью* – применимостью алгоритма для решения определенного класса задач.

Базовые структуры

Графическое обозначение	Наименование	Пояснения	Соответствующие структуры языка Python
	Пуск-останов	Начало, конец, прерывание процесса обработки данных	
	Процесс, действие	Операция, в результате которой изменяются значение данных Может быть составным (из нескольких операторов)	<pre>a=b <оператор 1> ... <оператор k></pre>
	Ввод-вывод данных	Ввод-вывод без указания конкретного носителя	<code>input/print</code>
	Условие	Разветвление алгоритма в зависимости от некоторых условий	<pre>if <условие 1>: <оператор 1> elif <условие 2>: <оператор 2> else: <оператор 3></pre>
	Программа, подпрограмма	Часть алгоритма, требующая дополнительной детализации на последующих шагах	<pre>def <имя_функции> (<параметры>): <операторы></pre>

Циклические структуры

Графическое обозначение	Наименование	Пояснения	Соответствующие структуры языка Python
	Итеративный цикл	Повторяется заданное количество раз Используется для последовательной обработки всех элементов	<pre>for <переменная> in <последовательность>: <оператор> for <переменная> in <функция_range>: <оператор></pre> <p>Есть три способа вызова range():</p> <ol style="list-style-type: none"> 1. range(стоп) 2. range(старт, стоп) 3. range(старт, стоп, шаг)
	Цикл с предусловием	Может ни разу не выполниться (если не сработает условие входа) Используется для поиска элементов (первый встретившийся, последний встретившийся)	<pre>while <условие продолжения>: <оператор></pre>
	Цикл с постусловием	Выполняется хотя бы один раз. Если условие выхода не сработало, возвращается к началу цикла. Используется для организации циклов многократного повторения некоторых блоков программы.	<p>В Python цикл с постусловием отсутствует!</p> <pre>while True: if <условие выхода>: break</pre> <p>Оператор break используется для выхода из цикла. Python – прерывает его досрочно</p>

Вспомогательные структуры

Графическое обозначение	Наименование	Пояснения
	Документ	Ввод-вывод данных, носителем которых является бумага.
	Дисплей	Ввод-вывод данных на монитор.
	Магнитный диск	Ввод-вывод данных на магнитный диск (в файл).
	Комментарий	Связь между элементом схемы и пояснением к нему.
	Соединители	Связь между прерванными линиями на одной страницы, связь между прерванными частями схем на разных стр.

КУРС «ИНФОРМАТИКА»

Часть 2. Построение схем алгоритмов
для одномерных массивов

2021 – 2022 УЧЕБНЫЙ ГОД

Одномерные массивы (векторы)

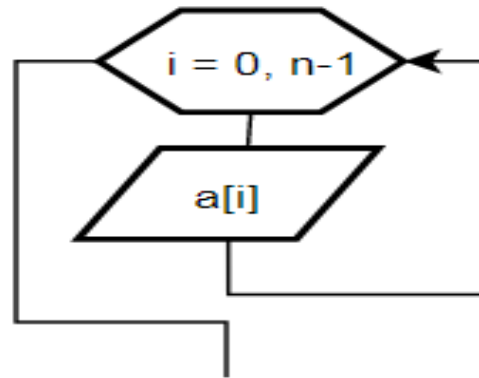
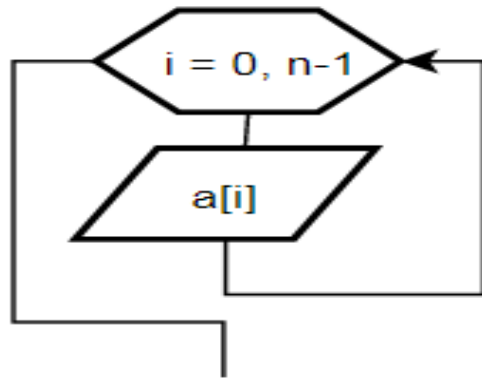


Массив – это совокупность однородных элементов, расположенных в памяти последовательно друг за другом и имеющих общее имя.

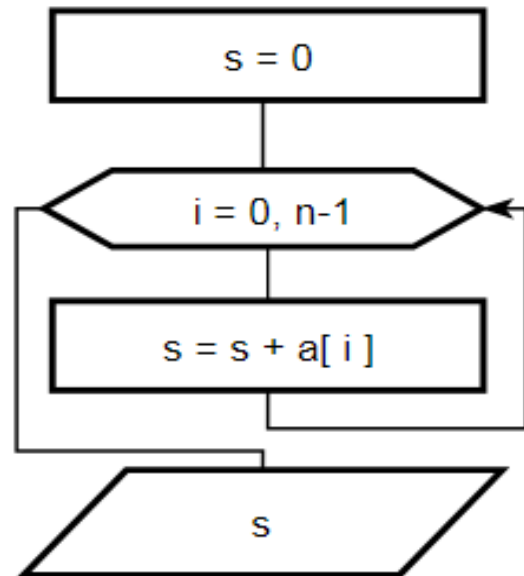
К любому элементу массива можно обратиться по его номеру (индексу)

Базовые алгоритмы обработки

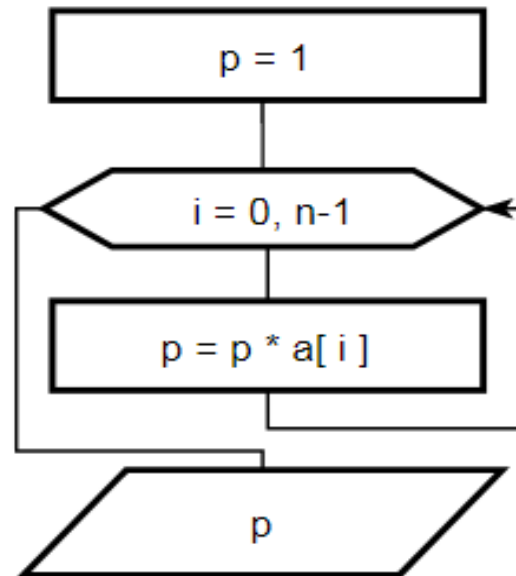
1. Ввод элементов массива 2. Вывод элементов массива



