Итерационные циклы
Работа с массивами

Чернецов Андрей Михайлович,

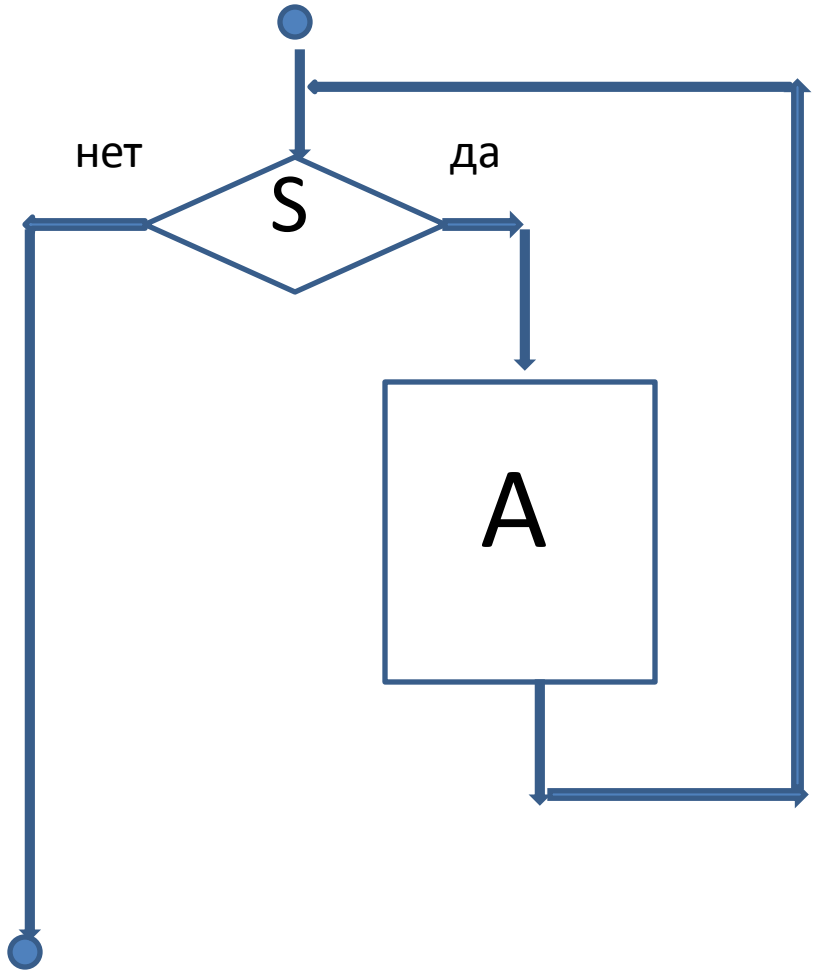
К.т.н. доцент каф. ПМИИ

Список литературы по курсу

- 1. Князев А.В. Основы языка С++. Учебное пособие. М.: Издательство МЭИ, 2013 – 80 с. ISBN 978-5-7046-1425-8.
- 2. Князев А.В. Работа со сложными структурами данных на языке С++. Учебное пособие. М.: Издательство МЭИ, 2015 – 48 с. ISBN 978-5-7046-1658-0
- 3. Программирование. Сборник задач. Учебное пособие. Санкт-Петербург: Лань, 2019 – 140 с. ISBN 978-5-8114-3857-0

URL: https://e.lanbook.com/book/121485

Оператор цикла с предусловием (**while**)

