

Лекция 3

Понятие алгоритма, свойства, виды (типы). Формы записи. Исходные и выходные данные алгоритмов, примеры

Разработала Фаерштейн Л.

В.

СОДЕРЖАНИЕ

1. Определение и

свойства

2. Формы

записи

3. Исходные и выходные данные

алгоритмов

4. Примеры

алгоритмов

Выход

ОПРЕДЕЛЕНИЕ И СВОЙСТВА

Определение.

Алгоритм – определенная последовательность действий, вместе с исходными данными приводящая к решению задачи. Алгоритм -перечень действий для решения любых задач, а не только математики и информатики.

Свойства

а) дискретность (процесс решения задачи разбит на отдельные шаги – этапы)

ОПРЕДЕЛЕНИЕ И СВОЙСТВА (ПРОДОЛЖЕНИЕ)

б) **определенность** (каждое действие должно быть понятно исполнителю алгоритма)

в) **конечность и результативность** (алгоритм всегда приводит к результату за конечное число шагов)

г) **универсальность** (может быть использован для всех задач одного и того же класса при разных исходных данных)

ОПРЕДЕЛЕНИЕ И СВОЙСТВА (ПРОДОЛЖЕНИЕ)

Алгоритм всегда создается для исполнителя, т. е. того, кто его будет выполнять. Под исполнителем понимается любое **техническое устройство**, а также **человек** или **животное**. Мы будем рассматривать алгоритмы, где исполнителем является **ЭВМ** (т.е техническое устройство).

ОПРЕДЕЛЕНИЕ И СВОЙСТВА (ПРОДОЛЖЕНИЕ)

Алгоритм составляется по
определенным правилам:

- а) проводятся **формализация** задачи, т. е. перевод ее в математическую форму
- б) **определение шагов (этапов)** задачи, которые исполнитель может выполнить без пояснений

ОПРЕДЕЛЕНИЕ И СВОЙСТВА (ПРОДОЛЖЕНИЕ)

- в) определение порядка этапов**
- г) определение исходных и выходных данных, а также формы их представления**
- д) признак завершения процесса**

ОПРЕДЕЛЕНИЕ И СВОЙСТВА (ПРОДОЛЖЕНИЕ)

Алгоритм для ЭВМ должен состоять из:

а) операций, понятных ЭВМ

б) должен быть написан на понятном ЭВМ языке

ОПРЕДЕЛЕНИЕ И СВОЙСТВА (ПРОДОЛЖЕНИЕ)

Таких языков в настоящее время существует несколько сотен (называются ЯПВУ), старейшими из них являются **Basic** (1964), **Алгол** (1961), **Фортран** (1961). **Паскаль** появился в 1974 году, существует несколько версий языка (и не только Паскаля, но и других). В настоящее время активно используется язык **С** и его модификации

ОПРЕДЕЛЕНИЕ И СВОЙСТВА (ПРОДОЛЖЕНИЕ)

Запись алгоритма на языке для ЭВМ называется программой, а перевод алгоритма на язык называется программированием.



ОПРЕДЕЛЕНИЕ И СВОЙСТВА (ПРОДОЛЖЕНИЕ)

Типы алгоритмов:

- а) линейный – все этапы выполняются в строгой последовательности
- б) условный – алгоритм, в котором выбирается один из нескольких возможных вариантов решения в зависимости от условия. Каждый вариант называется ветвью, отсюда второе название алгоритмов – ветвления.

ОПРЕДЕЛЕНИЕ И СВОЙСТВА (ПРОДОЛЖЕНИЕ)

Условия бывают двух видов: простые и составные

- **Простое:** состоит из двух величин или выражений, связанных знаками $<$, $>$, $=$, $<>$, $>=$, $<=$
- **Составное:** состоит из нескольких простых условий, попарно связанных операциями НЕ, И, ИЛИ и круглых скобок ()

ОПРЕДЕЛЕНИЕ И СВОЙСТВА (ПРОДОЛЖЕНИЕ)

Операция проверки условия называется **логический блок**. Блоков столько, сколько условий. Условия имеют два значения: **истина** или **ложь** (1 или 0). Если условие выполняется – **истина**, если нет – **ложь**. Результат условия вычисляется по **таблице истинности Алгебры Буля**.