3. Сумма элементов массива



4. Произведение элементов массива



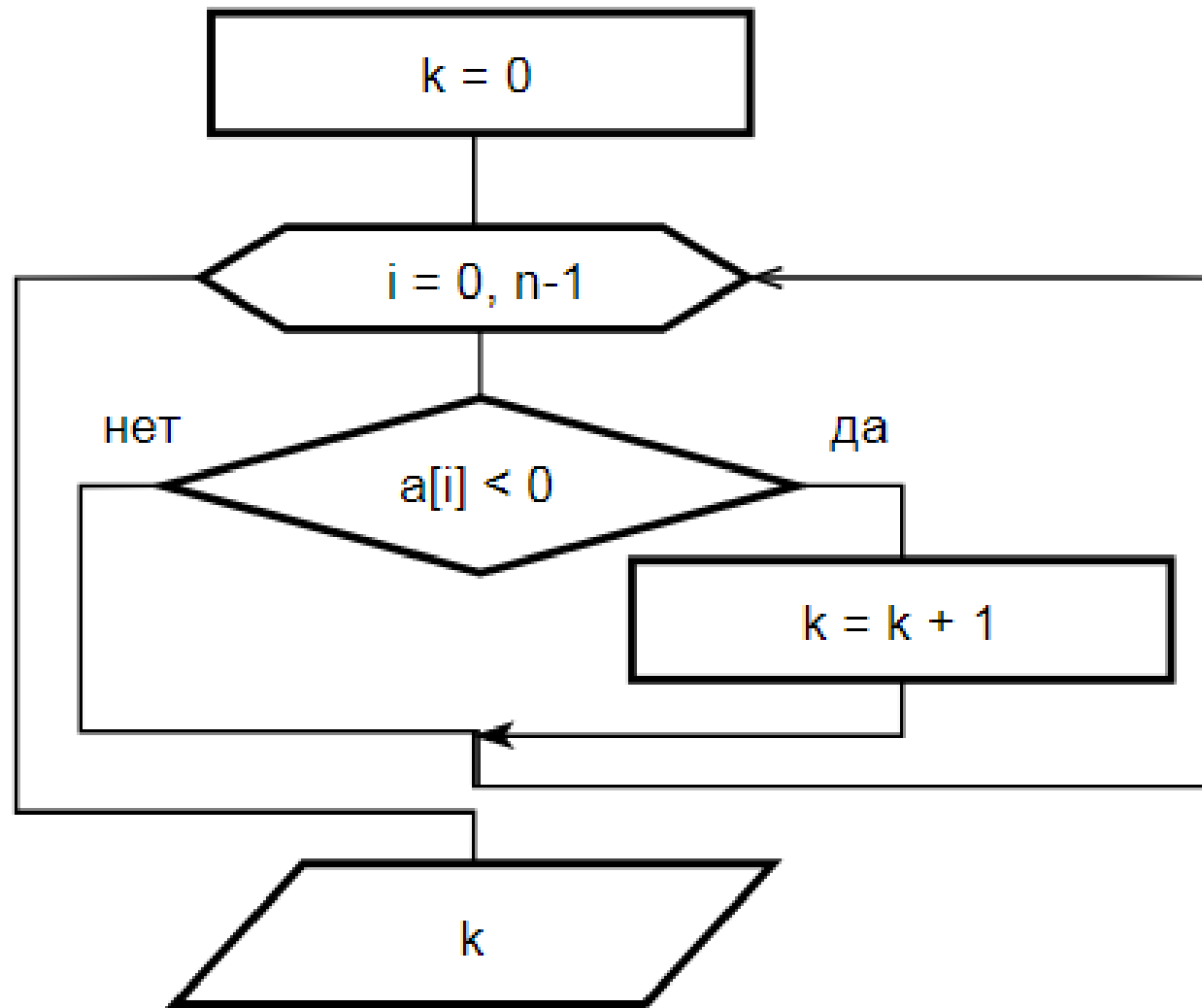
5	-1	0	7	8	2	2	3
0	1	2	...	i	...		n-1

$s = 0$

$p = 1$

Базовые алгоритмы обработки

5. Количество отрицательных элементов

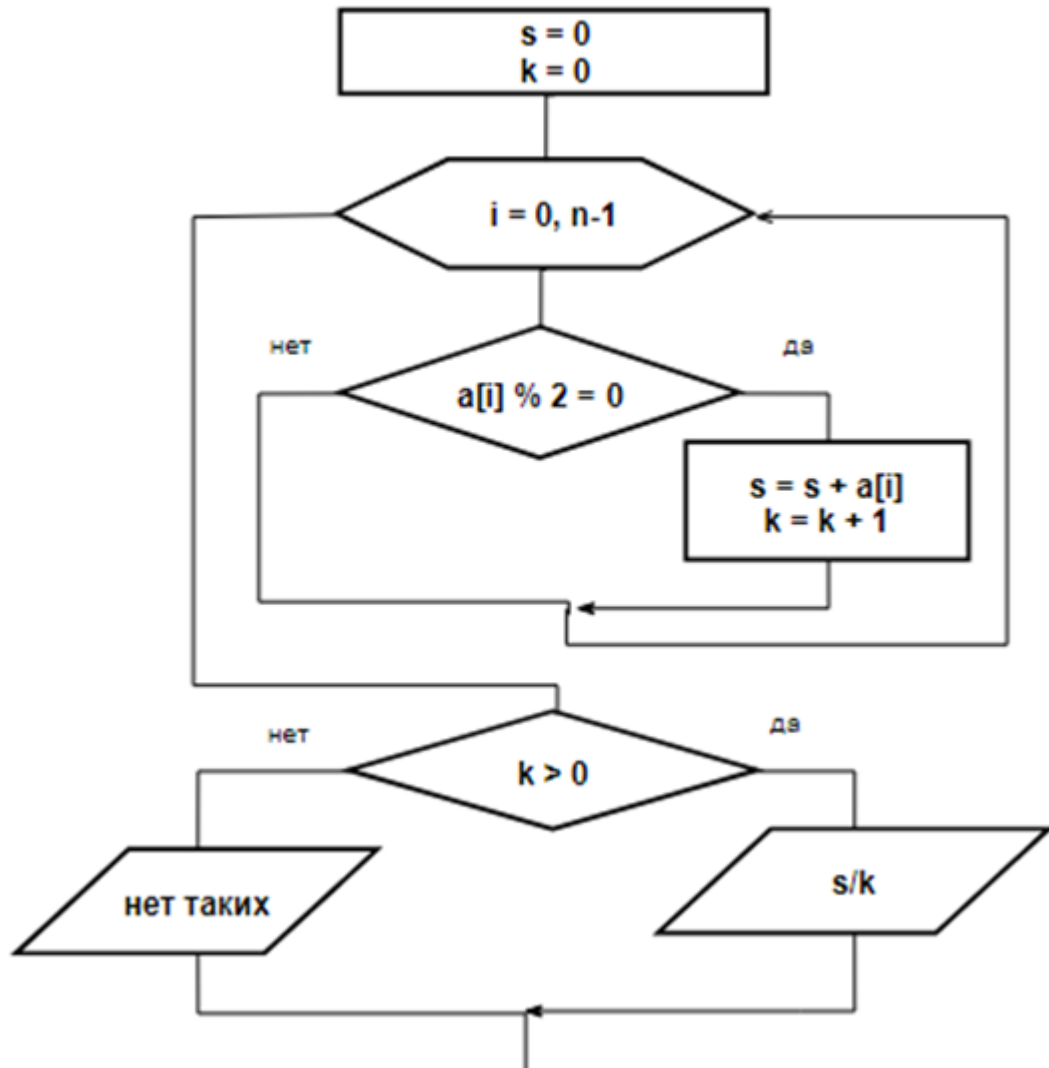


7	-5	9	1	0	-2	4	3	6
0	1	2	...	i	...			n-1

