

Одномерные массивы в языке программирования Паскаль.

Обучающая программа



Содержание

Введение

Одномерные массивы в Паскале

Примеры решения задач

Задания для самостоятельной
работы

Индивидуальные задания

Тестовые задания



Введение

Основной методический принцип обучения программированию –

все познается через труд,

через преодоление собственных ошибок,

через процесс решения задач.



Введение

Использование программирования позволяет:

- показать практическую значимость использования программирования для наглядного представления решения задач в различных областях жизнедеятельности человека;
- активизировать учебный процесс;
- использовать его как инструментарий индивидуального и развивающего обучения;
- содействовать развитию общей информационной культуры как одного из аспектов будущей профессиональной деятельности;
- развивать логическое и аналитическое мышление школьников.



Одномерные массивы в Паскале

Понятие «массив»

Описание одномерных массивов

Ввод одномерных массивов

Вывод одномерных массивов



Понятие «массив»

С понятием «массив» приходится сталкиваться при решении научно-технических и экономических задач обработки совокупностей большого количества значений. В общем случае **массив** – это структурированный тип данных, состоящий из фиксированного числа элементов, имеющих один и тот же тип.

Понятие «массив»

Название *регулярный тип* (или *ряды*) массивы получили за то, что в них объединены однотипные (логически однородные) элементы, упорядоченные (урегулированные) по индексам, определяющим положение каждого элемента в массиве.

Массив – это составной объект, образованный из элементов (компонент) одного и того же типа.

Понятие «массив»

Такой тип данных применяется в программировании для обозначения объектов, аналогичных **числовым последовательностям в математике**, где сразу целая группа чисел обозначается одним именем (чаще всего буквой), а для обращения к каждому отдельному числу данной последовательности используются различные **индексы** (номера элементов).

В математике это может выглядеть, например, так:

$$a_1, a_2, a_3, \dots, a_n$$

Понятие «массив»

*В программировании **массив** – это последовательность однотипных элементов, имеющих общее имя, причем каждый элемент этой последовательности определяется порядковым номером (индексом) элемента.*

Понятие «массив»

Примеры массивов:

X_1, X_2, \dots, X_n – одномерный массив,
состоящий из n элементов.

$A_0, A_1, A_2, \dots, A_{10}$ – одномерный массив,
состоящий из 11 элементов.

Массивы бывают одномерными (один индекс),
двумерными (два индекса) и т.д.

Понятие «массив»

Структура массива всегда однородна.

Массив может состоять из элементов типа *integer* , *real* или *char* , либо других однотипных элементов. Другая особенность массива состоит в том, что к любой его компоненте можно обращаться произвольным образом.

Программа может сразу получить нужный ей элемент по его порядковому номеру (индексу).

Номер элемента массива называется индексом. Индекс – это значение порядкового типа, определенного, как **тип индекса** данного массива.



Понятие «массив»

Рассмотрим массив X_1, X_2, \dots, X_n .

Здесь:

X – имя массива;

$1, 2, \dots, n$ – индексы
(порядковые номера) элементов;

X_7 – седьмой элемент массива X .

Описание одномерных массивов

Для описания массивов в программировании предварительно следует ввести соответствующий тип в разделе описания ТИПОВ.

Тип *массив* описывается следующим образом:

Type

*Имя типа = **Array** [тип индекса (ов)] **Of**
тип элементов;*

Var

Имя переменной: имя типа;

Описание одномерных массивов

Переменную типа *массив* можно описать сразу в разделе описания переменных **Var**:

Var

Имя переменной: **Array** [тип индекса (ов)] **Of** тип элементов;

Здесь

Array – служебное слово (в переводе с английского означает «массив»);

Of – служебное слово (в переводе с английского означает «из»).

Тип индекса – любой порядковый тип, кроме типов *integer*, *longint*.

Тип же самих элементов может быть любым, кроме файлового типа.

Описание одномерных массивов

Количество элементов массива называется его *размерностью*.

*Размерность массива равна:
максимальное значение индекса – n
минимальное значение индекса + 1.*

Например:

Type

```
mas = Array [1..20] of real;
```

Var

```
X: mas;
```

Массив *X* – одномерный, состоящий из двадцати элементов вещественного типа. Элементы массива хранятся в памяти компьютера последовательно друг за другом.

Ввод одномерных массивов Ввод одномерных массивов Ввод одномерных массивов в Паскале

Ввод массивов осуществляется поэлементно.

Введем одномерный массив X , состоящий из **30** элементов, то есть необходимо ввести некую последовательность элементов X_1, X_2, \dots, X_{30} .

Пусть i – индекс (порядковый номер) элемента в массиве X .

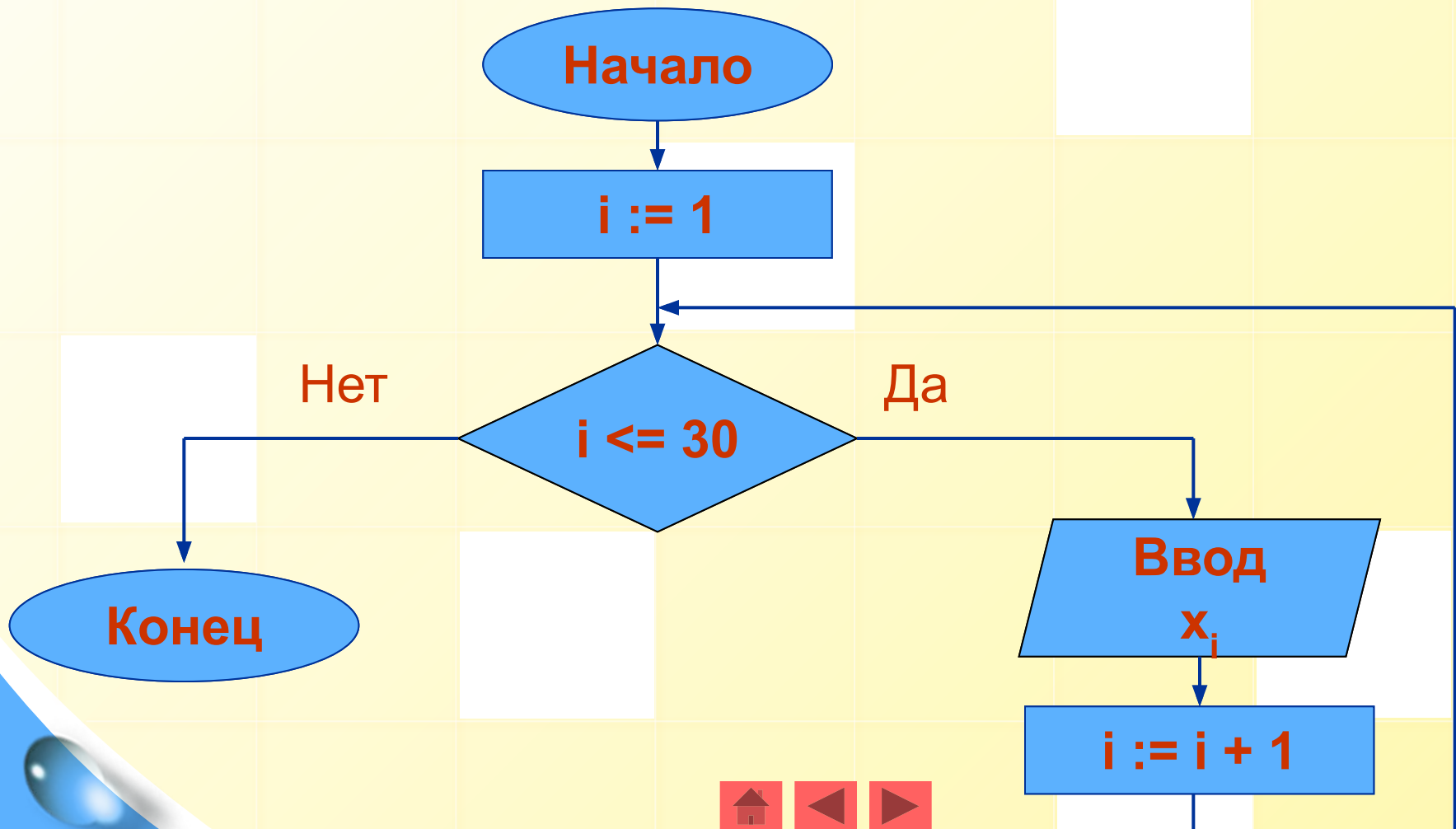
Тогда X_i – i -й элемент массива X , где $i = 1, 2, \dots, 30$.