ОПРЕДЕЛЕНИЕ И СВОЙСТВА (ПРОДОЛЖЕНИЕ)

в) алгоритм **циклической структуры** – алгоритм, в котором какая-либо часть действий выполняется несколько раз или ни разу в зависимости от некоторого условия. Повторяющаяся часть действий называется **цикл**.

ОПРЕДЕЛЕНИЕ И СВОЙСТВА (ОКОНЧАНИЕ)

Циклов *три* типа:

- 1) с параметром (счетчиком) – параметр считает количество раз выполнения цикла
- 2) с предусловием – проверяется условие, а затем цикл либо выполняется, либо нет
- 3) с постусловием – один раз цикл выполняется **всегда**, а потом в зависимости от условия выполняется **далее** или **заканчивается**



ФОРМЫ ЗАПИСИ

Формы записи алгоритмов:

- Словесно-формульная – на естественном языке, куда включены некоторые служебные слова – *начало цикла, конец цикла* и т. д.
- Графическая – каждое действие обозначается какой-либо геометрической фигурой

Любой алгоритм начинается со знака
и им же заканчивается



ФОРМЫ ЗАПИСИ (ПРОДОЛЖЕНИЕ)



 линейный блок алгоритма, может быть укрупненным, т. е. включать несколько линейных этапов

 - ввод/вывод данных с (на) экрана(н)

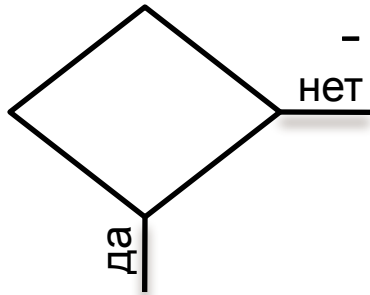
вывод данных на принтер (бумагу)



ФОРМЫ ЗАПИСИ (ОКОНЧАНИЕ)

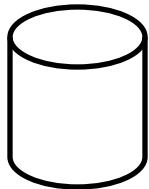


циклическое действие



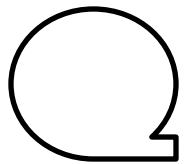
- логический блок (может иметь до

выходов, обычно имеется два)



- запись данных на магнитный диск,
магнитную ленту

ФОРМЫ ЗАПИСИ (ОКОНЧАНИЕ)



