

4. Конструкции структурного программирования в Си

4.1. Ветвление

Оператор проверки условия *if [else]*

Синтаксис:

if (логическое выражение)

{действия при истинном значении
выражения}

[else {действия при ложном значении
выражения}]

Пример

...

...

```
int m = 12, n = 18;
```

```
if (m < n)
```

```
    printf ("Сумма чисел %d", m+n);
```

```
    else printf ("Произведение чисел %d", m*n);
```

...

...

Пример сложного условия

```
...  
int x = 5, y = 7, z = 3;  
int min;  
if (x < y && x < z)  
    min = x;  
else if (y < x && y < z)  
    min = y;  
    else min = z;  
...
```

Поле шахматной доски определяется парой натуральных чисел, каждое из которых не превосходит 8. По введенным координатам двух полей выясните, одного ли цвета эти поля.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|-------|-------|-------|-------|-------|-------|-------|-------|
| 1 | Black | White | Black | White | Black | White | Black | White |
| 2 | White | Black | White | Black | White | Black | White | Black |
| 3 | Black | White | Black | White | Black | White | Black | White |
| 4 | White | Black | White | Black | White | Black | White | Black |
| 5 | Black | White | Black | White | Black | White | Black | White |
| 6 | White | Black | White | Black | White | Black | White | Black |
| 7 | Black | White | Black | White | Black | White | Black | White |
| 8 | White | Black | White | Black | White | Black | White | Black |

Черные клетки – нечетные строки и нечетные столбцы или четные строки и четные столбцы

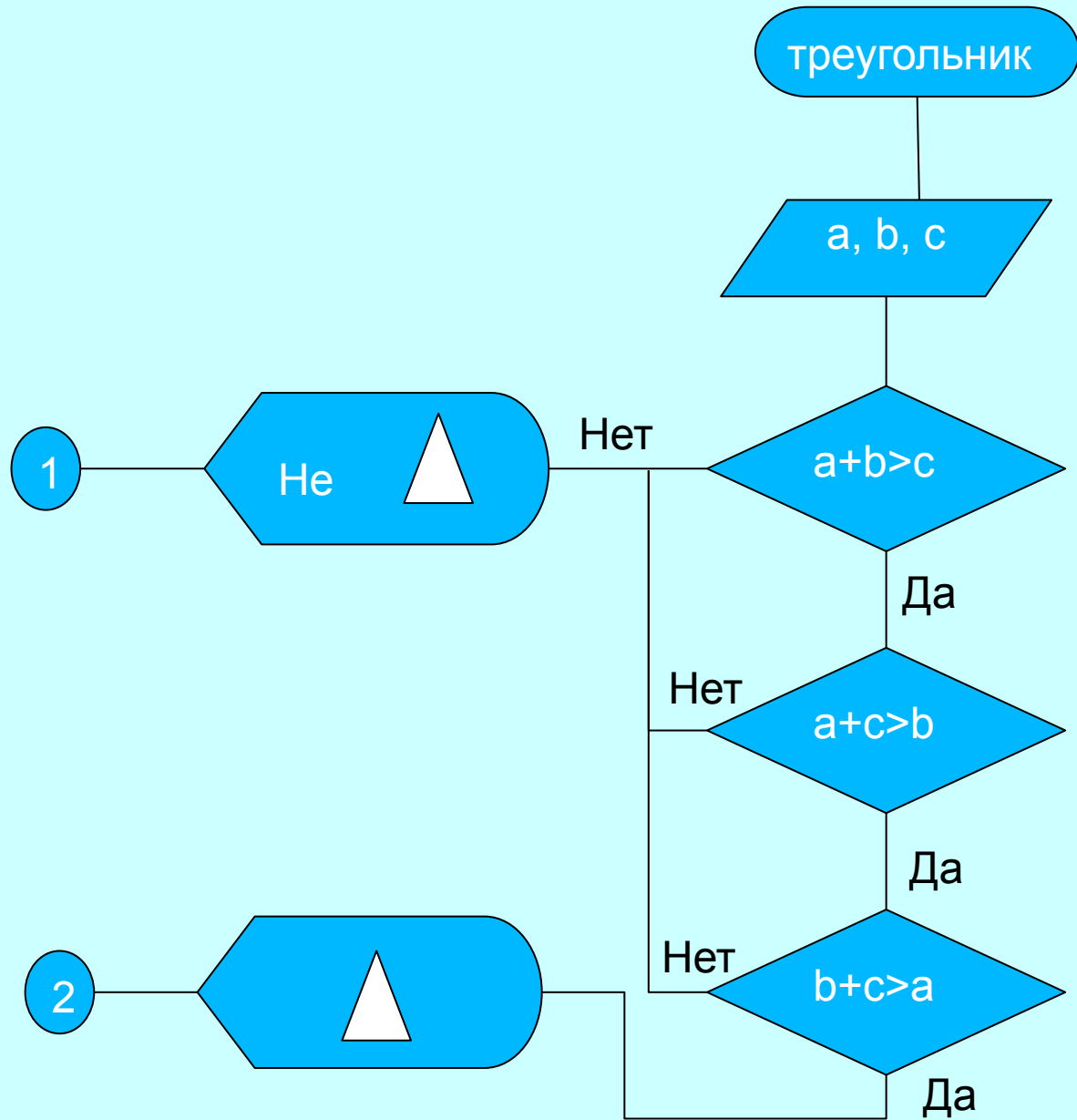
Белые клетки – нечетные строки и четные столбцы или четные строки и нечетные столбцы