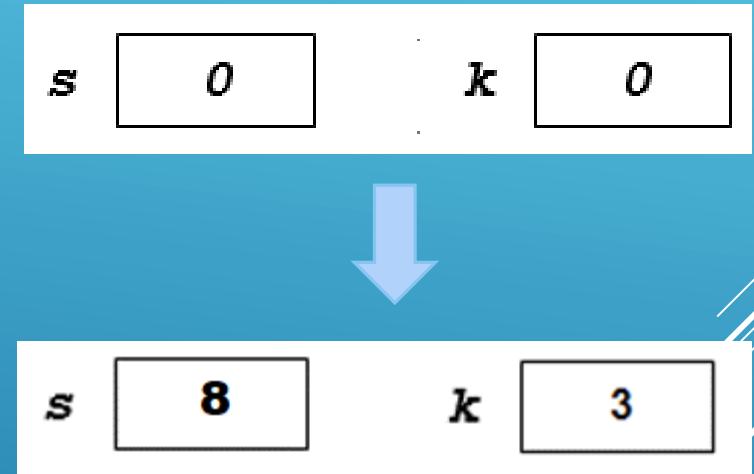
k 0

Базовые алгоритмы обработки

6. Среднее арифметическое четных элементов

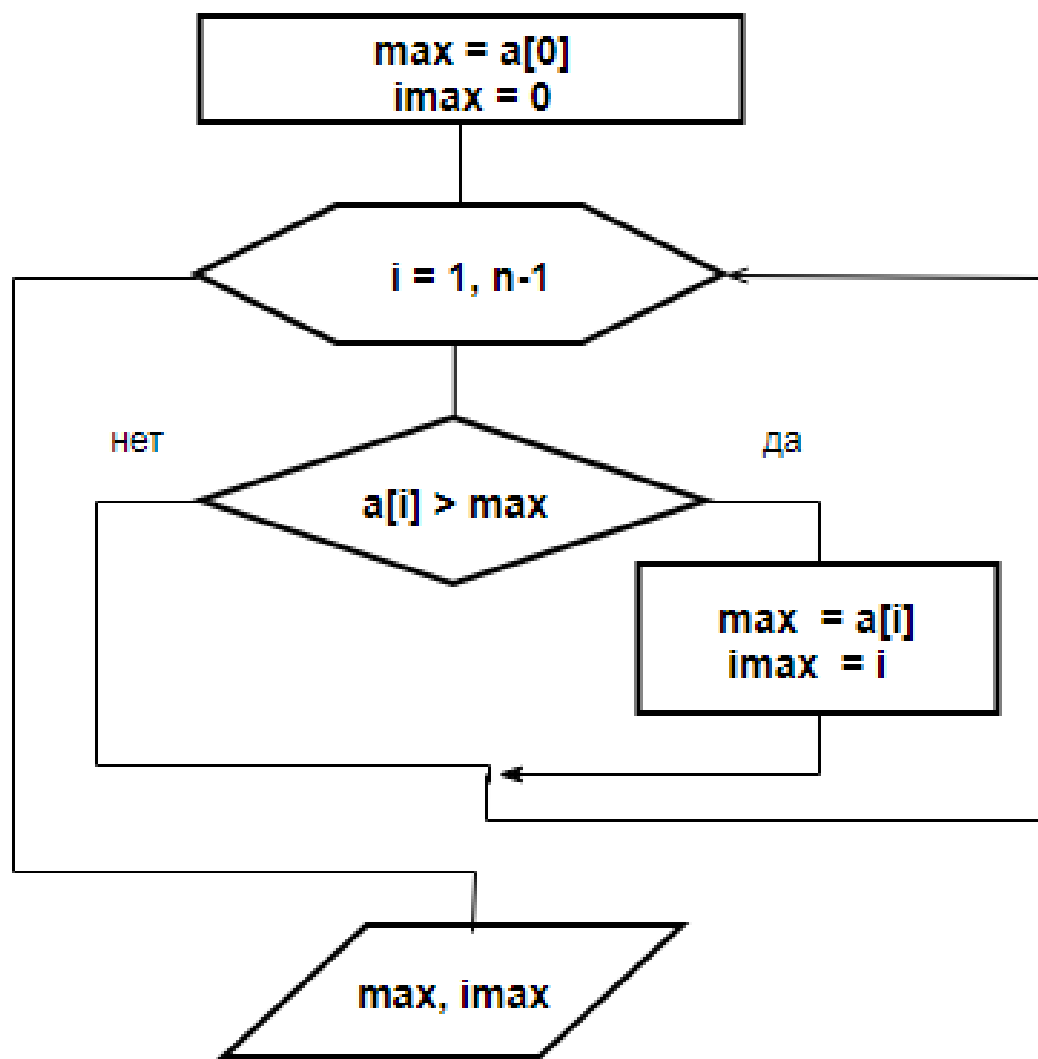


7	-5	9	1	1	-2	4	3	6
0	1	2	...	i	...			n-1

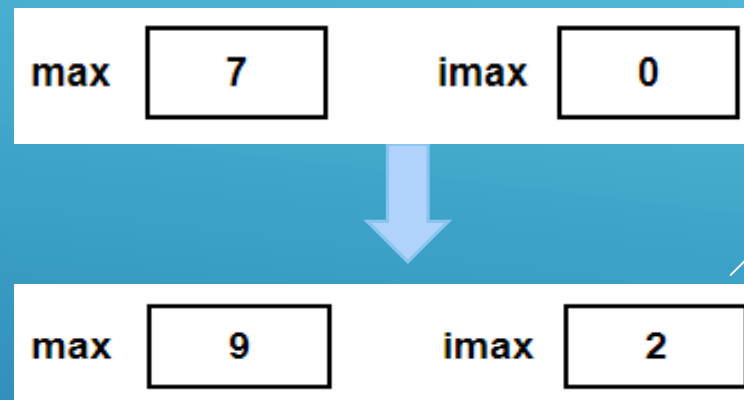


Базовые алгоритмы обработки

7. Максимальный элемент и его номер