- вывод на магнитную ленту

Все эти обозначения приведены в
Фигурах офисных программ



ИСХОДНЫЕ И ВЫХОДНЫЕ ДАННЫЕ АЛГОРИТМОВ



Примеры констант

-100

0,5

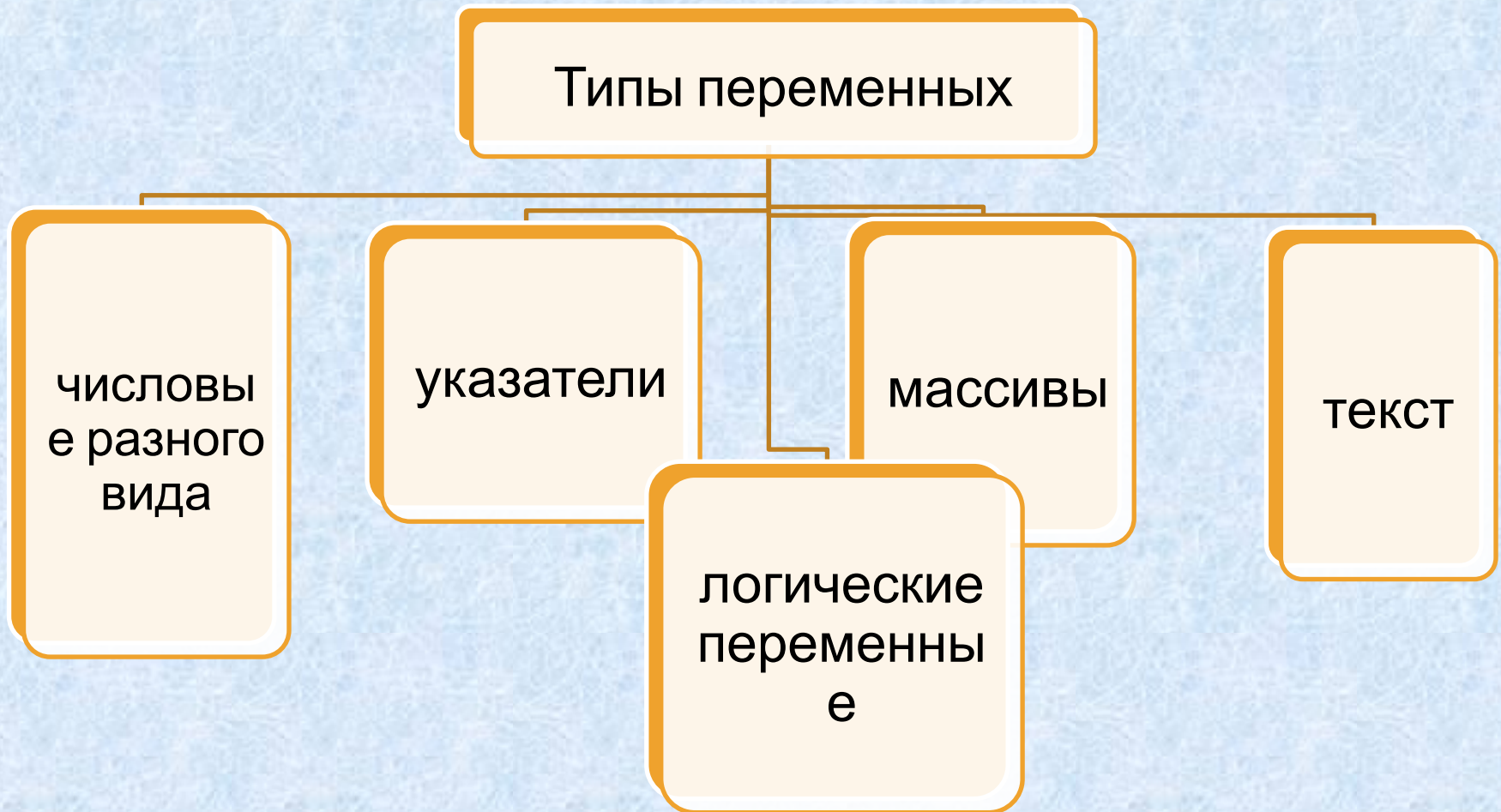
0,2E-10

False (логическая константа)

‘Пермь’ – это явное задание констант
K1=‘Пермь’ – задание текстовой константы именем

Примеры переменных
Задаются **именем**, состоящим из латинских букв, арабских цифр и некоторых служебных символов, например, **_**, начинаются всегда с буквы, обычно имеют длину до 40 символов
Например, L1, QU2, EX_3 и т.д.

ИСХОДНЫЕ И ВЫХОДНЫЕ ДАННЫЕ АЛГОРИТМОВ (ПРОДОЛЖЕНИЕ)



ИСХОДНЫЕ И ВЫХОДНЫЕ ДАННЫЕ АЛГОРИТМОВ (ПРОДОЛЖЕНИЕ)

Помещение информации в переменную любого типа или в имя константы называется присваиванием.

Присваивание является первым ключевым понятием алгоритмизации. Обычно обозначается двумя подряд (без пробела) идущими символами :=

Присваивание уничтожает старое содержимое переменной или константы

Примеры присваиваний.

1. $M1 := 0,125$ (числовая переменная или константа)
2. $M[5] := -25$ (в элемент массива M с номером 5 присвоили число -25)
3. $H3 := \text{true}$ (логическая константа или переменная - истина)

4. **$S := S + 1$** (к содержимому переменной S добавили 1 и полученное значение снова поместили в переменную S , при этом старое содержимое S уничтожилось). Данный пример является **вторым ключевым** понятием алгоритмизации – **накоплением суммы**.