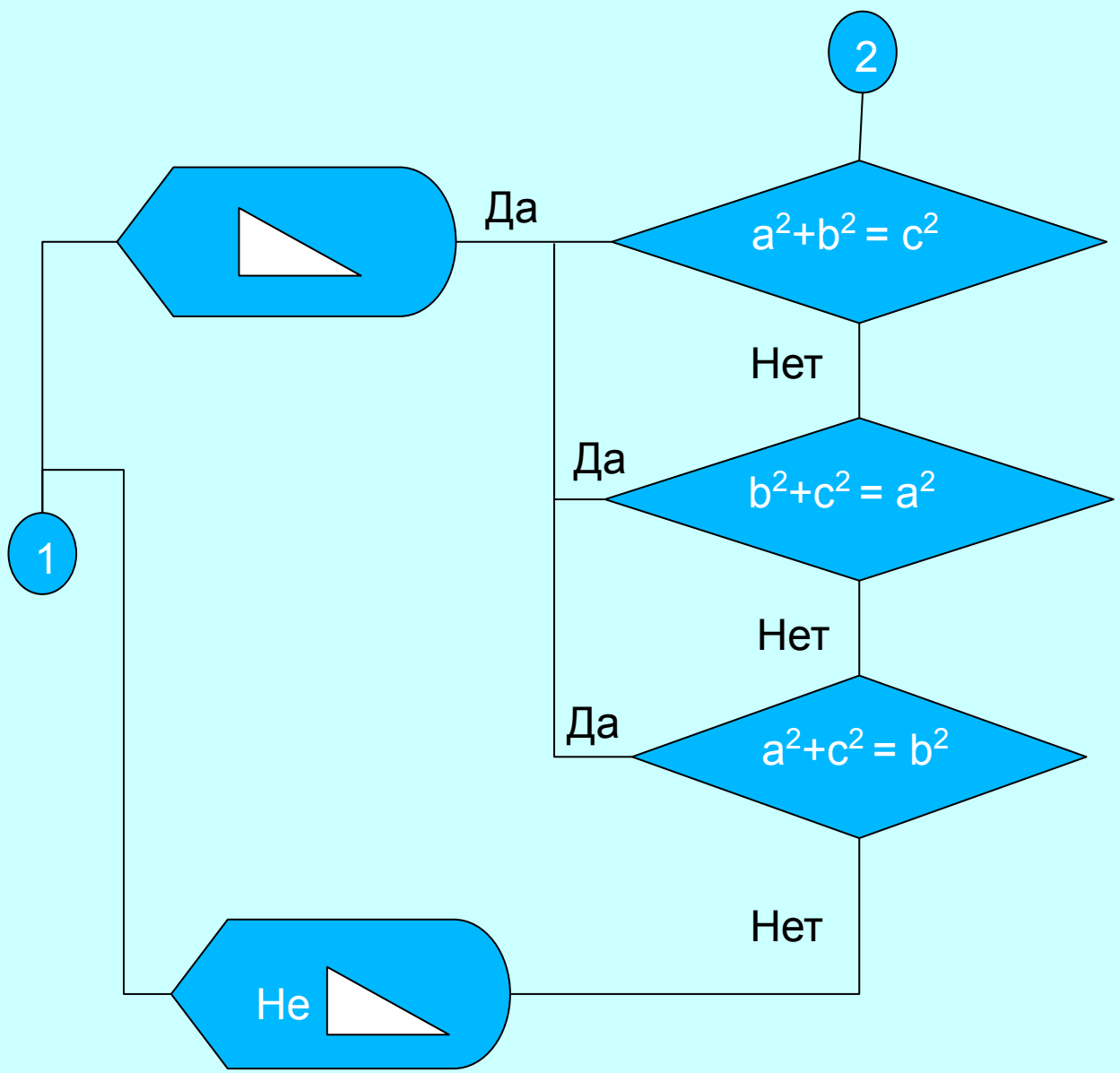
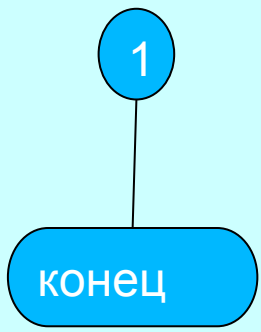
Тогда для черных клеток сумма строки и столбца всегда **четная**, а для белых клеток – всегда **нечетная**.