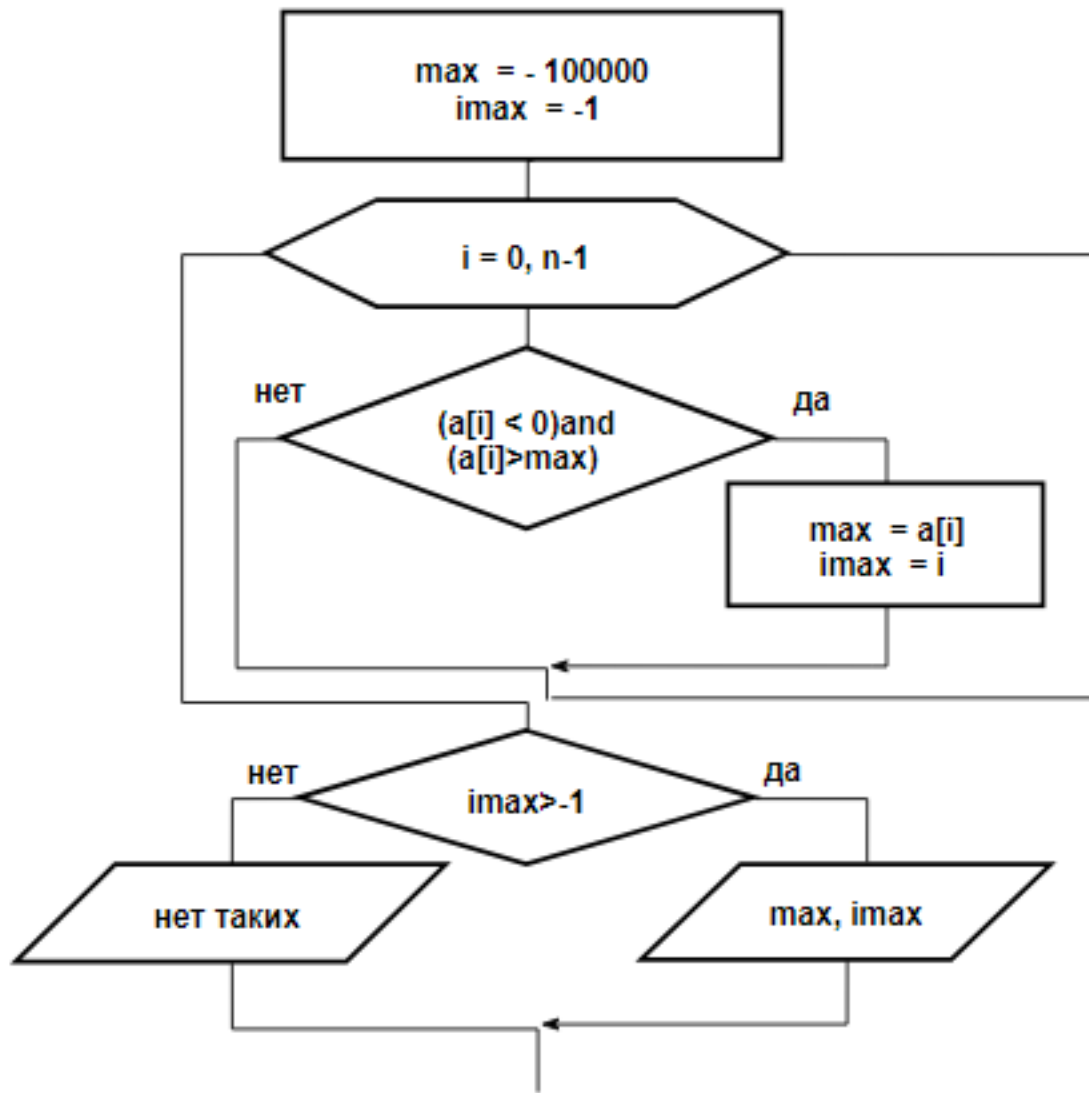


7	-5	9	1	0	-2	4	3	6
0	1	2	...	i	...			n-1

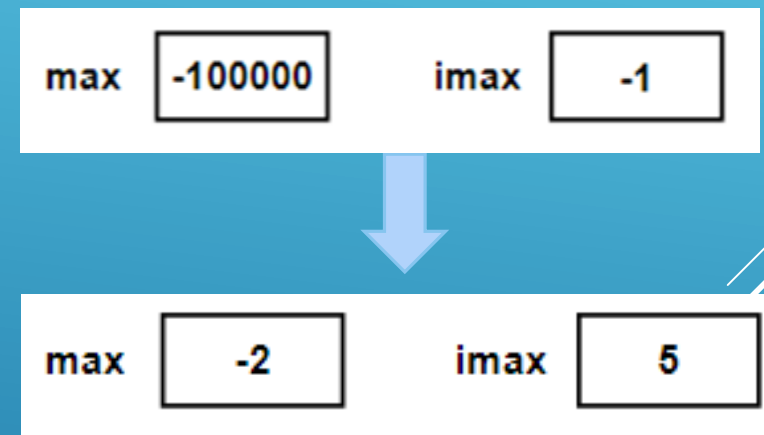


Базовые алгоритмы обработки

8. Максимальный отрицательный элемент и его номер

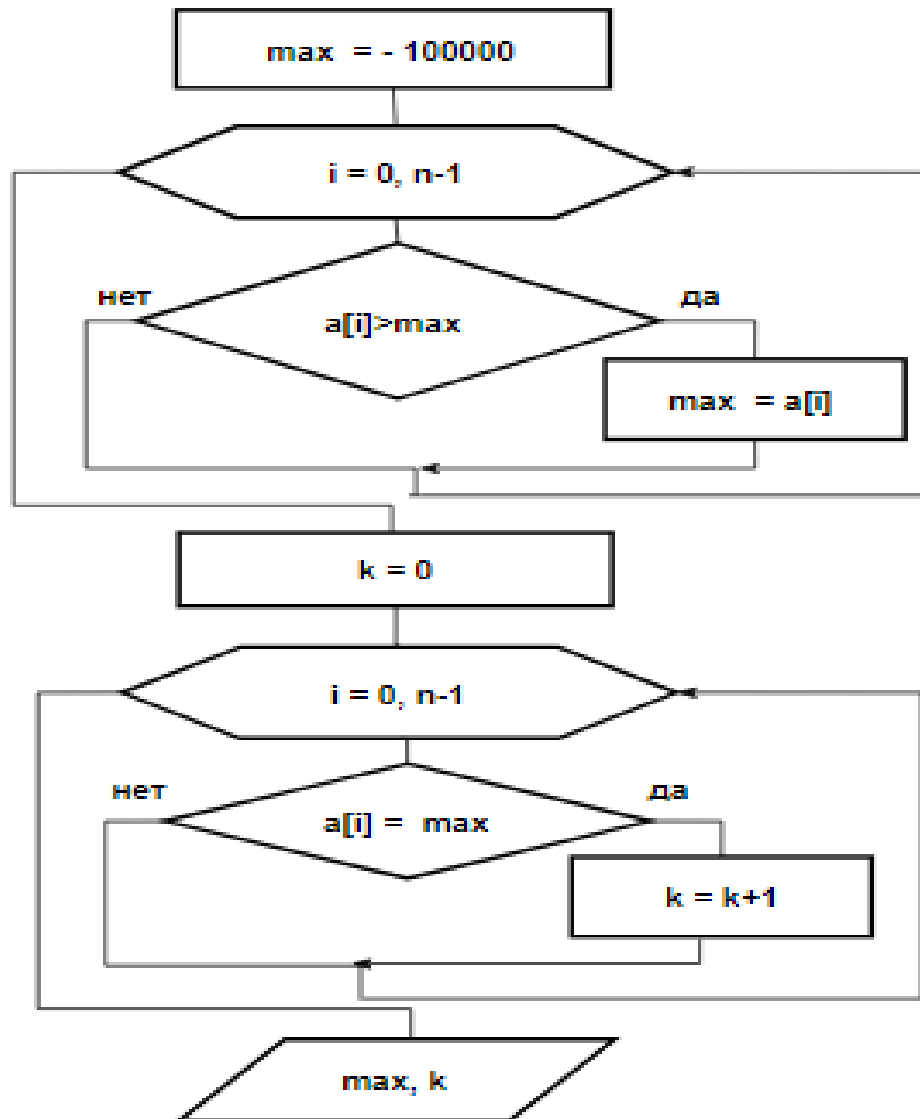


7	-5	9	1	0	-2	4	3	6
0	1	2	...	i	...			n-1