Для ввода массива можно использовать любой цикл.



Ввод одномерных массивов

Первый вариант: ввод массива с использованием цикла с предусловием:



Ввод одномерных массивов

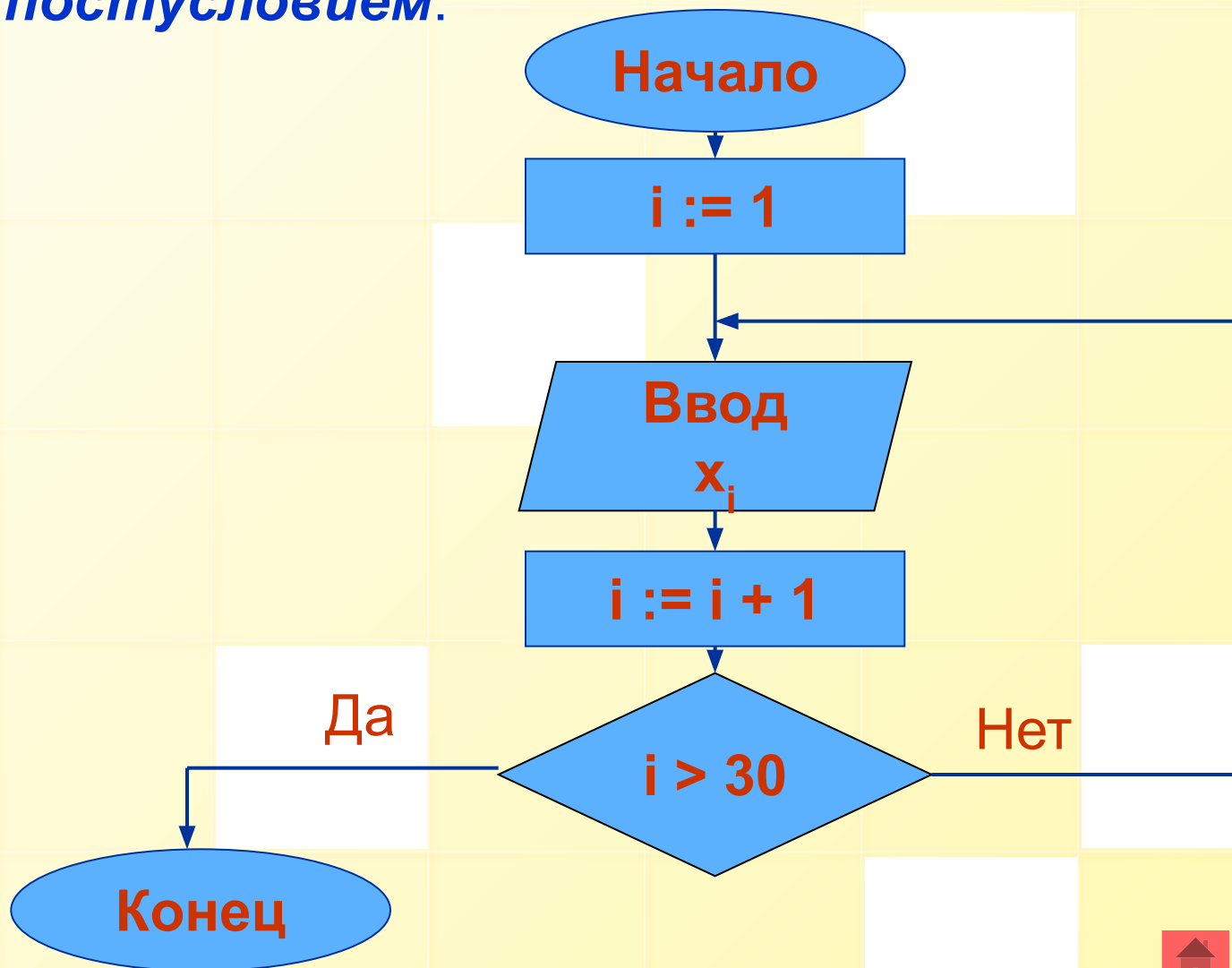
```
Program Primer_1;  
Var i: integer;  
      X: array [1..30] of Integer;  
Begin  
  i := 1;  
  While i <= 30 Do  
    Begin  
      Read (X[i]);  
      i := i + 1  
    End;  
  Readln  
End.
```

В первом варианте ввод массива X в программе осуществляется с использованием цикла с предусловием.



Ввод одномерных массивов в Паскале

Второй вариант: ввод массива с использованием цикла с постусловием:



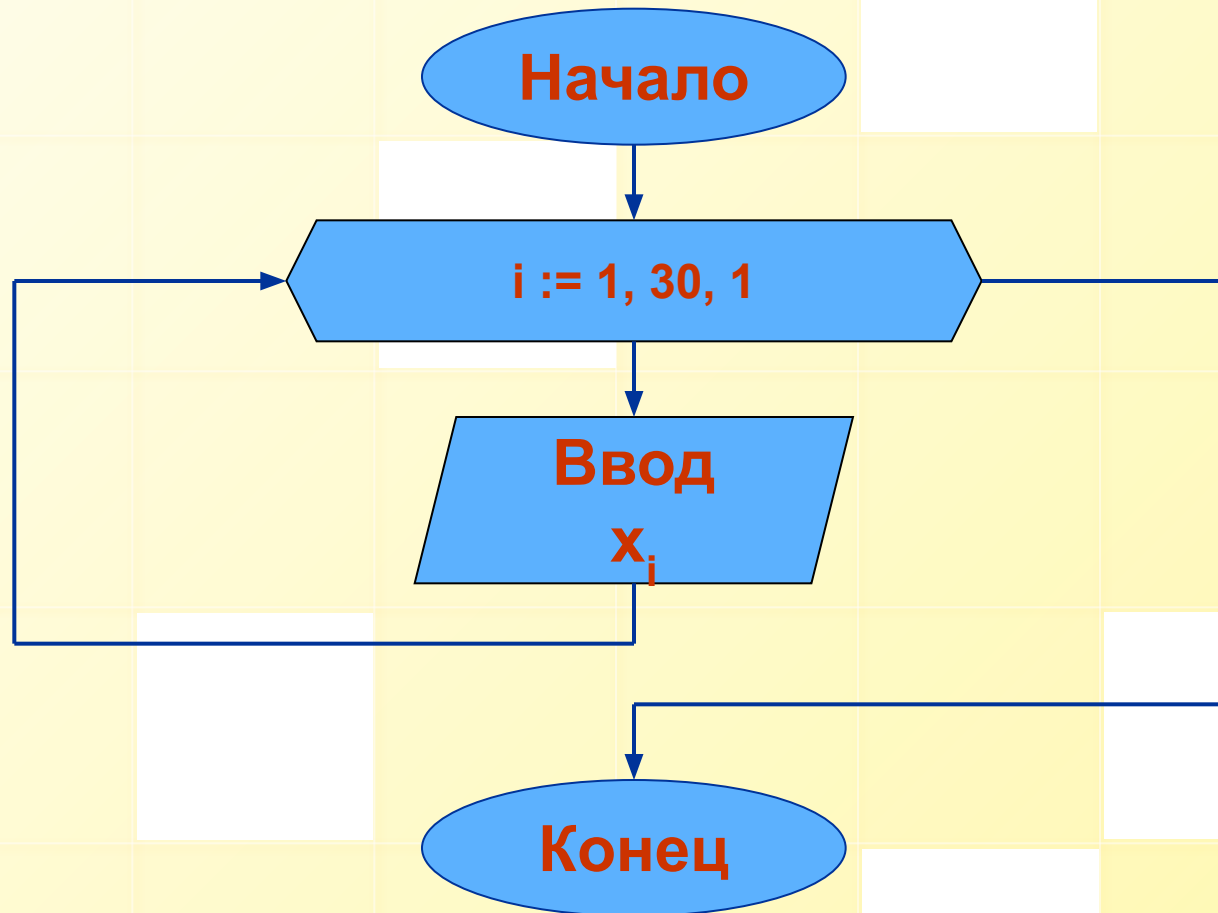
Ввод одномерных массивов

```
Program Primer_2;  
Var i: integer;  
    X: array [1..30] of Integer;  
Begin  
    i := 1;  
    Repeat  
        Read (X[i]);  
        i := i + 1  
    Until i > 30;  
    Readln  
End.
```

Массив **X** введен с использованием цикла с постусловием.