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[]) {
    system("chcp 1251");
    int x,y,x1,y1, a,b;
    printf("Введите номер столбца и номер строки первой клетки: ");
    scanf("%d%d",&x,&y);
    printf("Введите номер столбца и номер строки второй клетки: ");
    scanf("%d%d",&x1,&y1);
    a = x + y;
    b = x1 + y1;
    a = a+b;
    if (a%2==0) printf("Клетки одного цвета\n");
    else printf("Клетки разных цветов\n");
    system("pause");
    return 0;
}
```

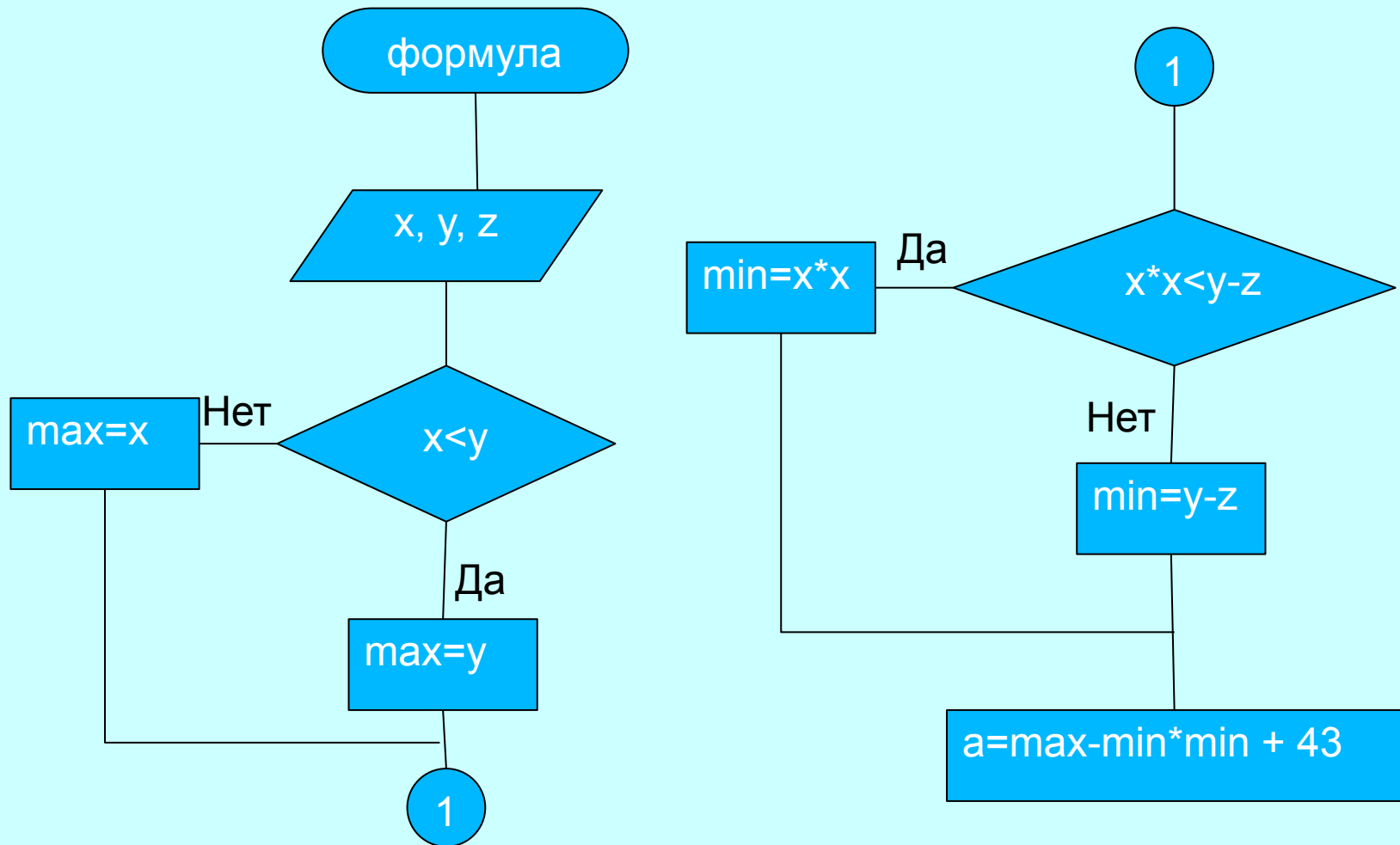
**Даны
вещественные
положительные
числа a, b, c .
Определите,
существует ли
треугольник со
сторонами a, b, c
и если да, то
определите,
является ли он
прямоугольным.**