Базовые алгоритмы обработки

9. Количество максимальных элементов



7	-5	9	1	0	-2	9	3	6
0	1	2	...	i	...			n-1

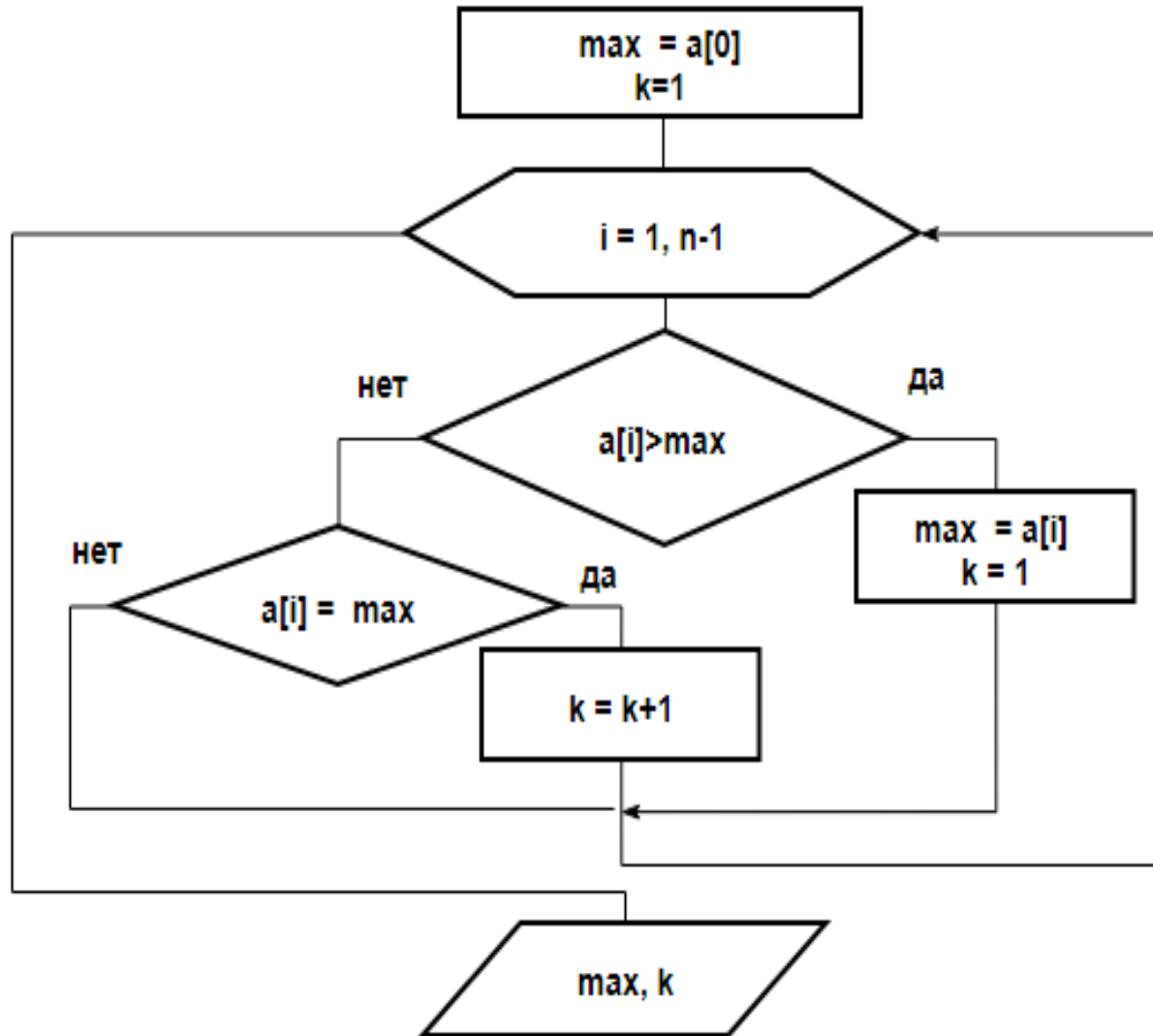
max -100000 k 0



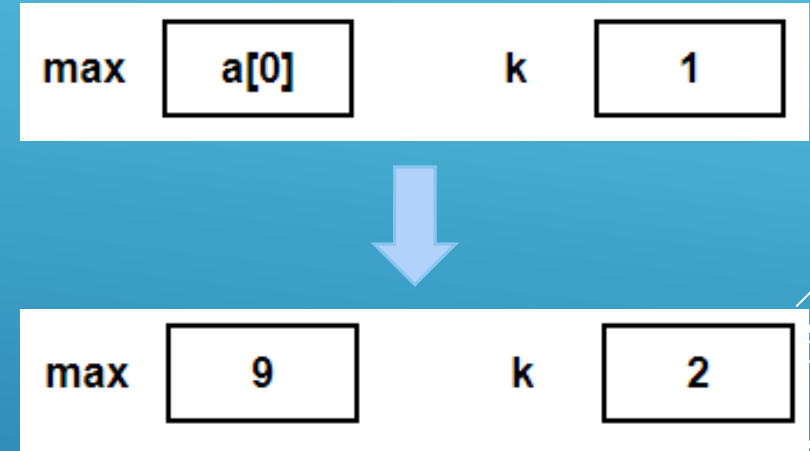
max 9 k 2

Базовые алгоритмы обработки

10. Количество максимальных элементов (за один проход массива)