Ввод одномерных массивов

Третий вариант: ввод массива с использованием цикла с параметром:



Ввод одномерных массивов

```
Program Primer_3;  
Var i: integer;  
    X: array [1..30] of Integer;  
Begin  
    For i := 1 To 30 Do Read (X[i]);  
    Readln  
End.
```

Массив **X** вводится с помощью цикла с параметром, где в качестве параметра используется *индекс* элемента массива (i).

Вывод одномерных массивов в Паскале

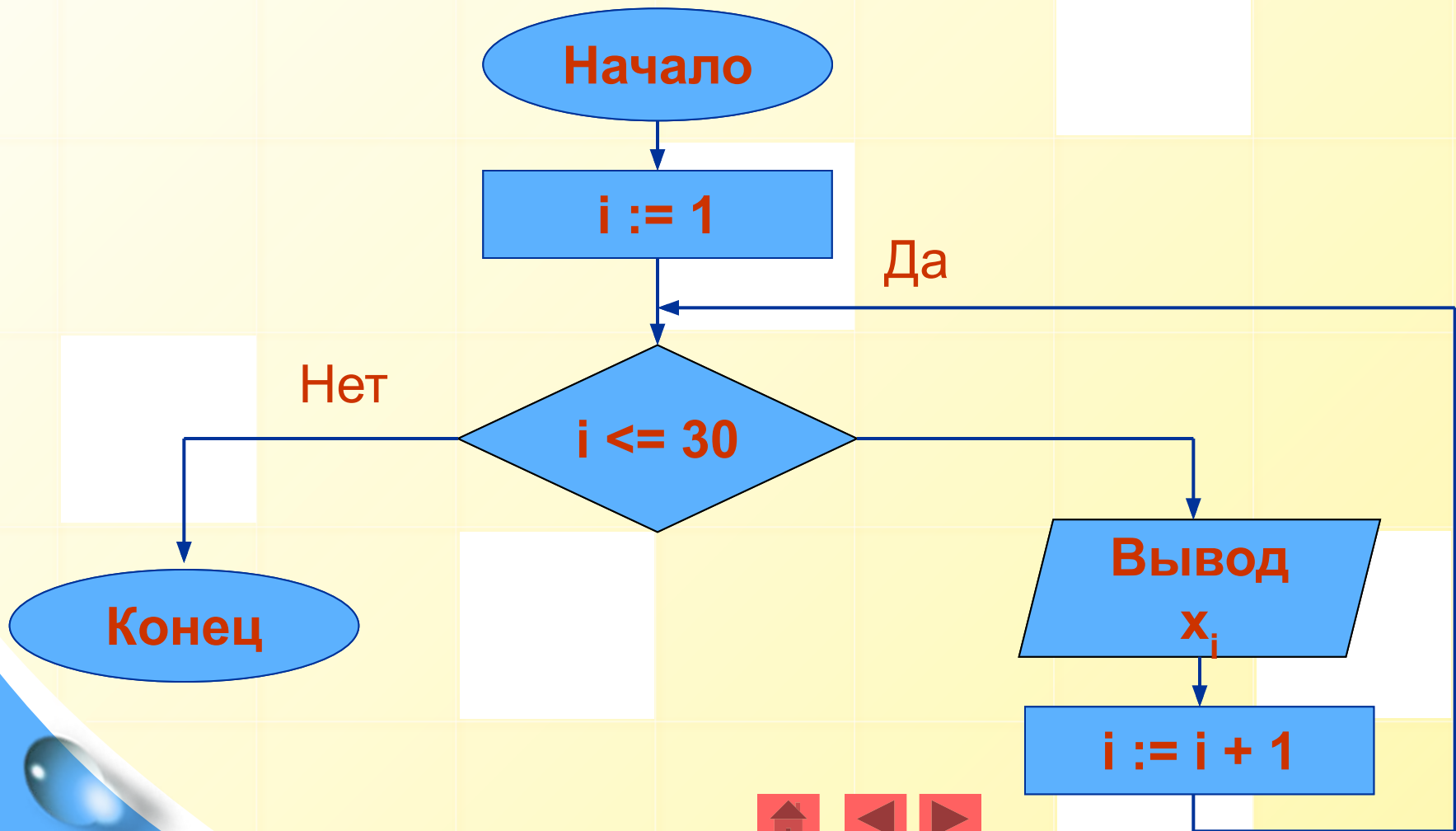
Вывод одномерного массива осуществляется также поэлементно.

Например, выведем одномерный массив X_1, X_2, \dots, X_n , состоящий из элементов вещественного типа.

Для вывода массива можно использовать любой цикл.

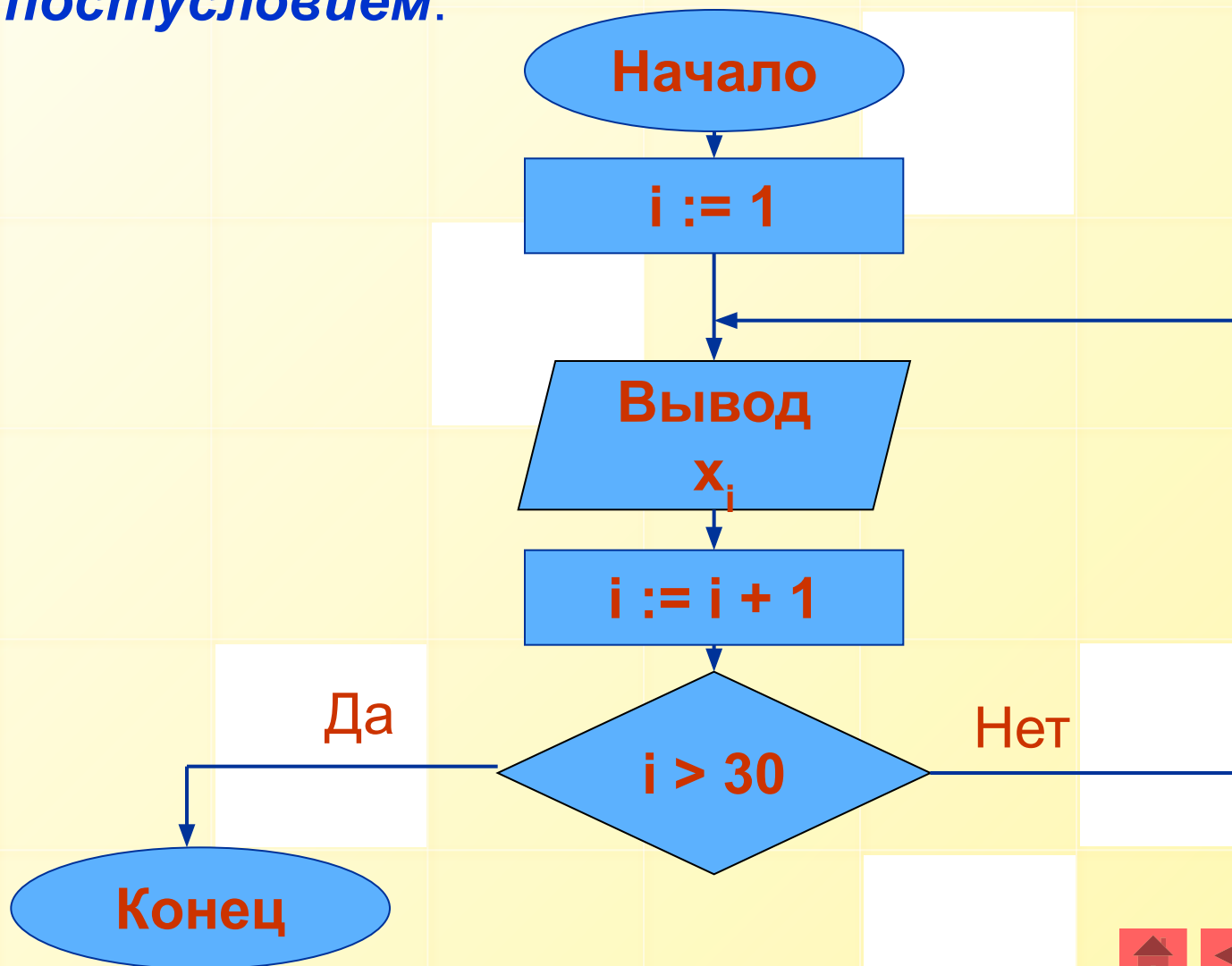
Вывод одномерных массивов

Первый вариант: вывод массива с использованием цикла с предусловием:



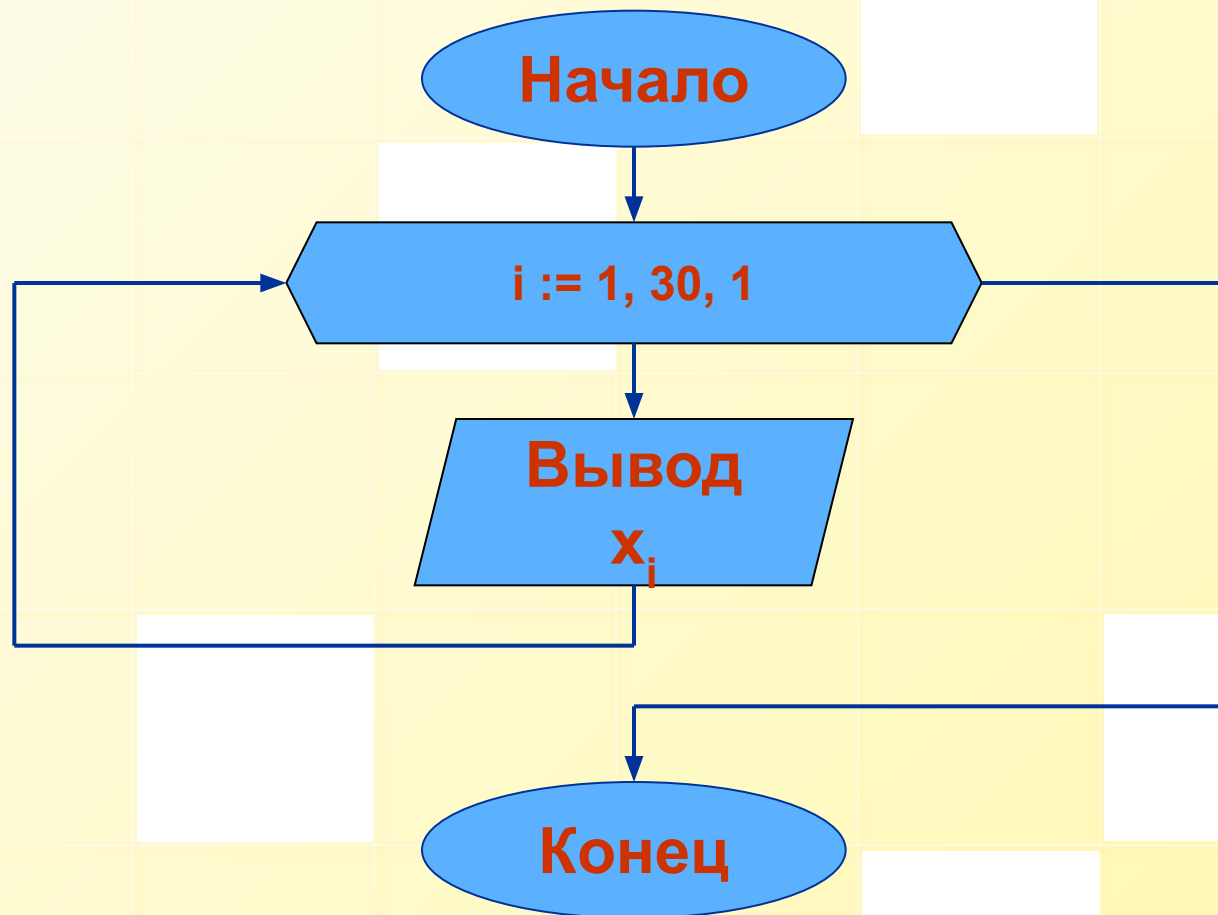
Вывод одномерных массивов

Второй вариант: вывод массива с использованием цикла с постусловием:



Вывод одномерных массивов

Третий вариант: вывод массива с использованием цикла с параметром:



Вывод одномерных массивов

В программе вместо операторов *Read* или *Readln* используются операторы *Write* или *Writeln*.

Но просто заменить одни операторы на другие здесь недостаточно.

Для того чтобы выводимые значения не сливались между собой, надо явным образом вставлять между ними *разделитель* – пробел или перевод строки.

Вывод одномерных массивов

Приведем два возможных способа вывода массива:

1) **For** $i := 1$ **To** n **Do** Write ($X[i]$, ' ')

2) **For** $i := 1$ **To** n **Do** Writeln ($x[i]$)

Вывод одномерных массивов

На первый взгляд второй способ может показаться более простым и удобным, но это далеко не всегда так.

Результат работы такой программы зачастую неудобно, а то и просто невозможно анализировать.

Ведь *каждый элемент массива будет располагаться в отдельной строке*, следовательно, мы не сможем увидеть более 25 элементов одновременно.

Вывод одномерных массивов

Кроме того, очень часто массив требуется распечатать дважды, чтобы сравнить состояние массива до обработки и результат его обработки.

В этом случае сравнение состояний массива гораздо удобнее проводить, если они распечатаны в двух соседних строках, а элементы выровнены по столбцам, то есть к варианту 1 должна быть добавлена еще и **форматная печать** (указано количество позиций, которое должно отводиться на печать одного элемента).



Вывод одномерных массивов

Программа вывода одномерного массива с помощью цикла с параметром:

```
Program Primer_4;  
Const n = 30;  
Var i: Integer;  
      X: Array [1..n] Of Real;  
Begin  
  For i:= 1 To n Do Write (X[i] : 6 : 2, ' ');  
  Writeln; {курсор переводит на новую строку}  
  Readln  
End.
```

Примеры решения задач

- Вычисление суммы и произведения элементов массива, удовлетворяющих заданному условию
- Нахождение количества элементов, удовлетворяющих заданному условию
- Нахождение номеров элементов, обладающих заданным свойством
- Поиск нужного элемента в массиве
- Изменение значений некоторых элементов массива
- Формирование нового массива
- Перестановка элементов массива
- Сортировка массива

Вычисление суммы и произведения
элементов массива,
удовлетворяющих заданному условию

Задача 1

Дан целочисленный одномерный массив, состоящий из n элементов. Найти сумму и произведение нечетных элементов, кратных 3.

Задача 1.

Введем обозначения:

n – количество элементов в массиве;

A – имя массива;

i – индекс элемента массива;

A_i – i -й элемент массива A ;

s – сумма нечетных элементов массива, кратных 3;