```
int main(int argc, char *argv[]) {
    system("chcp 1251");
    float a,b,c;
    printf("Введите три стороны треугольника: ");
    scanf("%f%f%f",&a,&b,&c);
    if (a+b>c&&a+c>b&&c+b>a)
        if (a*a+b*b==c*c||a*a+c*c==b*b||c*c+b*b==a*a)
            printf("Прямоугольный треугольник\n");
        else printf("Не прямоугольный треугольник\n");
    else
        printf ("Отрезки не могут образовать
                треугольник\n ");
    system("pause");
    return 0;
}
```


Даны три числа x, y, z . Найдите
 $a = \max(x, y) - \min^2(x^2, y - z) + 43$

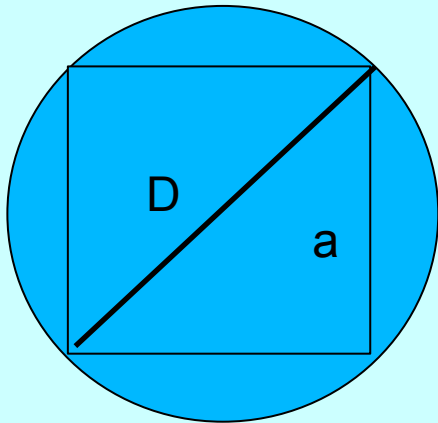


```
int main(int argc, char *argv[]) {
    system("chcp 1251");
    float x,y,z,max,min;
    float a;
    printf("Введите x,y,z: ");
    scanf("%f%f%f",&x,&y,&z);
    if (x<y) max = y; else max = x;
    if (x*x<(y-z)) min = x*x; else min = y-z;
    a = max - min*min +43;
    printf("Значение a = %.3f\n",a);
    system("pause");
    return 0;
}
```

Известны площади круга и квадрата.

Определить:

- уместится ли круг в квадрате
- уместится ли квадрат в круге



$$S_{\text{круга}} = \pi r^2 = \pi D^2 / 4$$

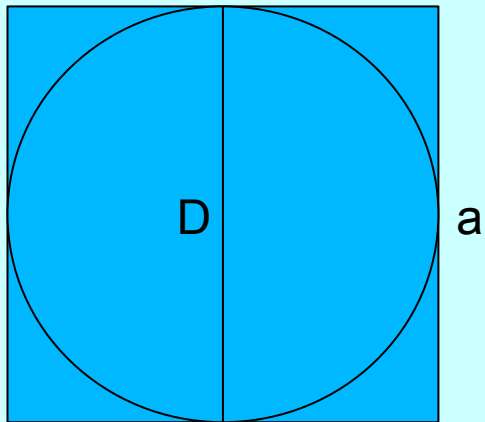
$$S_{\text{квадрата}} = a^2$$

$$D^2 = 2a^2$$

$$4S_{\text{круга}} / \pi \geq 2 S_{\text{квадрата}}$$

$$D^2 = 2 S_{\text{квадрата}}$$

$$D^2 = 4S_{\text{круга}} / \pi$$



$$4S_{\text{круга}} / \pi \leq S_{\text{квадрата}}$$

```
int main(int argc, char *argv[]) {
    system("chcp 1251");
    printf("Введите площадь круга и квадрата: ");
    float circle, quadro;
    scanf("%f%f", &circle, &quadro);
    if(4*circle/M_PI >= 2*quadro)
        printf("Квадрат помещается в круге\n");
    else
        printf ("Квадрат не помещается в круге\n");
    if (4*circle/M_PI <= quadro)
        printf("Круг помещается в квадрате\n");
    else
        printf ("Круг не помещается в квадрате\n");
    system("pause"); return 0;}

```

```
#include <stdio.h>
#include <stdlib.h>

/* Дано двузначное число. Определить: входит ли в него цифра 3
 */

int main(int argc, char *argv[]) {
    system("chcp 1251");
    printf("Введите двузначное число: ");
    int x,y,flag = 0;
    scanf("%d",&x);
    y = x%10;
    if (y==3) flag++;
    x = x-y;
    if (x==30) flag++;
    if (flag) printf("В числе есть цифра 3\n");
    else printf("В числе нет цифры 3\n");
    system("pause");
    return 0;
}
```