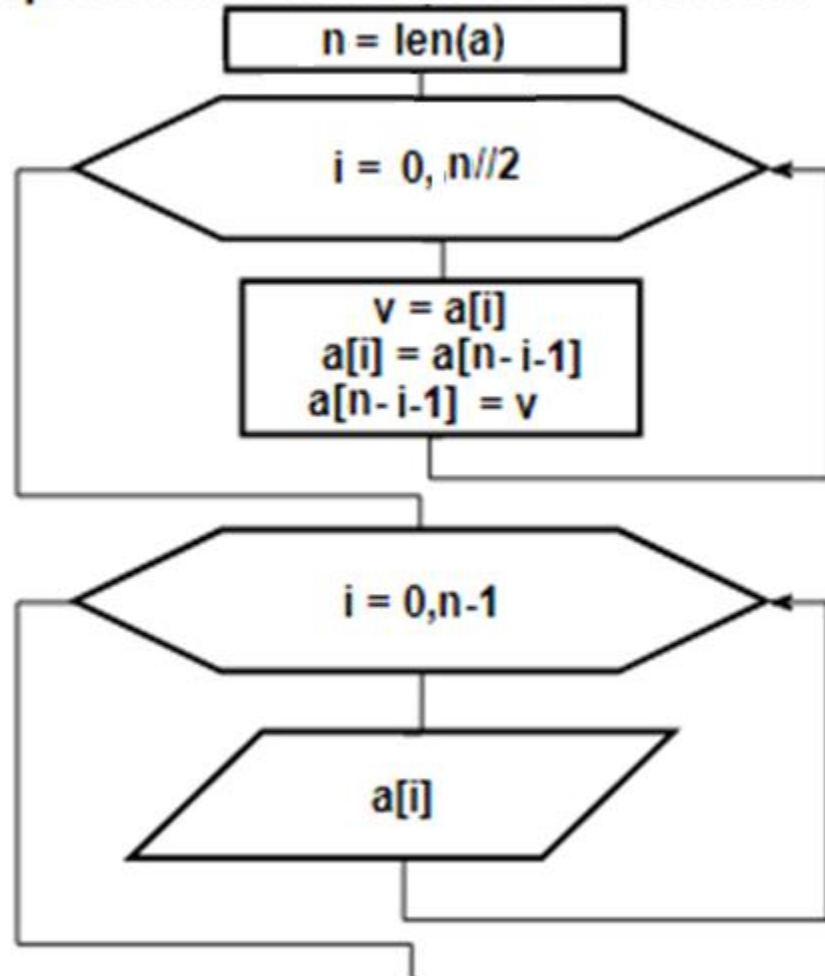


7	-5	9	1	0	-2	9	3	6
0	1	2	...	i	...			n-1



Базовые алгоритмы обработки

11. Переписать массив в обратном порядке



7	-5	9	1	0	-2	4	3	6
0	1	2	...	i	...			$n-1$

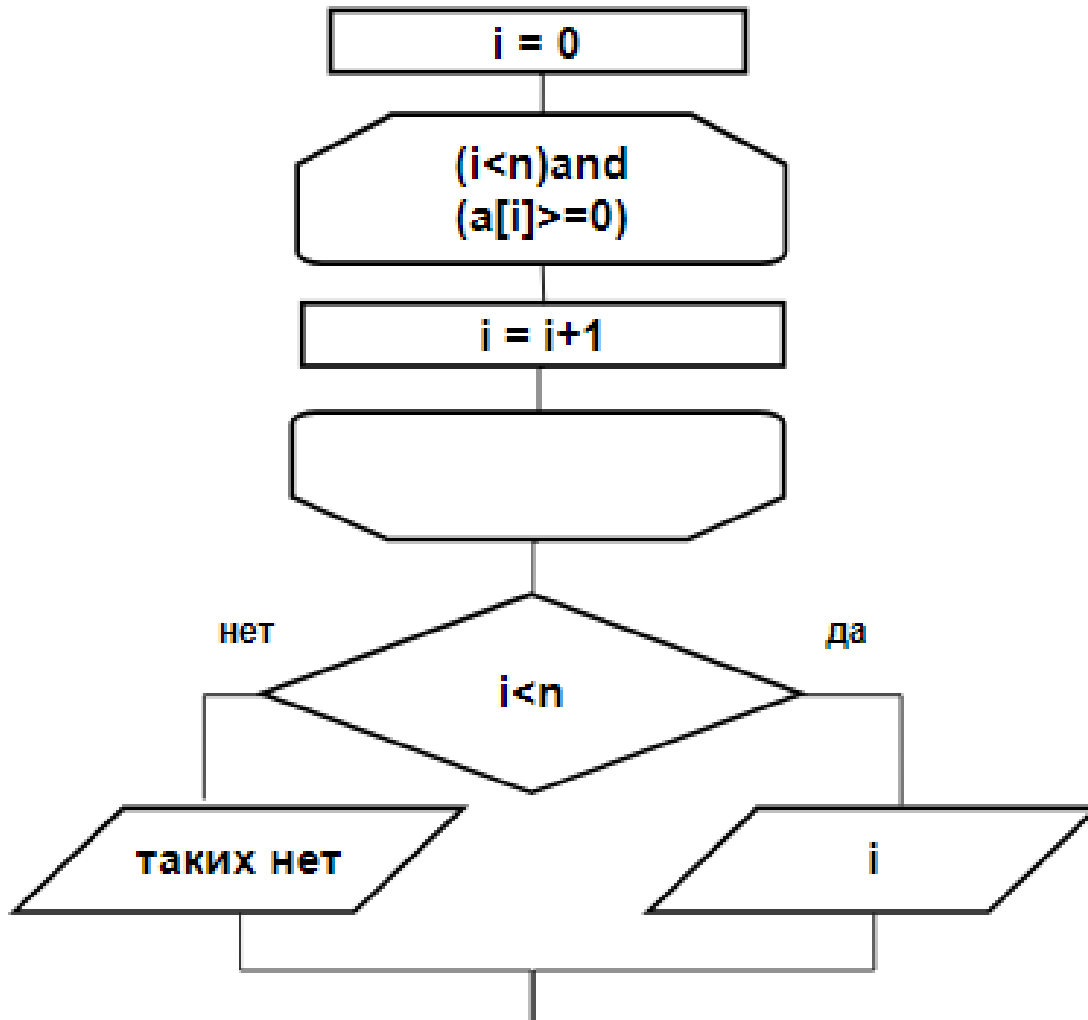
$n = \text{len}(a)$	
0	$n-1$
1	$n-2$
2	$n-3$
3	$n-4$