Вместо 1 можно использовать другую переменную или константу, или имя переменной или константы. **Классическое накопление суммы – это начальное значение S , равное 0, т.к. при сложении с нулем сумма не изменяется**

5. **$P := P * T$** (содержимое переменной P умножается на некоторое число, содержащееся в переменной T и снова помещается в переменную P . Старое содержимое P уничтожается). Данный пример является **третьим ключевым** понятием алгоритмизации – **накоплением произведения.**



Классическое накопление произведения – это начальное значение P , равное 1, т.к. при умножении на 1 произведение не меняется



ИСХОДНЫЕ И ВЫХОДНЫЕ ДАННЫЕ АЛГОРИТМОВ (ОКОНЧАНИЕ)

Исходные данные могут обрабатываться разными действиями, но в инженерных, математических или физических расчетах часто используются одинаковые действия. Такие действия называются **стандартными функциями** подобно тому, как это было в электронных таблицах, но некоторые действия могут записываться по-другому, не так, как в Excel



ЗАДАЧИ ДЛЯ АЛГОРИТМОВ

1. В трехзначном целом числе **T** выделить отдельно количество сотен, десятков, единиц и напечатать их соответственно в переменных **A, B, C**



ЗАДАЧИ ДЛЯ АЛГОРИТМОВ

2. В банк положили вклад в A у.е. на X дней на следующих условиях:

- если вклад лежит менее месяца, проценты не начисляются
- если он лежит от месяца до трех, начисляется 4% годовых
- если вклад лежит от трех месяцев до полугода, начисляется 6% годовых
- если вклад лежит свыше полугода, начисляются 10% годовых. В переменной $A1$ рассчитать сумму увеличенного вклада и напечатать ее.



ЗАДАЧИ ДЛЯ АЛГОРИТМОВ (ОКОНЧАНИЕ)

3. Имеется лист бумаги формата A4 массой 1 г. На сколько частей надо разрезать лист, чтобы получить кусочек бумаги массой молекулы целлюлозы