```
#include <stdio.h>
#include <stdlib.h>
```

```
/* Определить количество дней в году, который вводит пользователь. В високосном году - 366 дней, тогда как в обычном 365. Високосными годами являются все года делящиеся нацело на 4 за исключением столетий, которые делятся нацело на 400. */
```

```
int main(int argc, char *argv[]) {
    system("chcp 1251");
    printf("Введите год: ");
    int year;
    scanf("%d",&year);
    if (year%4==0&&year%100!=0)
        printf("В %d году 366 дней (високосный)\n",year);
    else if (year%100==0&&year%400!=0)
        printf("В %d году 366 дней (високосный)\n",year);
    else printf("В %d году 365 дней (не високосный)\n", year);
    return 0;
}
```

4.2. Множественный выбор

Синтаксис: *switch* (выражение)

```
{  
case значение выражения1: операторы;  
case значение выражения2: операторы;  
...  
[default: операторы;]  
}
```

Выражение в *switch*

int, long int, char, ... целочисленные типы

Принцип работы *switch*

- значение ***выражения*** сравнивается со значением, указанным в первом блоке *case*
- если значения совпали, выполняются операторы из первого блока *case*
- управление без проверки условия передается в последующие блоки *case*
- если *case* заканчивается оператором *break*, управление передается на оператор, следующий за блоком *switch*

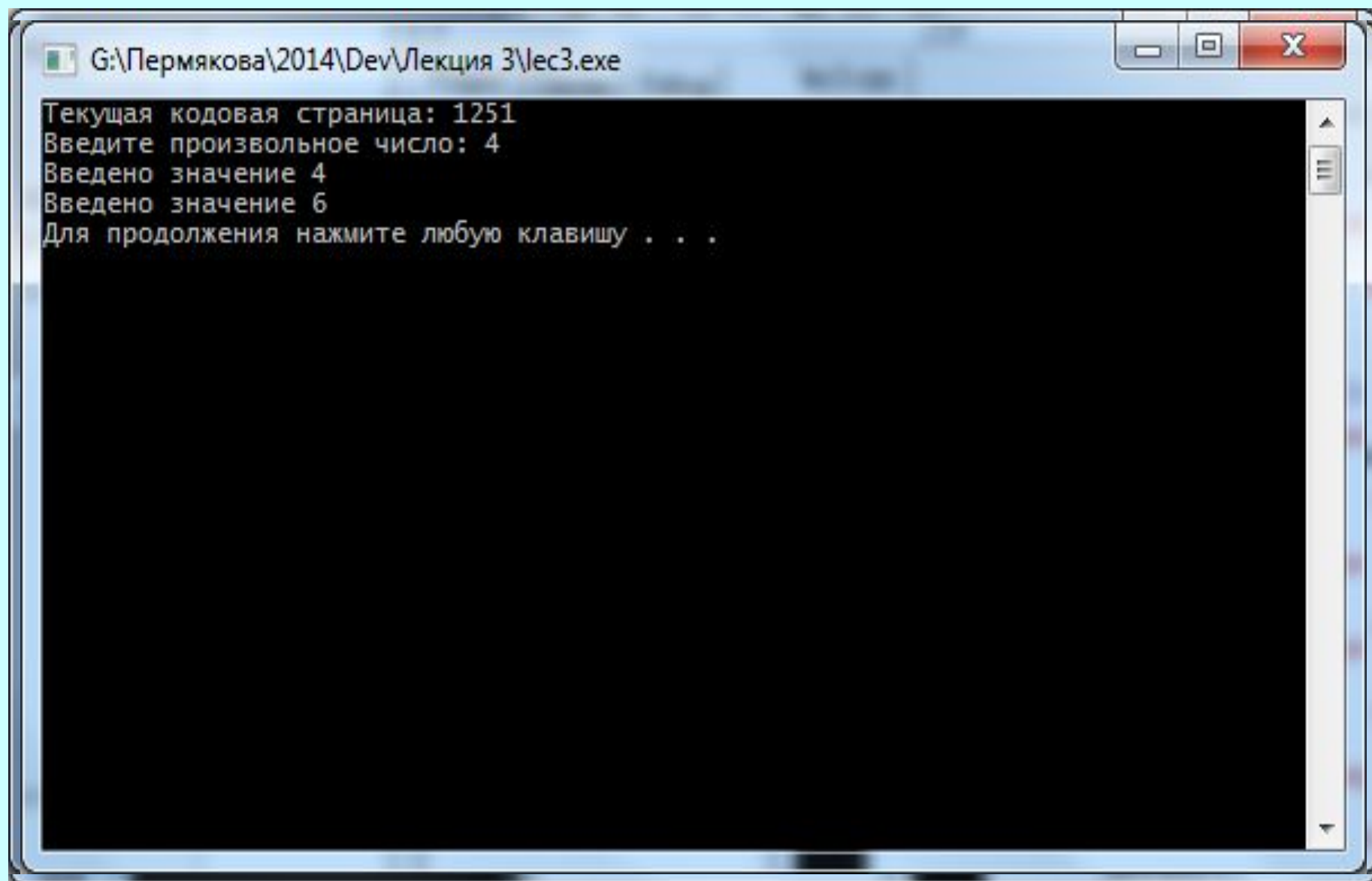
Принцип работы *switch*

- если совпадения значений не произошло, последовательно проверяются все следующие *case*
- в блок *default* управление передается в случае, если не произошло ни одного совпадения значений
- для определения диапазонов используется перечисление *case*:

case a: case b: case c: ...