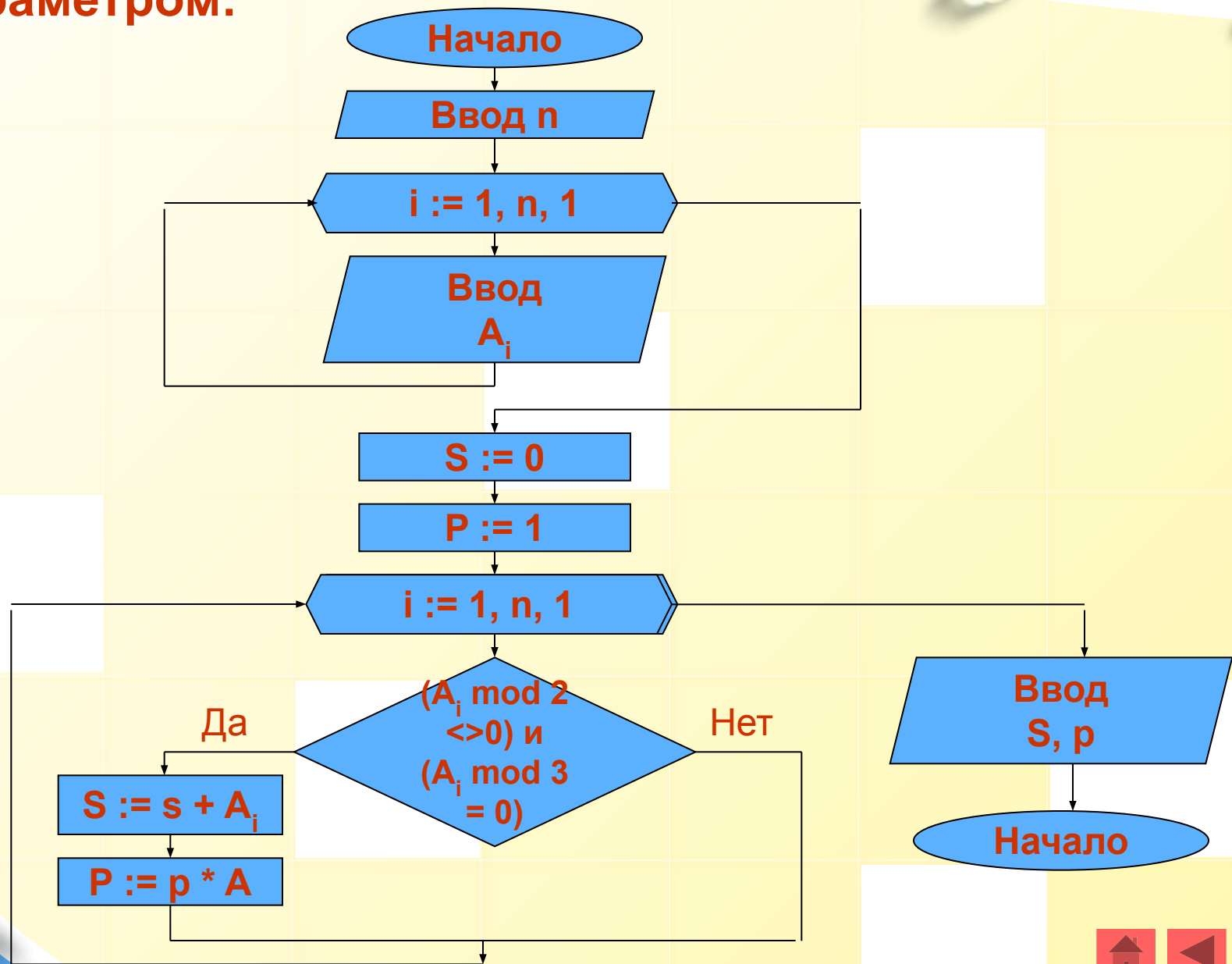
p – произведение нечетных элементов массива, кратных 3.

Входные данные: n, A .

Выходные данные: s, p .



Первый способ. Для решения используется цикл с параметром:



Задача 1.

*Дан целочисленный одномерный массив, состоящий из n элементов.
Найти сумму и произведение нечетных элементов, кратных 3.*

```
Program Primer1_1;  
Var A: Array[1..20] Of Integer;  
    i, n, s, p: Integer;  
Begin  
    Write ('n='); Readln (n);  
    For i:=1 To n Do Readln (A[i]); {ввод массива}  
    s:= 0; p:=1;  
    For i:=1 To n Do {обработка массива}  
        If (A[i] mod 2 <>0) and (A[i] mod 3 = 0) Then  
            Begin  
                s:=s+A[i];  
                p:= p*A[i]  
            End;  
    Writeln ('s=', s, 'p=', p);  
    Readln  
End.
```



Задача 1.

Второй способ. Для решения используется цикл с предусловием:

```
Program Primer1_2;  
Var A: Array[1..20] Of Integer;  
    i, n, s, p: Integer;  
Begin  
    Write ('n='); Readln (n);  
    i:=1;  
    While i <= n Do  
        Begin  
            Readln (A[i]); i:= i + 1 End;  
            s:= 0; p:=1; i:=1;  
            While i<=n Do Begin  
                If (A[i] mod 2 <>0) and (A[i] mod 3 = 0)  
                    Then  
                        Begin  
                            s:=s+A[i]; p:= p*A[i]  
                        End;  
                i:= i + 1 End;  
            Writeln ('s=', s, 'p=', p);  
            Readln  
        End.
```



Нахождение количества элементов,
удовлетворяющих заданному
условию

Задача 2

Дан массив целых чисел. Найти количество тех элементов, значения которых положительны и не превосходят заданного натурального числа A .

Задача 2.

Введем обозначения:

n – количество элементов в массиве;

X – имя массива;

i – индекс элемента массива;

X_i – i -й элемент массива X ;

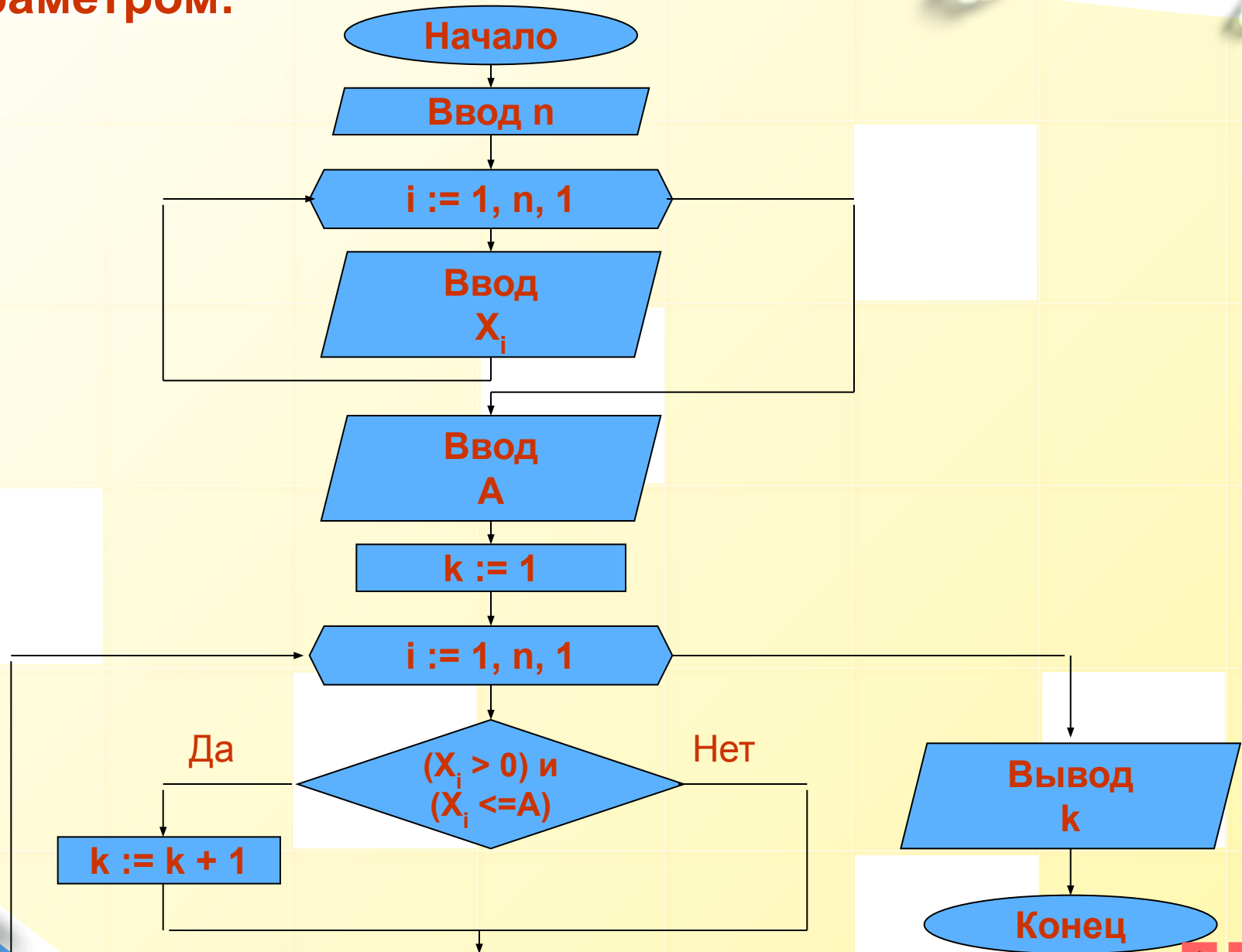
A – заданное число;

k – количество элементов, значения которых положительны и не превосходят заданного числа A .