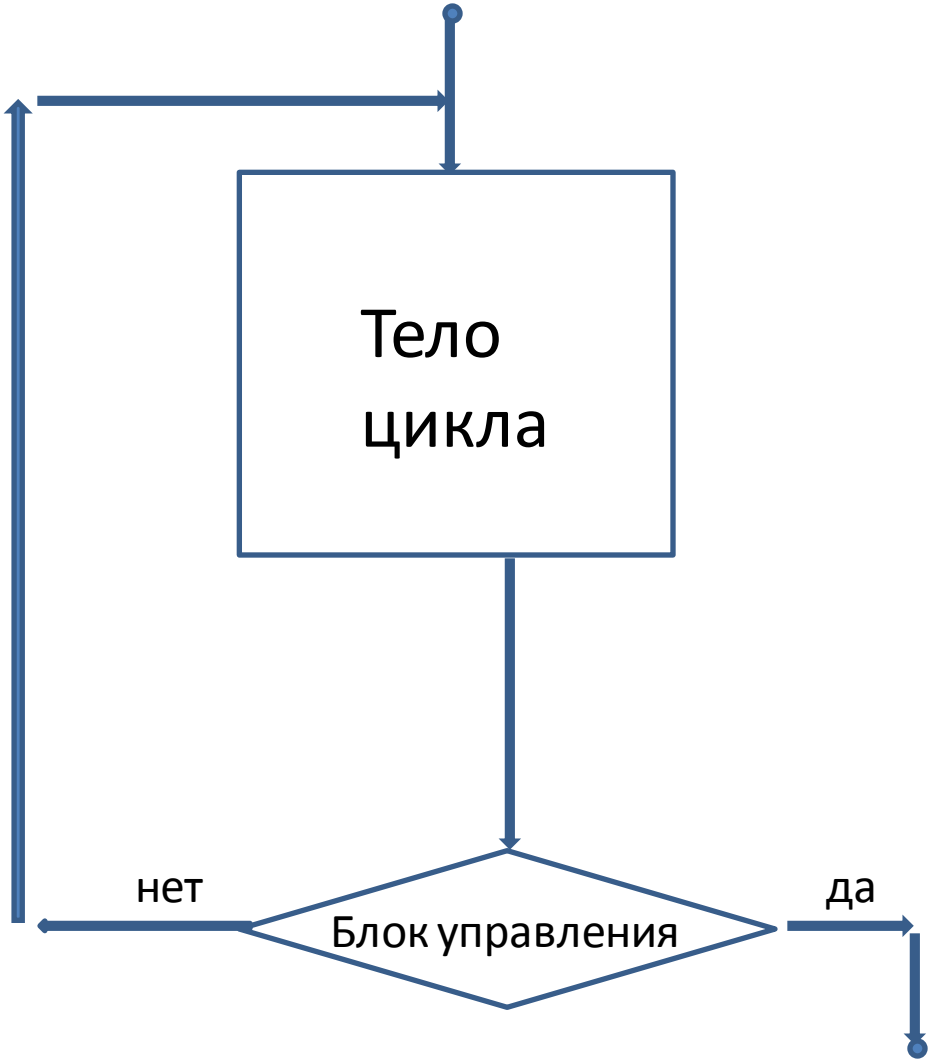



```
while(S)
{
  A;
}
```

В этом операторе тело цикла будет выполняться до тех пор, пока значение выражения **S** истинно.

Если при входе в цикл значение **S** есть **False**, тело цикла не выполнится ни разу.

Оператор цикла с постусловием


Тело
цикла
нет
Блок управления
да

- Оператор начинается словом **do**.
- Затем следуют операторы, составляющие тело цикла.
- За ними записывается слово **while** и логическое выражение, определяющее условие завершения цикла.

do

<оператор> ;

<оператор> ;

. . .

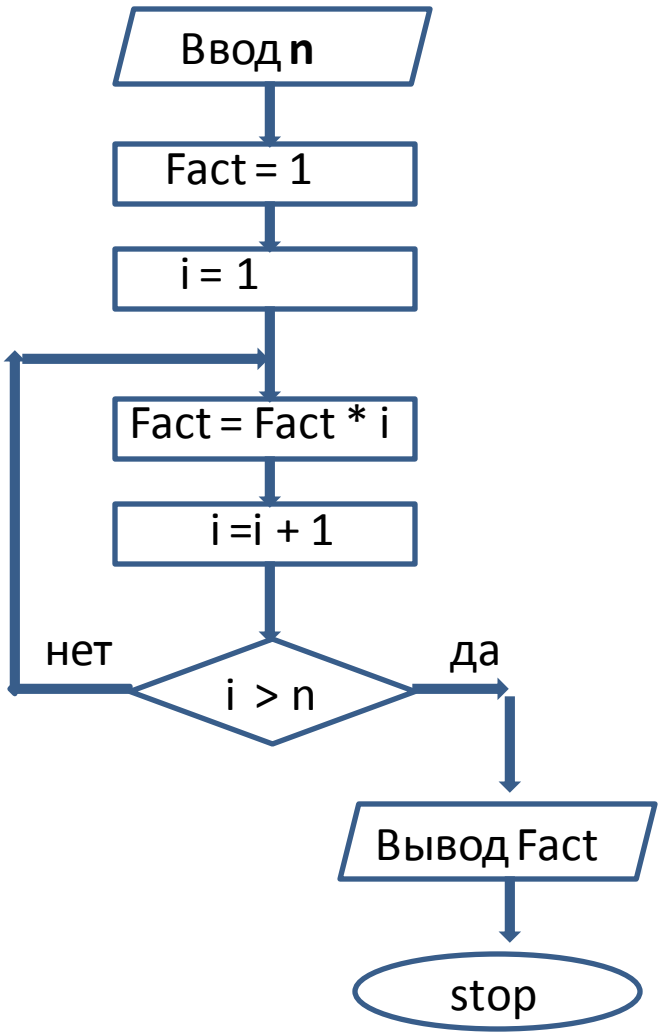
<оператор> ;