```
int k;
printf("Введите произвольное целое число: ");
scanf("%d",&k);
switch (k)
{ case 1:case 2: case 3:
    printf("Введено число 1 или 2 или 3\n");
    break;
case 4:
    printf("Введено число 4\n");
case 6:
    printf(" Введено число 6\n");
    break;
default:
printf(" Управление передано в блок DEFAULT \n");
}
```

Результаты работы



The image shows a screenshot of a Windows command prompt window. The title bar at the top reads "G:\Пермякова\2014\Dev\Лекция 3\lec3.exe". The window contains the following text:

```
Текущая кодовая страница: 1251  
Введите произвольное число: 4  
Введено значение 4  
Введено значение 6  
Для продолжения нажмите любую клавишу . . .
```

The text is displayed in a monospaced font on a black background. The window has standard Windows window controls (minimize, maximize, close) in the top right corner.

С клавиатуры вводятся три числа DD MM YYYY, обозначающие день, месяц и год. Проверить корректность введенной даты.

```
int main(int argc, char *argv[]) {
    system("chcp 1251");
    int dd,mm, yy, f = 0;
    printf("Вводите день, месяц, год: ");
    scanf("%d%d%d", &dd,&mm,&yy);
    if (dd>31||mm>12) f = 1;
    else {
        switch (mm) {
            case 1: if (dd>31) f=1; break;
            case 2: if (yy%4==0&&yy%100!=0||yy%100==0&&yy%400==0)
                    {if (dd>29) f=1; }
                    else if (dd>28) f=1; break;
        }
    }
}
```

```
case 3: if (dd>31) f=1; break;
case 4: if (dd>30) f=1; break;
case 5: if (dd>31) f=1; break;
case 6: if (dd>30) f=1; break;
case 7: if (dd>31) f=1; break;
case 8: if (dd>31) f=1; break;
case 9: if (dd>30) f=1; break;
case 10: if (dd>31) f=1; break;
case 11: if (dd>30) f=1; break;
case 12: if (dd>31) f=1; break;
default: f=0;
```