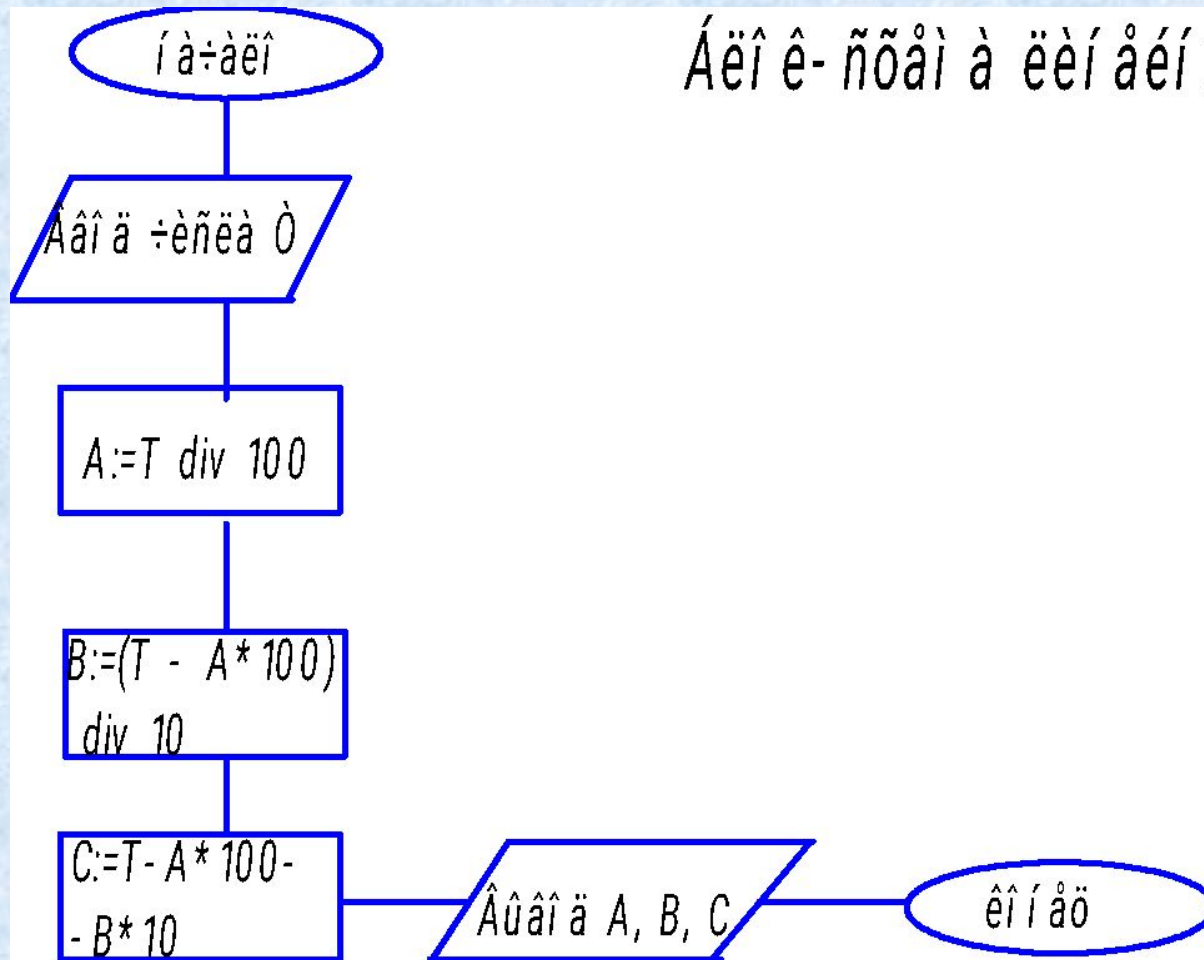
$(2.6 \cdot 10^{-24} \text{ г})$

4. С клавиатуры вводятся 10 любых вещественных чисел. Найти и напечатать среднее арифметическое