тело цикла

while <условие выхода из цикла> ;

- При работе такого цикла, сначала выполняются все операторы тела цикла, затем вычисляется логическое выражение, записанное после **while**.
- Если значение этого выражения **true**, повторяется выполнение тела цикла.
- Если значение логического выражения **false,** цикл заканчивается.

Пример. Подсчёт факториала числа **n**.




```
{
int Fact, i, n;

 cin >> ‘n= \n‘;
 cout << n;
 Fact = 1; //подготовка
 i = 1;       //    цикла
   //цикл с постусловием
do
   Fact = Fact * i;
   i = i + 1;
 while i > n;
 cout << “n! = “  <<Fact <<endl;
return 0;  }
```

РЕГУЛЯРНЫЙ ТИП

МАССИВ

- Массив - это упорядоченная непрерывная в памяти совокупность однотипных компонентов, имеющая имя.
- Для того чтобы выбрать один компонент, необходимо указать имя массива и индекс, т.е. номер компонента в массиве.

A

- Тип компонента называется базовым типом для массива.

Массив задается конструкцией:

<тип компонентов> <имя> [<число элементов>]

Например:

```
#define NMAX 100
int  main ()
{
    int vek[NMAX];
 double A[20];
…..
```

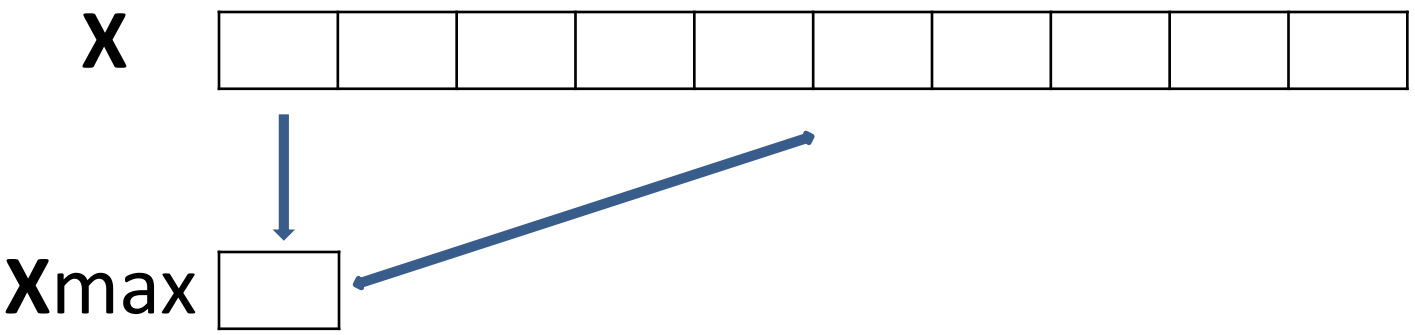
!! Индексация в массиве в языке С/С++ начинается с **0**, т.е. индекс у самого первого элемента в массиве равен 0