i	$n-i-1$
$a[i]$	$a[n-i-1]$

6	3	4	-2	0	1	9	-5	7
0	1	2	...	i	...			$n-1$

Базовые алгоритмы обработки

12. Номер первого отрицательного элемента



7	-5	9	1	0	-2	4	3	6
0	1	2	...	i	...			n-1

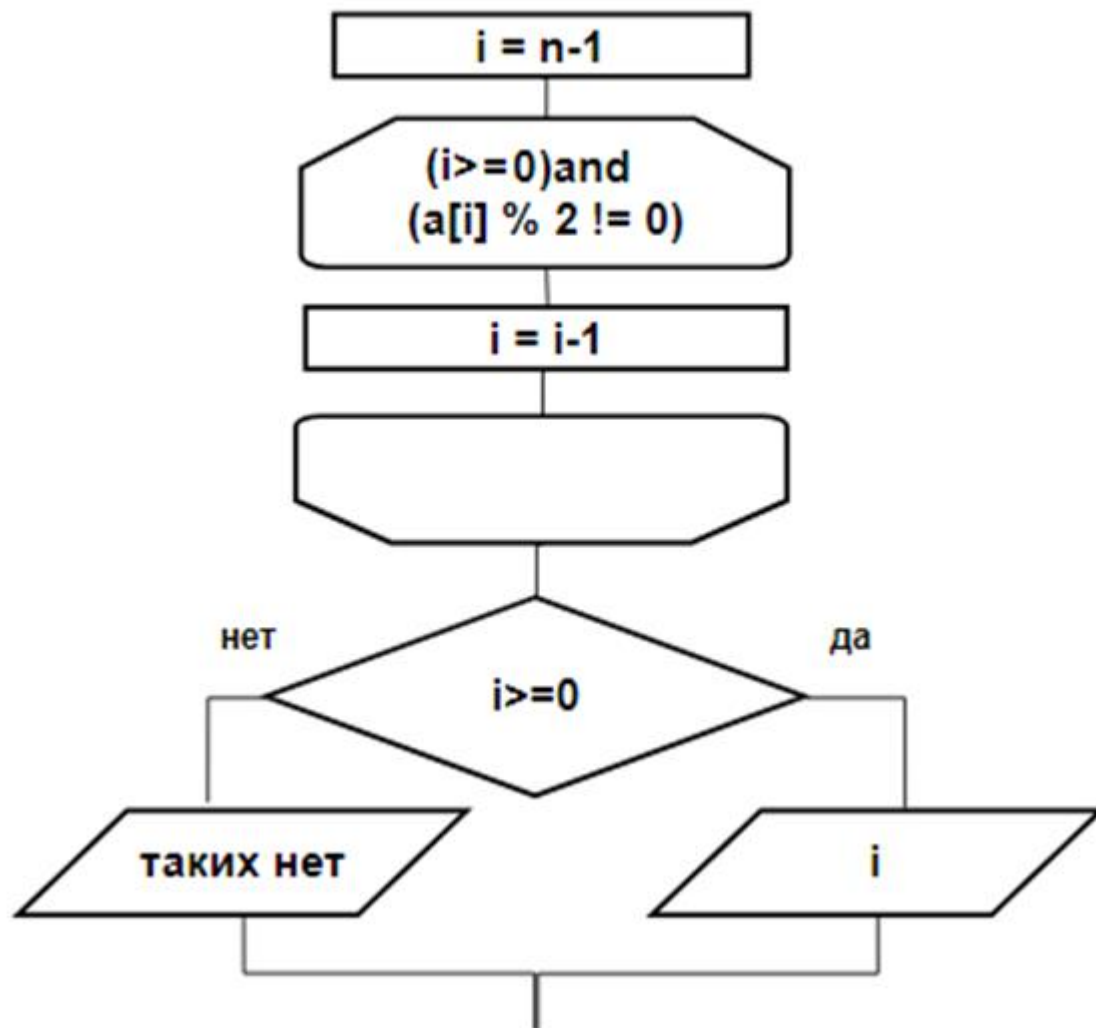
$i = 1$

7	5	9	1	0	2	9	3	6
0	1	2	...	i	...			n-1

$i = 9$
 $i \geq n$
нет таких

Базовые алгоритмы обработки

13. Номер последнего четного элемента



7	-5	9	1	0	-2	4	3	6
0	1	2	...	i	...			n-1

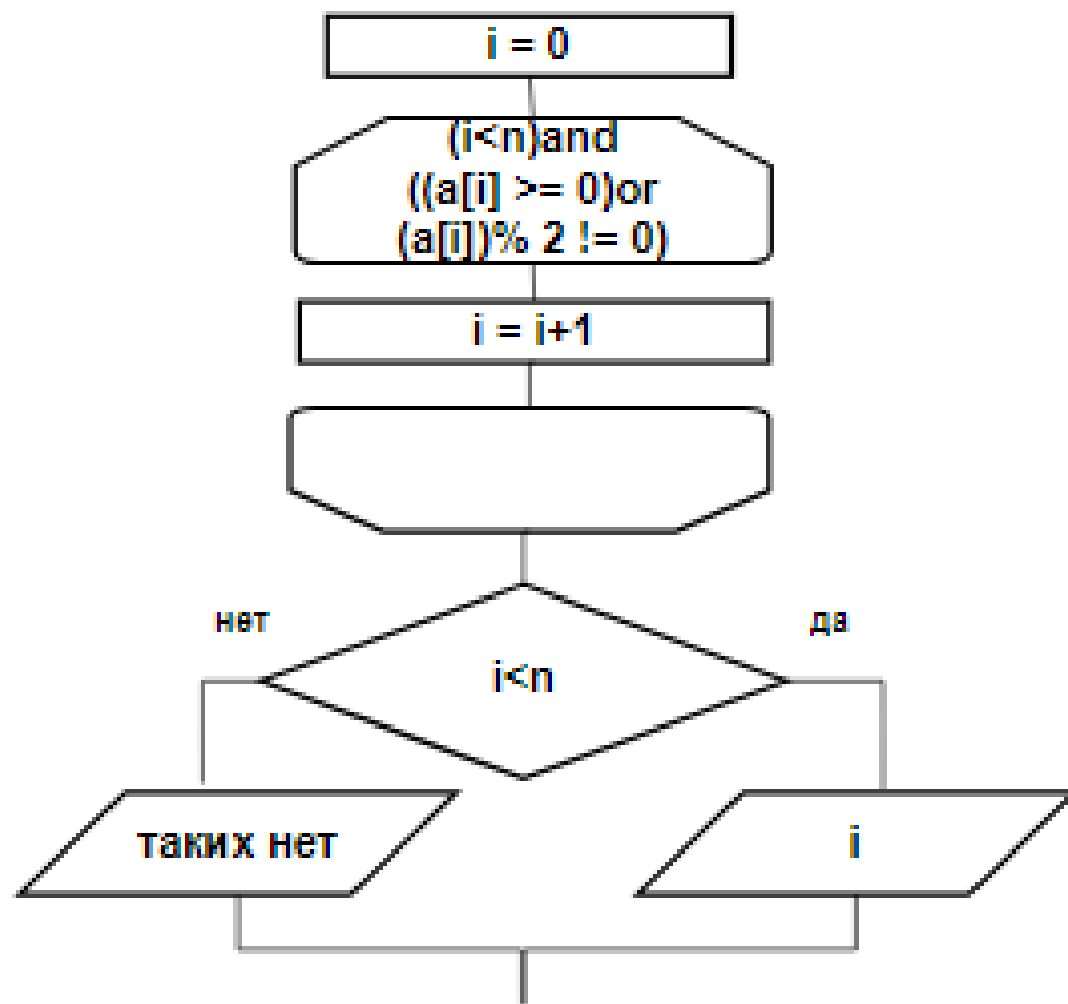
$i = 8$

7	5	9	1	5	3	9	3	7
0	1	2	...	i	...			n-1

$i = -1$
нет таких

Базовые алгоритмы обработки

14. Номер первого отрицательного четного числа



7	-5	9	1	0	-2	4	3	6
0	1	2	...	i	...			n-1

$i = 5$

7	5	9	2	5	4	9	3	7
0	1	2	...	i	...			n-1

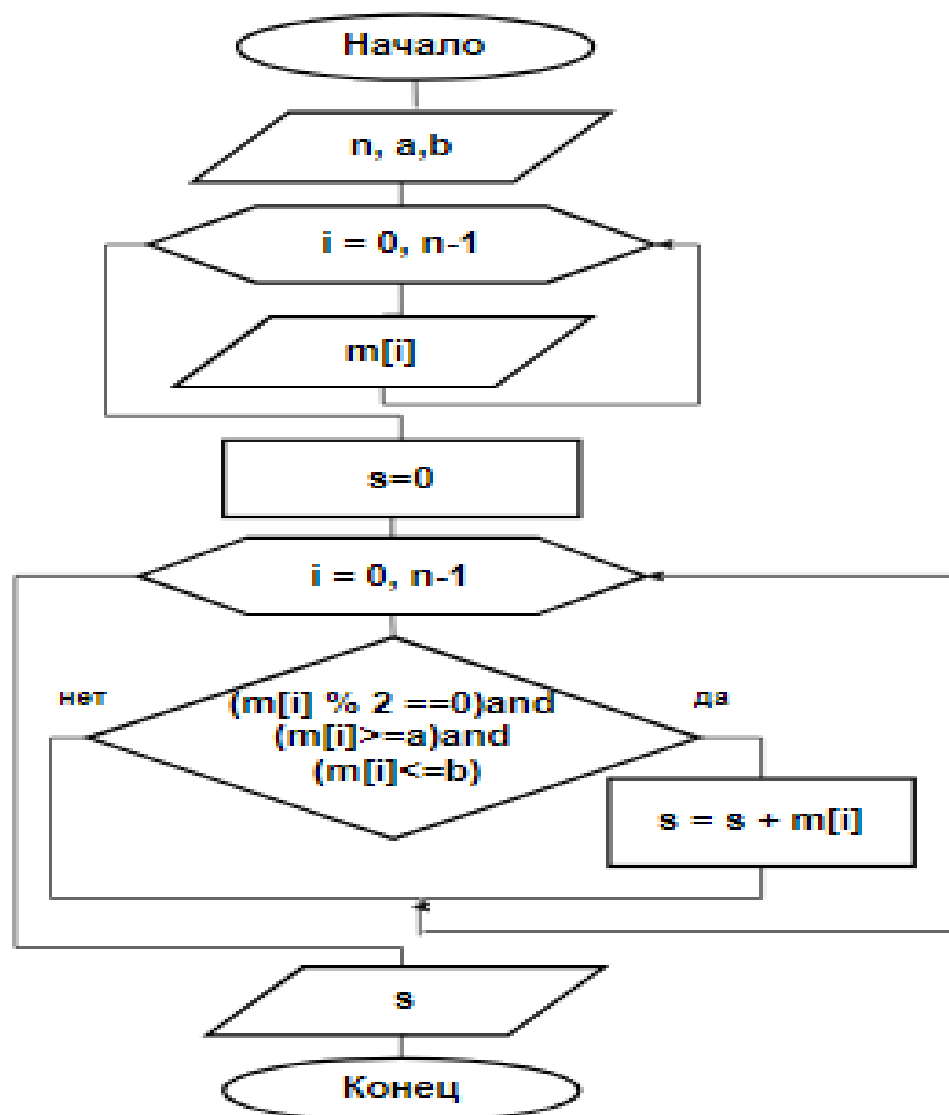
$i = 9$
 $i \geq n$
нет таких

Примеры заданий

№	Задача 1	Задача 2
1	Найти сумму четных (по значению) элементов массива, попадающих в диапазон от a до b. массив m	С помощью цикла WHILE найти индекс первого четного элемента массива
2	Найти произведение всех нечетных (по индексу) элементов массива	С помощью цикла WHILE найти индекс последнего нулевого элемента массива
3	Найти среднее арифметическое отрицательных нечетных (по индексу) элементов массива	С помощью цикла WHILE найти индекс последнего отрицательного нечетного элемента