этих чисел



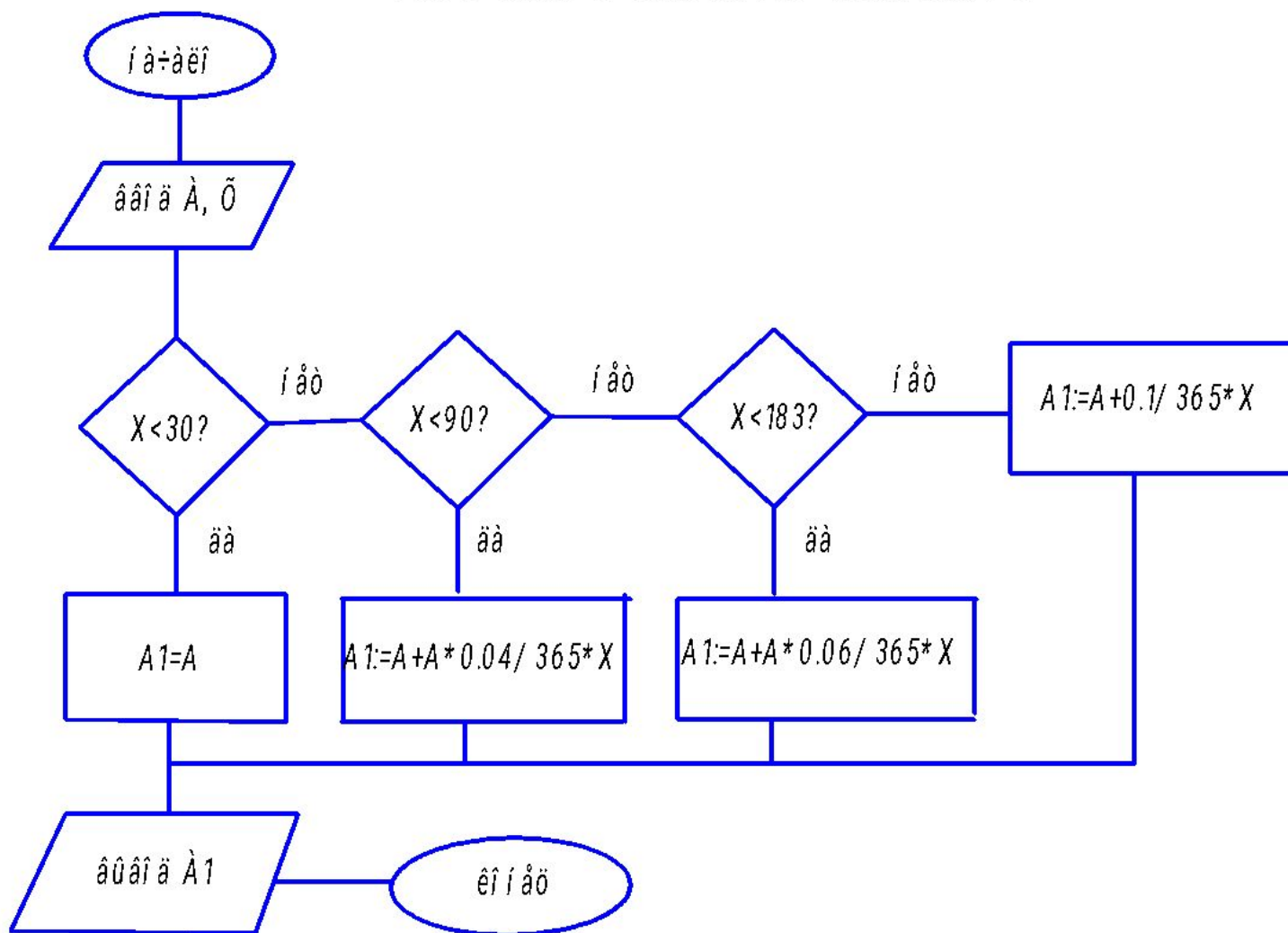
ПРИМЕРЫ АЛГОРИТМОВ



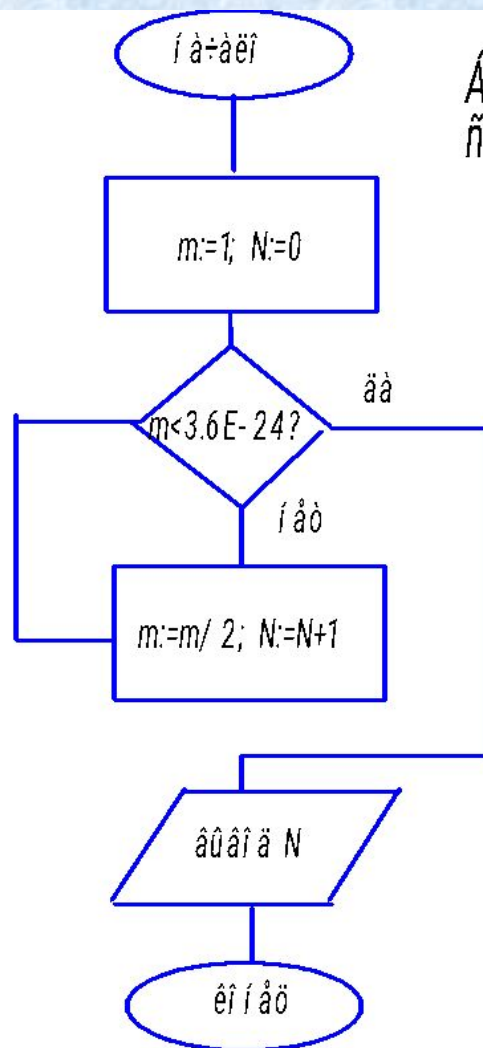
Алгоритм вычисления суммы цифр числа T

ПРИМЕРЫ АЛГОРИТМОВ (ПРОДОЛЖЕНИЕ)