Обращение к элементам массива

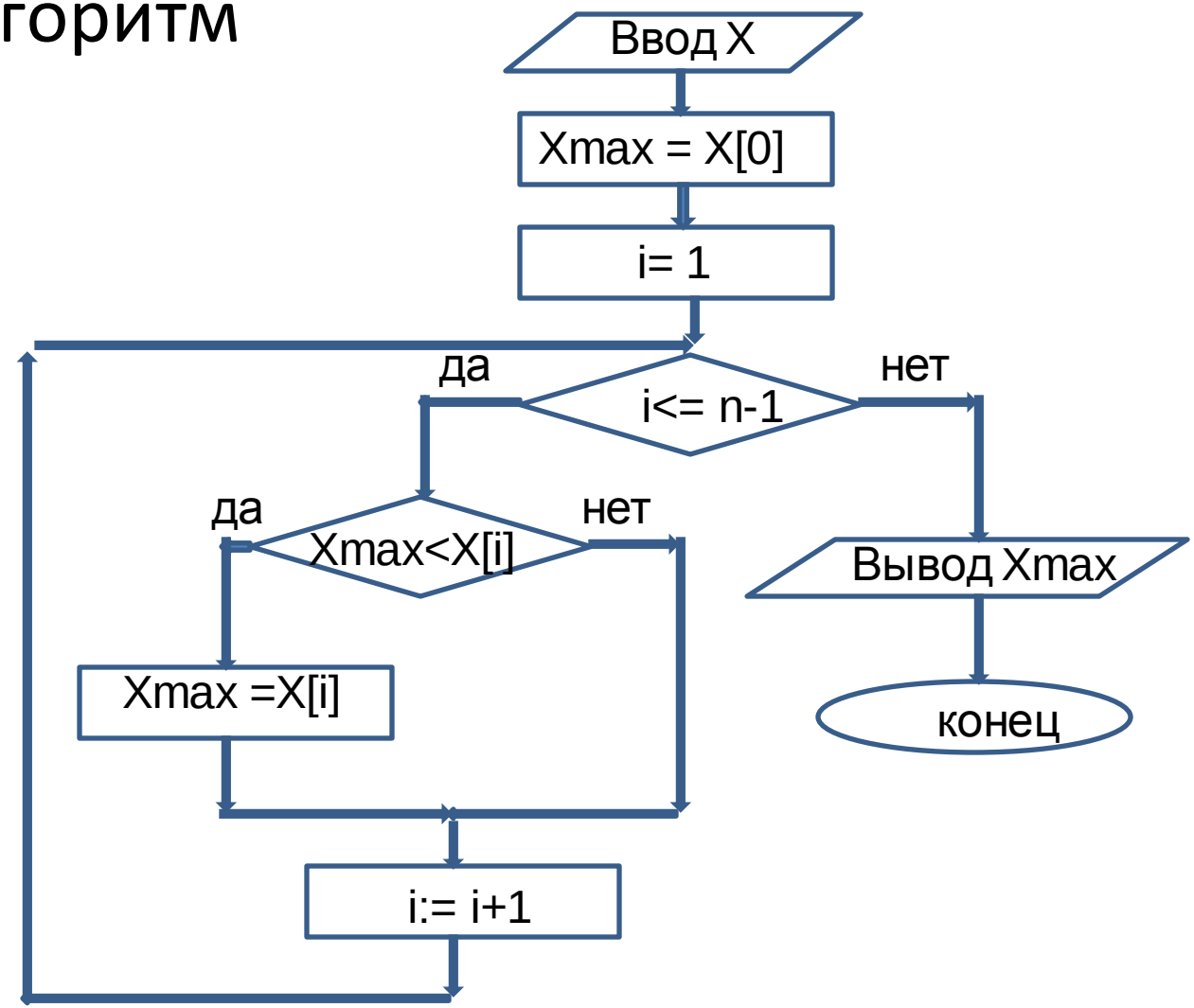
- Чтобы обратиться к элементу массива, надо написать имя массива и за ним в квадратных скобках индекс.
- Например: **x[10]**, **x[i]**, **x[i+k]**.
- Помним про нумерацию элементов.

Пример программы с использованием массива.

- Пусть задан **x** - массив из **n** чисел.
- Надо найти максимальное значение в массиве.