Входные данные: n, X, A .

Выходные данные: k .

Первый способ. Для решения используется цикл с параметром:



Задача 2.

Дан массив целых чисел. Найти количество тех элементов, значения которых положительны и не превосходят заданного натурального числа A.

```
Program Primer2_1;  
Var X: Array[1..20] Of Integer;  
    i, n, k, A: Integer;  
Begin  
    Write ('n='); Readln (n);  
    For i:=1 To n Do Readln (X[i]); {ввод массива}  
    Write ('A='); Readln (A);  
    k:= 0;  
    For i:=1 To n Do {обработка массива}  
        If (X[i] >0) and (X[i] <= A) Then  
            k:=k + 1;  
    Writeln ('k=', k);  
    Readln  
End.
```

Задача 2.

Второй способ. Для решения используется *цикл с постусловием*:

```
Program Primer2_2;  
Var X: Array [1..20] Of Integer;  
    i, n, k, A: Integer;  
Begin  
    Write ('n='); Readln (n); i:=1;  
    Repeat  
        Read (X[i]); i := i + 1  
    Until i > n;  
    Write ('A='); Readln (A);  
    k:= 0; i:=1;  
    Repeat  
    If (X[i] >0) and (X[i] <= A) Then  
        k:=k + 1; i := i + 1  
    Until i > n;  
    Writeln ('k=', k);  
    Readln  
End.
```

Нахождение номеров элементов, обладающих заданным свойством

Задача 3

**Найти номера четных элементов
массива, стоящих на нечетных
местах.**

Задача 3

Введем обозначения:

n – количество элементов в массиве;

X – имя массива;

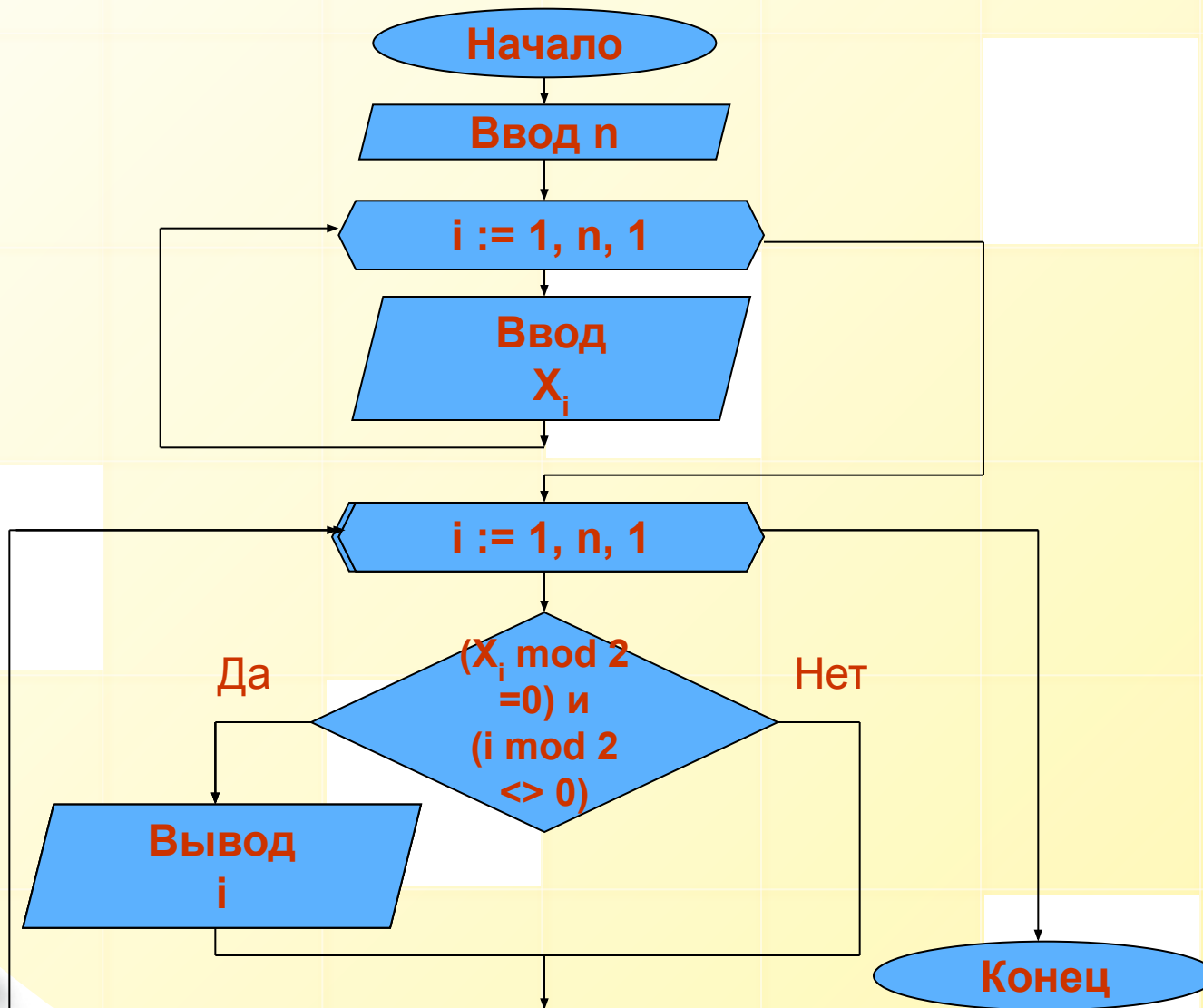
i – индекс элемента массива;

X_i – i -й элемент массива X .

Входные данные: n, X .

Выходные данные: i .

Задача 3



Задача 3

```
Program Primer3_1;  
Var X: Array[1..20] Of Integer;  
    i, n: Integer;  
Begin  
    Write ('n='); Readln (n);  
    For i:=1 To n Do Readln (X[i]); {ввод массива}  
    For i:=1 To n Do {обработка массива}  
        If (X[i] mod 2 = 0) and (i mod 2 <> 0) Then  
            Write (i:5);  
    Readln  
End.
```



Задача 4

**Найти номер последнего отрицательного
элемента массива.**



Задача 4

Введем обозначения:

n – количество элементов в массиве;

A – имя массива;

i – индекс элемента массива;

A_i – i -й элемент массива A ;

m – номер последнего отрицательного элемента массива.

Входные данные: n, A .

Выходные данные: m .

Задача 4

```
Program Primer3_2;  
Const n=10;  
Var A: Array[1..n] Of Integer;  
    i, m: Integer;  
Begin  
    For i:=1 To n Do Readln (A[i]); {ввод массива}  
    m := 0; i:=n;  
    While (i >= 1) and (A[i] >=0) Do  
        i:=i-1;  
        m:=i;  
    Writeln ('m=', m);  
    Readln  
End.
```

Поиск нужного элемента в массиве

Задача 5

Дан одномерный массив X_1, X_2, \dots, X_n .
Найти минимальный элемент и его
порядковый номер среди элементов,
расположенных на четных местах.

Задача 5

Введем обозначения:

n – количество элементов в массиве;

X – имя массива;

i – индекс элемента массива;

X_i – i -й элемент массива X ;

min – значение минимального элемента массива;

k – порядковый номер минимального элемента среди элементов, расположенных на четных местах.

Входные данные: n, X .

Выходные данные: min, k .



Задача 5

Первый способ

```
Program Primer4_1;  
Var X: Array[1..30] Of Real;  
    i, n, k: Integer; min: Real;  
Begin  
    Write ('n='); Readln (n); i:=1;  
    While i <= n Do Begin    {ввод массива}  
        Readln (X[i]); i:= i + 1  End;  
        min:=X[2]; k:=2; i:=4;  
    While i <= n Do Begin  
        If X[i] < min Then Begin min:= X[i]; k:= i  End;  
        i:= i + 2 End;  
    Writeln ('min=', min:6:1, ' k=', k);  
End.
```

Задача 5

Второй способ

```
Program Primer4_2;  
Var X: Array[1..30] Of Real;  
    i, n, k: Integer; min: Real;  
Begin  
    Write ('n='); Readln (n); i:=1;  
    Repeat  
        Read (X[i]); i := i + 1  
    Until i > n;  
    min:=X[2]; k:=2; i:=4;  
    Repeat  
    If X[i] < min Then Begin  
        min:= X[i]; k:= i End; i:= i + 2  
    Until i > n;  
    Writeln ('min=', min:6:1, ' k=', k)  
End.
```

Задача 6

Дан целочисленный одномерный массив. Есть ли в нем отрицательный элемент?