4	Найти количество нечетных положительных элементов массива	С помощью цикла WHILE найти индекс первого нечетного элемента, кратного

Задача 1

Найти сумму четных (по значению) элементов массива, попадающих в диапазон от a до b.

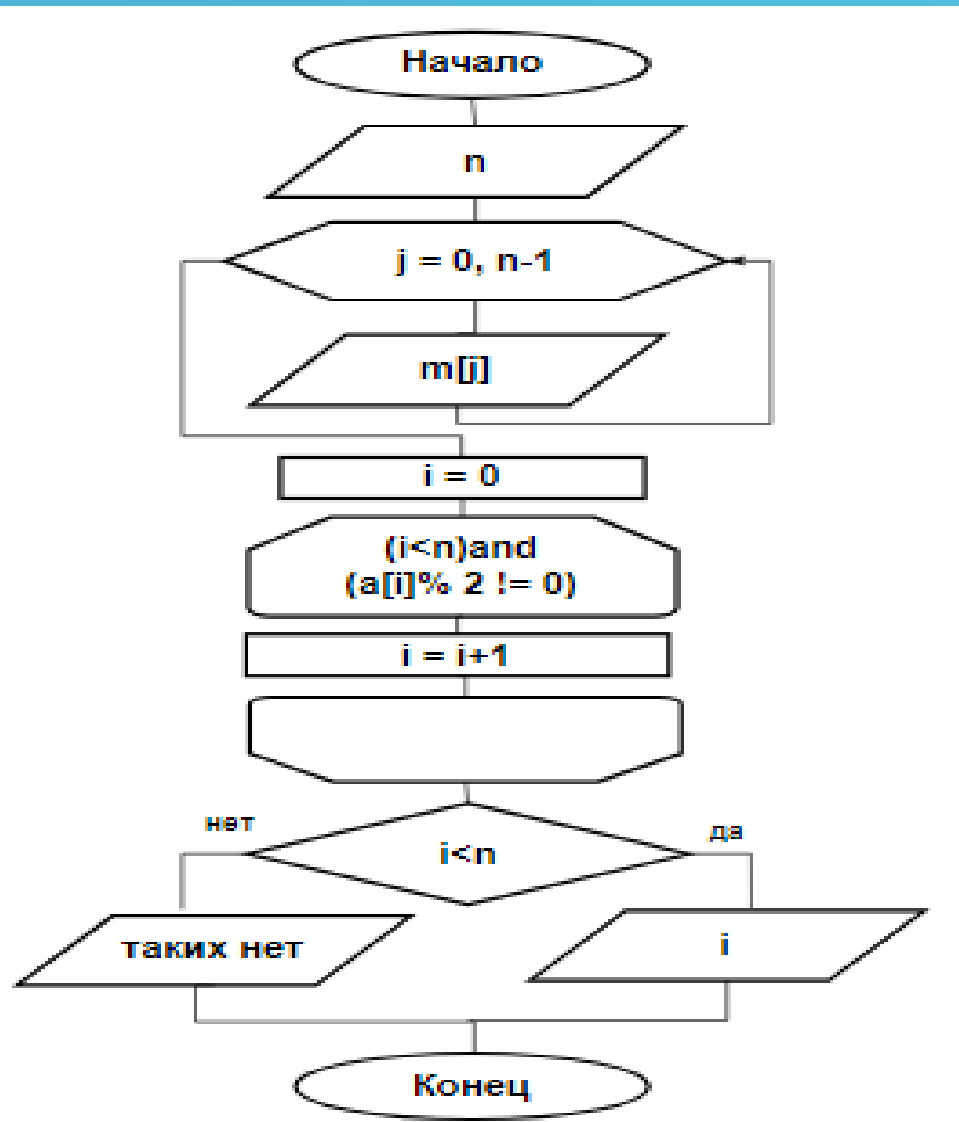


```
n = int(input("Введите длину n массива m: "))
m = []
for i in range(n):
    m.append(int(input("Введите элемент массива m:")))
print("Массив:" + str(m))
a = int(input("Введите a: "))
b = int(input("Введите b: "))
print("a=" + str(a) + "b=" + str(b))

s = 0
for i in range(0, len(m)):
    if ((m[i] % 2 == 0) and (m[i] >= a) and (m[i] <= b)):
        s = s + m[i]
print("Сумма =" + str(s))
```

Задача 2

С помощью цикла WHILE найти индекс первого четного элемента массива.



```
n = int(input("Введите длину n массива m: "))
m=[]
for j in range(n):
    m.append(int(input("Введите элемент массива m:")))
print("Массив:" ,m)

i=0
while((i<n)and(m[i]%2!=0)):
    i=i+1
print("i =" ,i)
```

Содержание контрольной работы № 1

Обработка одномерных массивов

1 Задача (for, while – выбор по целесообразности)

1.1 Схема алгоритма

1.2 Программа обработки

2 Задача (обязательное использование while)

2.1 Схема алгоритма

2.2 Программа обработки

Источники

Теоретический материал



Тема 3 - Методы работы с процедурной информацией.doc

Обзорный материал (данная презентация)



КУРС «ИНФОРМАТИКА». Онлайн - Лекция 2. Построение схем алгоритмов