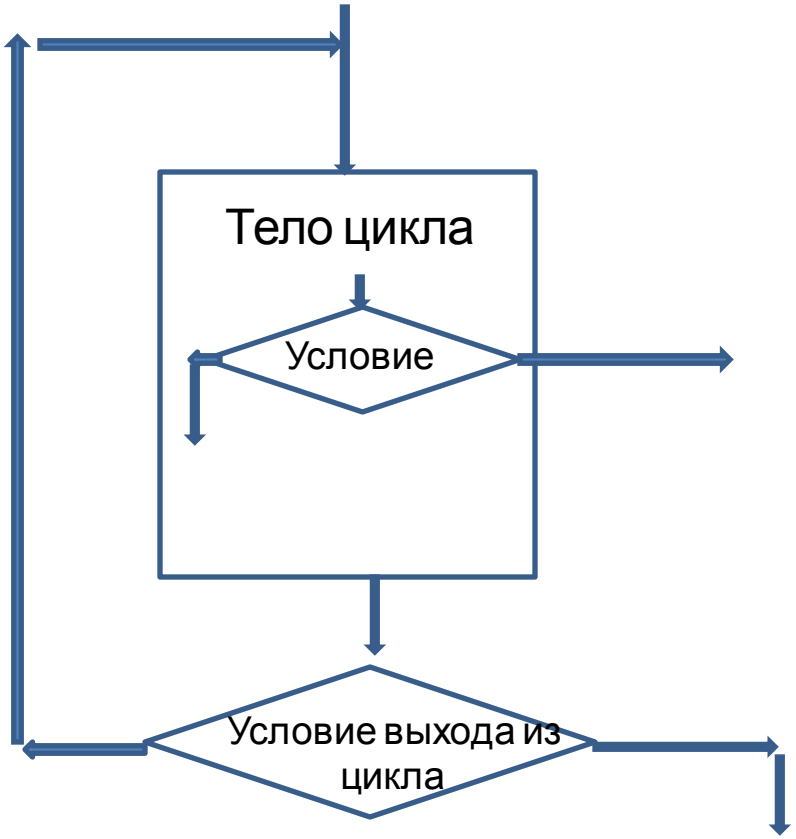
- Алгоритм


Ввод X
Xmax = X[0]
i= 1
да
i<= n-1
нет
да
Xmax<X[i]
нет
Вывод Xmax
Xmax =X[i]
конец
i:= i+1

```
#dfefine NMAX 50
 int main ()
{
    int i;
    int x [NMAX];
    float Xmax;
    printf("Введите массив x");
    for (i=0;i<NMAX;i++)
    {
    scanf("%f" , &x[ i ] ); }
      Xmax = x[ 0 ];   //подготовка цикла
```

```
    for (i =1; i<n; i++) {
        if x[ i ]  > Xmax
         { Xmax := x[ i ]; }  //if
       } //for
      printf(" Xmax = %f", &Xmax);
  return 0;
}
```

Досрочный выход из цикла



Это не базовая структура.

Метод флажка

- Вводится некоторая переменная (флажок), принимающая два значения.
- При одном значении цикл должен завершаться досрочно, при другом значении, надо продолжать вычисления в цикле.
- Проверка значения этой переменной (флажка) добавляется к критерию выхода из цикла.

Пример.

В одномерном массиве найти, под каким номером стоит первый чётный по значению элемент.

Метод решения.

- Будем нумеровать элементы исходного массива, начиная с нуля.
- Организуем просмотр исходного массива с начала до конца.
- При обнаружении чётного значения, его номер надо запомнить в результирующей переменной и закончить просмотр массива, то есть завершить цикл досрочно.

- В качестве флажка в данной задаче можно использовать результирующую переменную.
- В эту переменную перед началом анализа массива надо задать значение ноль.
- При этом признаком завершения цикла является значение этой переменной равное номеру чётного элемента массива, то есть большее нуля.
- Нулевой результат означает отсутствие в массиве чётных чисел.

Введём следующие обозначения.