Задача 6

Введем обозначения:

n – количество элементов в массиве;

X – имя массива;

i – индекс элемента массива;

X_i – i -й элемент массива X ;

k – количество отрицательных элементов в массиве;

m – признак наличия или отсутствия отрицательного элемента в массиве X .

Входные данные: n, X .

**Выходные данные:
сообщение 'yes' или 'no'.**



Задача 6

Первый способ

```
Program Primer4_3;  
Const n=10;  
Var X: Array[1..n] Of Integer;  
    i, k: Integer;  
Begin  
    For i:=1 To n Do Readln (X[i]); {ввод массива}  
    k:= 0;  
    For i:=1 To n Do  
        If X[i] < 0 Then k:= k + 1;  
        If k <> 0 Then Writeln ('yes')  
            Else Writeln ('no');  
    Readln  
End.
```


Задача 6

Второй способ

```
Program Primer4_4;  
Const n=10;  
Var X: Array[1..n] Of Integer;  
    i: Integer; m: Boolean;  
Begin  
    For i:=1 To n Do Readln (X[i]); {ввод массива}  
    m := false; i:=1;  
    While (i <= n) and not m Do  
        Begin  
            If X[i] < 0 Then m:= true; i:=i+1  
        End;  
    If m Then Writeln ('yes')  
        Else Writeln ('no')  
End.
```

Изменение значений некоторых элементов массива

Задача 7

Дан одномерный массив, состоящий из n элементов. Если очередной элемент массива четный, то прибавить к нему первый, если нечетный – прибавить последний. Первый и последний элементы не изменять.

Задача 7

Введем обозначения:

n – количество элементов в массиве;

A – имя массива;

i – индекс элемента массива;

A_i – i -й элемент массива A .

Входные данные: n, A .

Выходные данные: A .



Задача 7

```
Program Primer5_1;  
Const n = 30;  
Var A: Array[1..n] Of Integer;  
    i: Integer;  
Begin  
    For i:=1 To n Do Readln (A[i]); {ввод массива}  
    For i:=2 To n-1 Do  
        If A[i] mod 2 = 0 Then A[i]:= A[i] + A[1]  
            Else A[i]:= A[i] + A[n];  
    For i:= 1 To n Do Write (X[i] : 5);  
        WriteLn;  
        ReadLn  
End.
```



Задача 8

Дан одномерный массив A_1, A_2, \dots, A_n .
Элементы, не превышающие 15,
заменить средним арифметическим всех
элементов массива.

Задача 8

Введем обозначения:

n – количество элементов в массиве;

A – имя массива;

i – индекс элемента массива;

A_i – i -й элемент массива A ;

s – сумма элементов массива A ;

sr – среднее арифметическое всех элементов массива.

Входные данные: n, A .

Выходные данные: A .



Задача 8

```
Program Primer5_2;  
Const n = 30;  
Var A: Array[1..n] Of Real;  
    i: Integer; s, sr: Real;  
Begin  
    For i:=1 To n Do Readln (A[i]); {ВВОД МАССИВА}  
    s:= 0;  
    For i:=1 To n Do  
        s:=s+A[i];  
        sr:= s/n;  
    For i:=1 To n Do  
        If A[i]<=15 Then A[i]:= sr;  
    For i:=1 To n Do Write (A[i]:5);  
        Writeln;  
    Readln  
End.
```

Формирование нового массива

Задача 9.

Сформировать целочисленный массив X_1, X_2, \dots, X_n , элементы которого: 2, 4, 8, 16, ...

Задача 9

Введем обозначения:

n – количество элементов в массиве;

X – имя массива;

i – индекс элемента массива;

X_i – i -й элемент массива X .

Входные данные: n .

Выходные данные: X .



Задача 9

По исходным данным получаем следующее правило формирования массива: $X_1=2$; $X_i = X_{i-1} * 2$.

```
Program Primer6_1;  
Const n = 20;  
Var X: Array[1..n] Of Integer;  
    i: Integer;  
Begin  
    X[1]:=2; {начало формирования массива}  
    For i:=2 To n Do  
        X[i]:=X[i-1]*2; {конец формирования массива}  
    For i:=1 To n Do Write (X[i]:5); {вывод массива}  
        WriteLn;  
    ReadLn  
End.
```

Задача 10

Сформировать одномерный массив так, что первые два элемента заданы, а остальные элементы с нечетными номерами больше предыдущего на величину первого, а с четными – на величину второго.

Задача 10

Введем обозначения:

n – количество элементов в массиве;

X – имя массива;

i – индекс элемента массива;

X_i – i -й элемент массива X ;

a, b – соответственно значения первого и второго элементов массива.

Входные данные: n, a, b .

Выходные данные: X .

Задача 10

```
Program Primer6_2;  
Const n = 20;  
Var X: Array[1..n] Of Integer;  
    i, a, b: Integer;  
Begin  
    Write ('a='); Readln (a);  
    Write ('b='); Readln (b);  
    X[1]:=a; X[2]:=b; {начало формирования массива}  
    For i:=3 To n Do  
        If i mod 2 <> 0 Then X[i]:= X[i-1] + X[1]  
            Else X[i]:= X[i-1] + X[2];  
        {конец формирования массива}  
    For i:=1 To n Do Write (X[i]:5); {вывод массива}  
    Readln  
End.
```



Задача 11

Дан одномерный массив X_1, X_2, \dots, X_n . Из четных элементов этого массива сформировать новый одномерный массив.

Задача 11

Введем обозначения:

n – количество элементов в массиве;

X – имя заданного массива;

i – индекс элемента массива;

X_i – i -й элемент массива X ;

Y – имя формируемого массива;

k – порядковый номер элемента в массиве Y ;

Y_k – k -й элемент массива Y .

Входные данные: n , X .

Выходные данные: Y .



Задача 11

```
Program Primer6_3;  
Const n = 20;  
Var X, Y: Array[1..n] Of Integer;  
    i, k: Integer;  
Begin  
    For i:=1 To n Do Read (X[i]); {ввод массива X}  
    k:=0;  
    For i:=1 To n Do  
        If X[i] mod 2 = 0 Then  
            Begin  
                k:=k + 1;  
                Y[k] := X[i]  
            End;  
    For i:=1 To k Do Writeln (Y[i]:5); {вывод массива Y}  
    Readln  
End.
```


Перестановка элементов массива

Задача 12

Дан одномерный массив X_1, X_2, \dots, X_n .
Поменять местами значения двух
элементов массива с номерами k_1 и k_2 ,
где k_1 и k_2 - вводятся с клавиатуры.

Задача 12

Введем обозначения:

n – количество элементов в массиве;

X – имя заданного массива;

i – индекс элемента массива;

X_i – i -й элемент массива X ;

$k1, k2$ – порядковые номера элементов, значения которых нужно поменять местами;

b – дополнительная переменная, используемая для перестановки двух элементов массива.

Входные данные: n, X .

Выходные данные: Y .

Задача 12

```
Program Primer7_1;  
Const n = 20;  
Var X: Array[1..n] Of Integer;  
    i, k1, k2, b: Integer;  
Begin  
    For i:=1 To n Do Read (X[i]); {ввод массива}  
    Write ('k1='); Readln (k1);  
    Write ('k2='); Readln (k2);  
    b:= X[k1];  
    X[k1]:= X[k2];  
    X[k2]:= b;  
    For i:=1 To n Do Writeln (X[i]:5); {вывод массива X}  
    Readln  
End.
```



Задача 13

**Переставить все элементы
одномерного массива X
в обратном порядке.**



Задача 13

Введем обозначения:

n – количество элементов в массиве;

X – имя заданного массива;

i – индекс элемента массива;

X_i – i -й элемент массива X ;

b – дополнительная переменная.