```
}
```

```
}
```

```
if (f) printf("Введенная дата некорректна\n");
```

```
system("pause");
```

```
return 0;
```

```
}
```

АЛГОРИТМ РАЗБИЕНИЯ ЧИСЛА НА ЦИФРЫ

$1567 \% 10 = 7$ (получение остатка от деления)

$1567 / 10 = 156$ (деление на 10 нацело)

$156 \% 10 = 6$ (получение остатка от деления)

$156 / 10 = 15$ (деление на 10 нацело)

$15 \% 10 = 5$ (получение остатка от деления)

$15 / 10 = 1$ (деление на 10 нацело)

$1 \% 10 = 1$ (получение остатка от деления)

$1 / 10 = 0$ (деление на 10 нацело)

4.3. Циклы

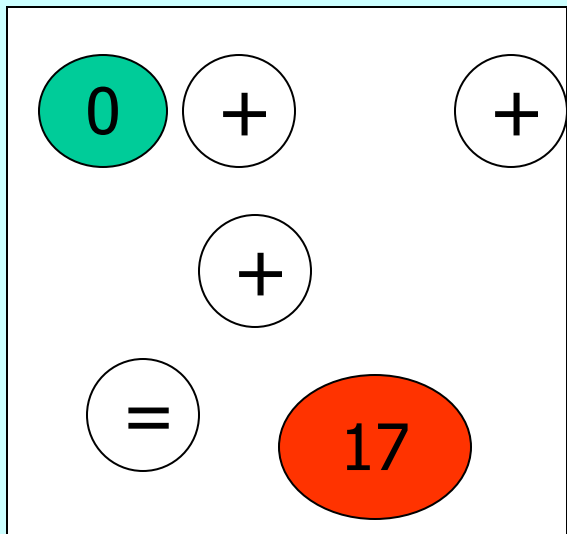
4.3.1. Цикл с фиксированным числом операций *for*

Цикл, это конструкция структурного программирования, повторяющая определенные действия (***итерации***) несколько раз.

Синтаксис:

for (***секция инициализации значения; секция проверки условия; секция коррекции***) {***тело цикла***}

Алгоритмы суммы и произведения



$S := 0$

для i от 1 до n

ВВОД k ;

$S := S + k$;

кц

рез S .

Алгоритм произведения

$S := 1$

для i от 1 до n

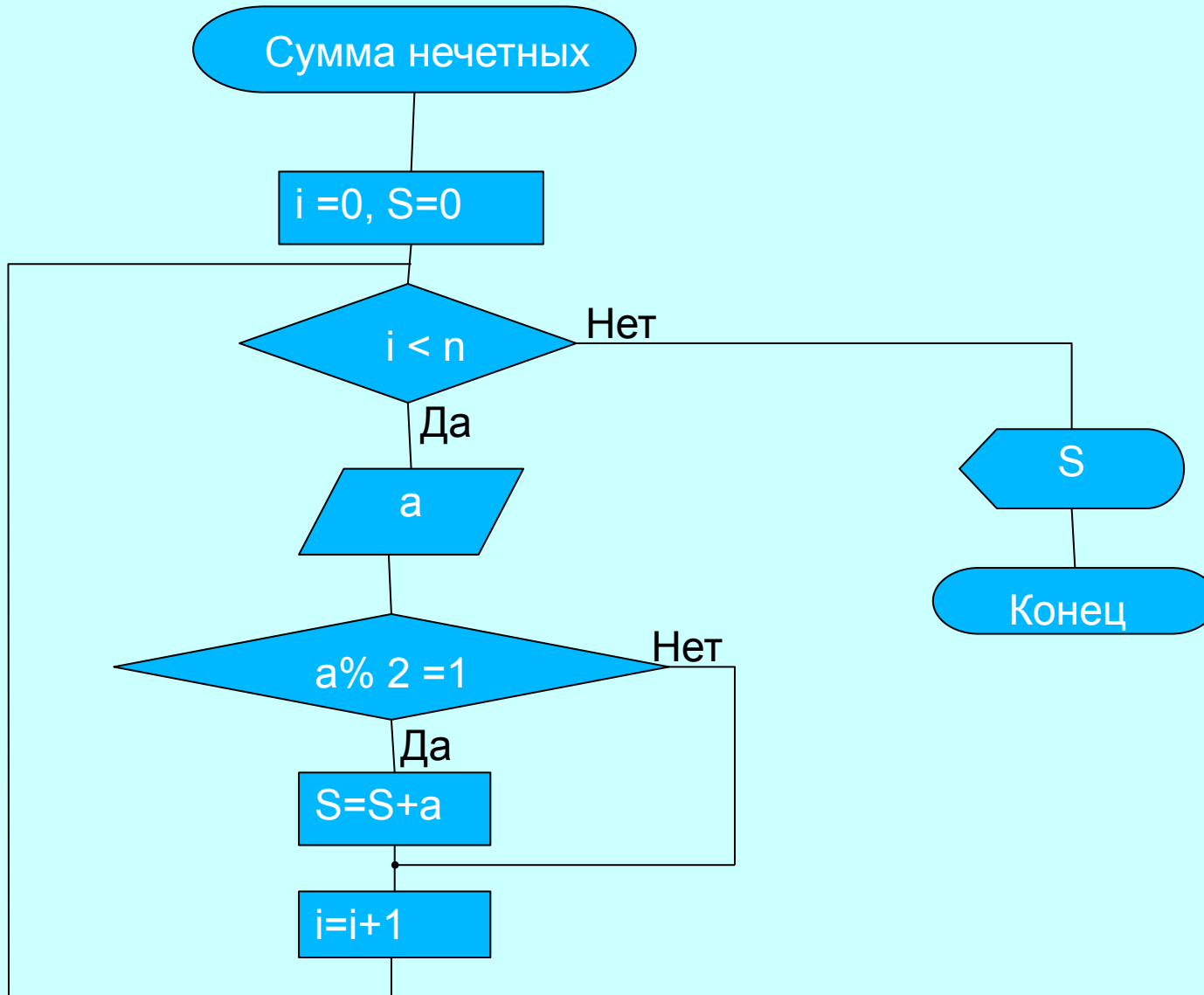
ввод k ;

$S := S * k$;

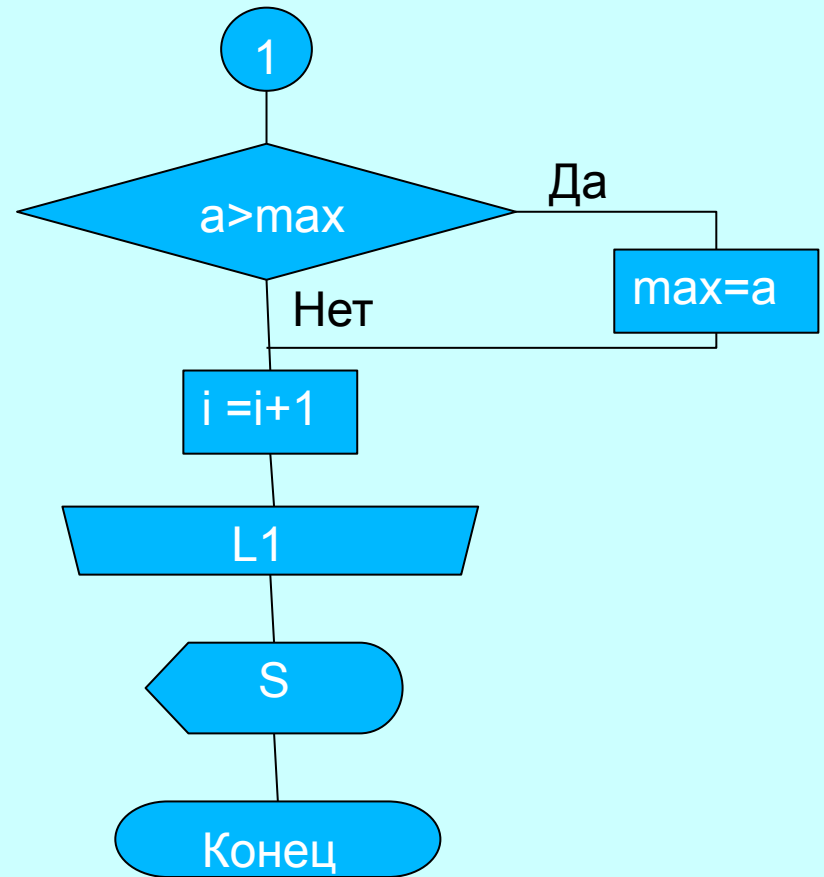
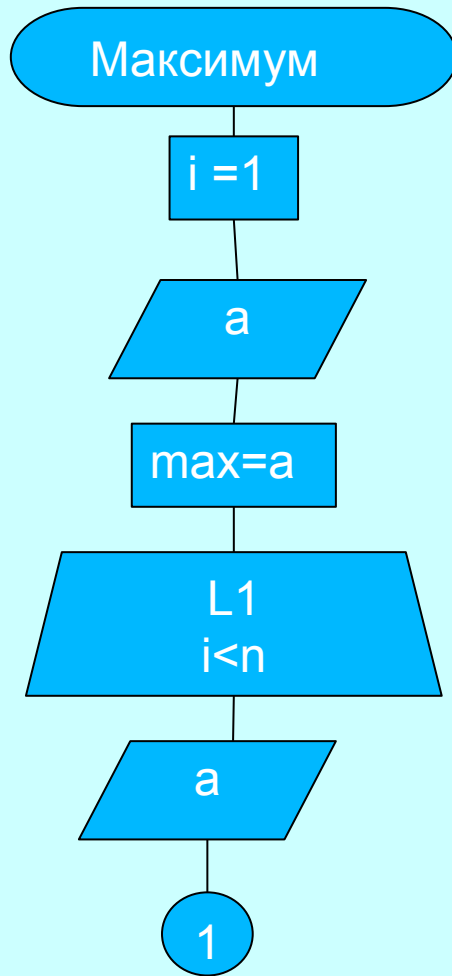
кц

рез S .

Дана последовательность из n произвольных чисел.
Найти сумму нечетных элементов



Дана последовательность из n произвольных чисел.
Найти значение самого большого элемента



Вывести на экран таблицу значений функции $x^2 + 2$ для $x = 1, 1.2, 1.4, \dots 3$.

| | |
|------------------|---------------|
| Таблица значений | |
| $x = 1$ | |
| $h = 0.2$ | |
| i от 1 до 15 | |
| | $f = x^2 + 2$ |
| | Вывод f, x |
| | $x = x + h$ |
| Конец | |