x - исходный массив

n - число элементов **x**

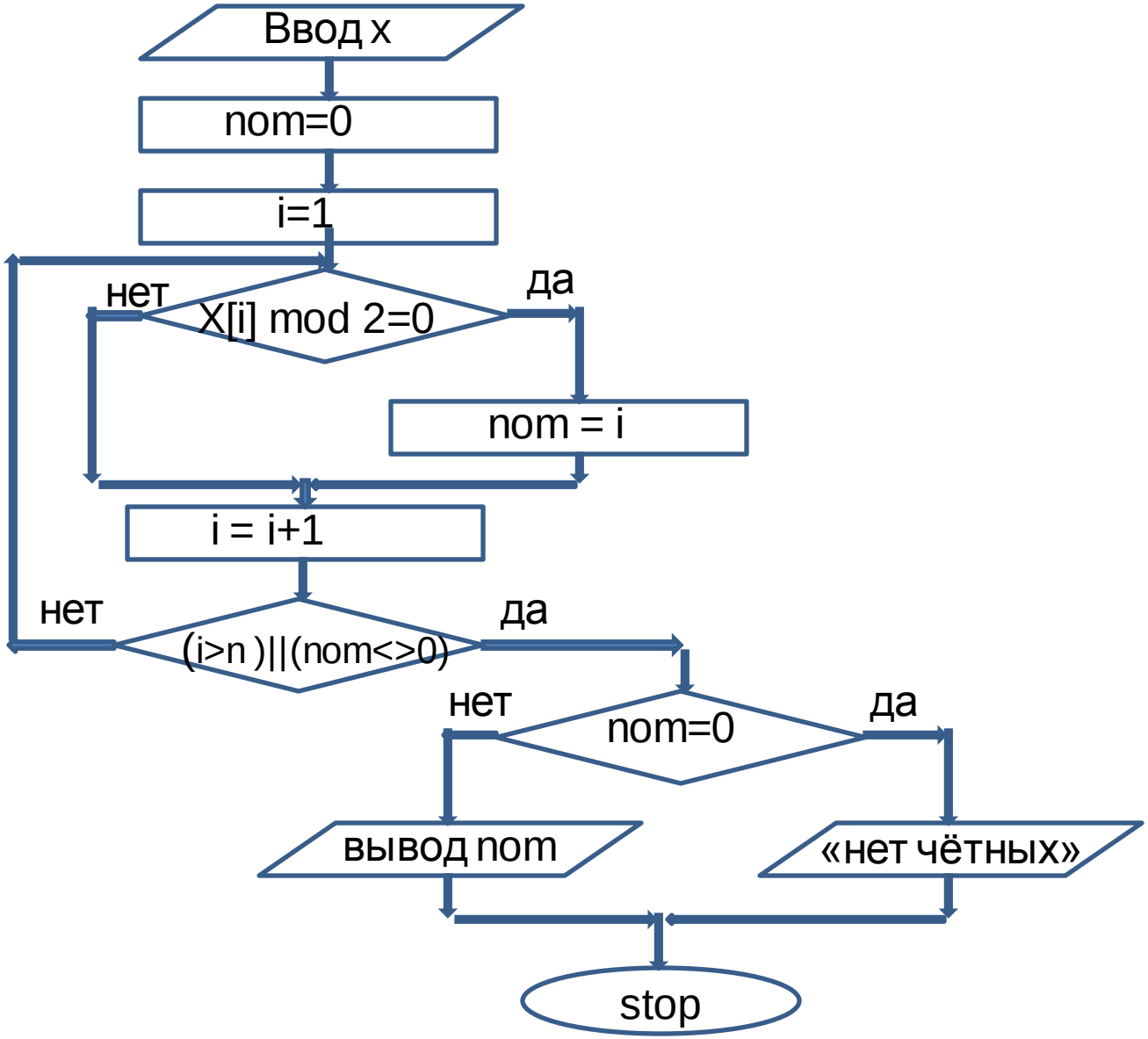
nom - искомая переменная

i – номер исследуемого элемента массива x

Исходные данные				
Имя	Смысл	Тип	Структура	Диапазон
n	Число элементов в x	целый	простая переменная	n>0
x	Исходный массив	вещественный	1-мерный массив	

Выходные данные				
nom	Искомое число	целый	простая переменная	nom >=0

Промежуточные данные				
i	Порядковый номер текущего элемента массива	целый	простая переменная	


Ввод x
nom=0
i=1
нет
X[i] mod 2=0
да
nom = i
i = i+1
нет
(i>n)||(nom<>0)
да
нет
nom=0
да
вывод nom
«нет чётных»
stop

```
#define n 8                    {длина массива}
int main ()  {
 int x [n] of integer;
      int nom,i;
{
   for (i=1 ; i<n; i++) do  //цикл для ввода массива
    {
      printf('x[' , i , ']= ');
      scanf(&x[i]);
    };
```

```
nom=0;  //определение флажка (результата)
  i=1;
  do
    if (x[i] % 2 == 0)
    { nom=i; } // анализ эл-та
    i++;
  while (i <= n) || (nom <>0);
                    //  печать результата
  if (nom == 0) {printf(‘ not ‘) ;}
                  else  { printf(‘nom=%d‘, nom); }
return 0;}
```