Входные данные: n , X .

Выходные данные: Y .



Задача 13

```
Program Primer7_2;  
Const n = 20;  
Var X: Array[1..n] Of Integer;  
    i, b: Integer;  
Begin  
    For i:=1 To n Do Read (X[i]); {ввод массива}  
    For i:=1 To n div 2 Do  
        Begin  
            b:= X[i];  
            X[i]:= X[n+1-i];  
            X[n+1-i]:= b  
        End;  
    For i:=1 To n Do Write (X[i]:5); {вывод массива X}  
        WriteLn;  
        ReadLn  
End.
```

Сортировка массива (сортировка «пузырьком»)

Задача 14

**Отсортировать целочисленный
одномерный массив по
возрастанию.**

Сортировка «пузырьком»

Суть метода состоит в том, что последовательно сравниваются все пары соседних элементов массива, и если значения элементов в паре стоят в неправильном порядке (правый меньше левого), то они меняются местами.

В результате одного такого прохода по массиву самый большой элемент обязательно окажется на последней позиции массива. При этом от своей позиции он, путем постепенных обменов, дойдет до последнего места.
«Всплывает» как пузырек.

Сортировка «пузырьком»

Но массив при этом вовсе не обязательно окажется упорядоченным. Гарантированно **на нужном месте при этом окажется только самый большой элемент**. Поэтому эту процедуру нужно повторить. Но только теперь можно не рассматривать самую последнюю пару.

В результате предпоследний элемент тоже окажется на нужном месте.

Значит, **всю процедуру нужно повторить столько раз, сколько элементов нужно поставить на требуемое место, то есть $n - 1$** (оставшийся элемент окажется при этом на своем месте сам).

Задача 14

Введем обозначения:

n – количество элементов в массиве;

X – имя заданного массива;

i – индекс элемента массива;

X_i – i -й элемент массива X ;

k – номер просмотра, изменяется от 1 до $n-1$;

w – промежуточная переменная для перестановки местами элементов массива.

Входные данные: n , X .

Выходные данные: X .



Задача 14

```
Program Primer8_1;
Const n = 100;
Var X: Array[1..n] Of Integer;
    i, k, w: Integer;
Begin
  For i:=1 To n Do Read (X[i]); {ввод массива}
  For k:=n - 1 Downto 1 Do {цикл по номеру просмотра}
    For i:=1 To k Do
      If X[i] > X[i + 1] Then {перестановка элементов}
        Begin
          w:= X[i];
          X[i]:= X[i + 1];
          X[i + 1]:= w
        End;
    For i:=1 To n Do Writeln (X[i]:5); {вывод массива X}
  Readln
End.
```



Задания для самостоятельной работы

1. Дан целочисленный одномерный массив, состоящий из n элементов, и натуральные числа k, m ($k < m < n$). Вычислить среднее арифметическое элементов массива с номерами от k до m включительно.
2. Дан целочисленный одномерный массив, состоящий из n элементов. Подсчитать количество его положительных, отрицательных и нулевых элементов.
3. Дан одномерный массив X_1, X_2, \dots, X_n . Найти максимальный элемент и его порядковый номер.
4. Дан одномерный массив X_1, X_2, \dots, X_n . Есть ли в массиве положительные элементы, кратные k (k вводить с клавиатуры)?
5. Дан одномерный массив X_1, X_2, \dots, X_n . Найти все элементы, кратные 3 или 5. Сколько их?

Задания для самостоятельной работы

6. Заменить отрицательные элементы массива на их абсолютные величины.
7. Сформировать одномерный массив, содержащий n первых положительных нечетных чисел: 1, 3, 5, ...
8. Дано целое число a и массив X_1, X_2, \dots, X_n . Из элементов, меньших a , сформировать новый одномерный массив.
9. Дан одномерный массив A_1, A_2, \dots, A_n . Все элементы уменьшить на величину квадрата наименьшего элемента.
10. Поменять местами первый и максимальный элементы массива.
11. Проверить, что массив упорядочен строго по убыванию (каждый последующий элемент строго меньше предыдущего).

Индивидуальные задания

Вариант № 1

1. Найти сумму положительных элементов в массиве.
2. Дан одномерный массив A_1, A_2, \dots, A_n . Наибольший элемент этого массива заменить средним арифметическим отрицательных элементов.

Вариант № 2

1. Найти номера элементов массива, значения которых кратны 3 и 5.
2. Сформировать массив: $x_1 = 2, x_2 = 3, x_i =$, где $i=3, 4, \dots, n$.

Вариант № 3

1. Найти сумму всех четных элементов массива, стоящих на четных местах, то есть имеющих четные номера.
2. Есть ли в данном массиве элементы, равные заданному числу b ?

Индивидуальные задания

Вариант № 4

1. Найти номера элементов массива, значения которых кратны 4 или 6.
2. Дан одномерный массив X_1, X_2, \dots, X_n . Заменить каждый двухзначный элемент на вторую цифру в его записи.

Вариант № 5

1. Найти сумму и произведение элементов массива, больших данного числа b (b вводится с клавиатуры).
2. Есть ли в данном массиве элементы равные заданному числу? Если есть, то вывести номер одного из них.



Индивидуальные задания

Вариант № 6

1. Найти количество элементов массива, значения которых больше заданного числа B и кратны 5.
2. Сформировать одномерный массив, первые два элемента которого есть заданные числа, а остальные элементы с нечетными номерами в два раза больше предыдущего, а с четными – на 3 меньше второго.

Вариант №7

1. Найти сумму и произведение положительных элементов массива, значения которых меньше 10.
2. Дан одномерный массив A_1, A_2, \dots, A_n . Все четные элементы массива возвести в квадрат, а нечетные удвоить.

Индивидуальные задания

Вариант №8

1. Найти номера тех элементов массива, значения которых больше заданного числа C .
2. Из элементов массива A сформировать элементы массива B по правилу: $B[i] := A[1] + A[2] + \dots + A[i]$.

Вариант №9

1. Найти произведение элементов массива, имеющих нечетное значение.
2. Дан одномерный массив X_1, X_2, \dots, X_n . Все ли элементы массива больше 3?

Вариант №10

1. Найти количество тех элементов массива, значения которых положительны и не превосходят заданного числа D .
2. Дан одномерный массив A_1, A_2, \dots, A_n . Заменить все элементы, кратные 3, на третий элемент массива.

Индивидуальные задания

Вариант №11

1. Найти сумму элементов массива, имеющих нечетные индексы.
2. Определить, есть ли в данном массиве положительные элементы, кратные k (k вводится с клавиатуры).

Вариант №12

1. Найти количество нечетных элементов в массиве.
2. Из элементов массива A сформировать массив B той же размерности по правилу: если номер четный, то $B[i] := i * A[i]$, если нечетный, то $B[i] := A[i]$.

Индивидуальные задания

Вариант №13

1. Найти сумму и произведение первых пяти элементов массива.
2. Дан одномерный массив X_1, X_2, \dots, X_n . Из положительных элементов массива вычесть элемент с номером k_1 , а отрицательные увеличить на значение элемента с номером k_2 , нулевые элементы оставить без изменения.

Вариант №14

1. Найти сумму элементов массива, принадлежащих промежутку от x до y (x и y вводятся с клавиатуры).
2. Дан одномерный массив A_1, A_2, \dots, A_n . Верно ли, что произведение первого и последнего элементов больше, чем сумма нечетных элементов массива.

Индивидуальные задания

Вариант №15

1. Дан одномерный массив X_1, X_2, \dots, X_n . Найти количество и номера таких элементов массива, которые больше своего соседа справа.
2. Из элементов массива A сформировать массив B той же размерности по правилу: если номер четный, то $B[i] := A[i] * A[i]$, если нечетный, то $B[i] := A[i] \text{ Div } i$.

Вариант №16

1. Найти удвоенную сумму положительных элементов массива.
2. Дан одномерный массив A_1, A_2, \dots, A_n . Все элементы с нечетными номерами разделить нацело на первый элемент.

Тестовые задания

Кликнуть мышью по гиперссылке персонального сайта учителя информатики Камаловой Н.А.

<http://infoworld.lensch5.edusite.ru/p24aa1.html>