Àëî ê- ñõàì à óñëî áíî ãî àëãî ðèò ì à



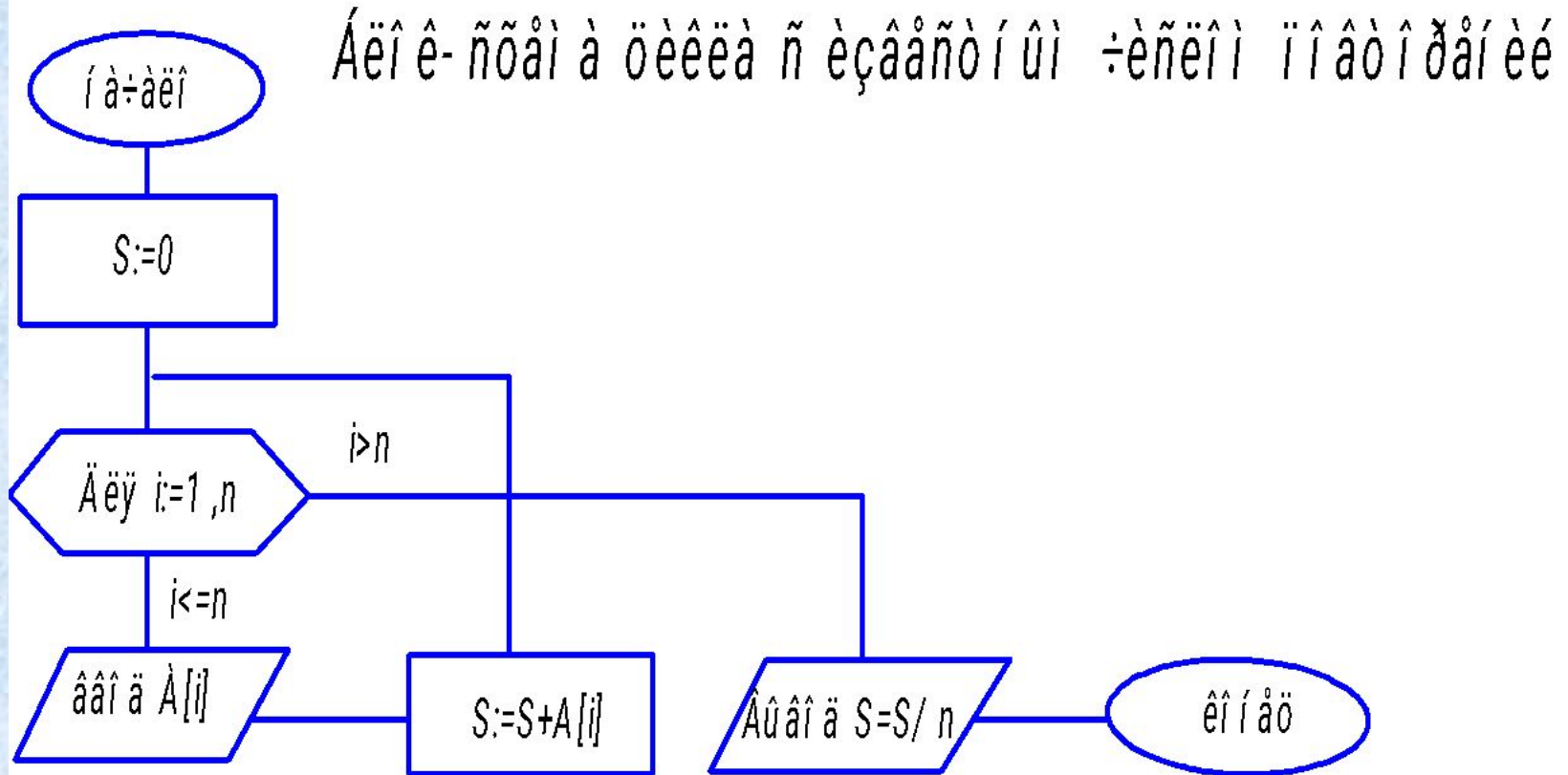
ПРИМЕРЫ АЛГОРИТМОВ (ПРОДОЛЖЕНИЕ)



Áëî ê- ñõâî à àëãî ðèò ì à öèèèà
 ñ ì ðããóñëî âèàî



ПРИМЕРЫ АЛГОРИТМОВ (ОКОНЧАНИЕ)



СПИСОК ЛИТЕРАТУРЫ

1. Ляхович В.Ф. и др. Основы информатики. Учебное пособие. – Р-Д.: Феникс - 2005
2. Каймин В.А. Информатика. – М.: Инфра-М - 2002

