

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

$$\begin{array}{r} 0 + + \\ + \\ = 17 \end{array}$$

$S := 0$

для i от 1 до n

ввод k ;

$S := S + k$;

кц

рез S .

Алгоритм произведения

$S := 1$

для i от 1 до n

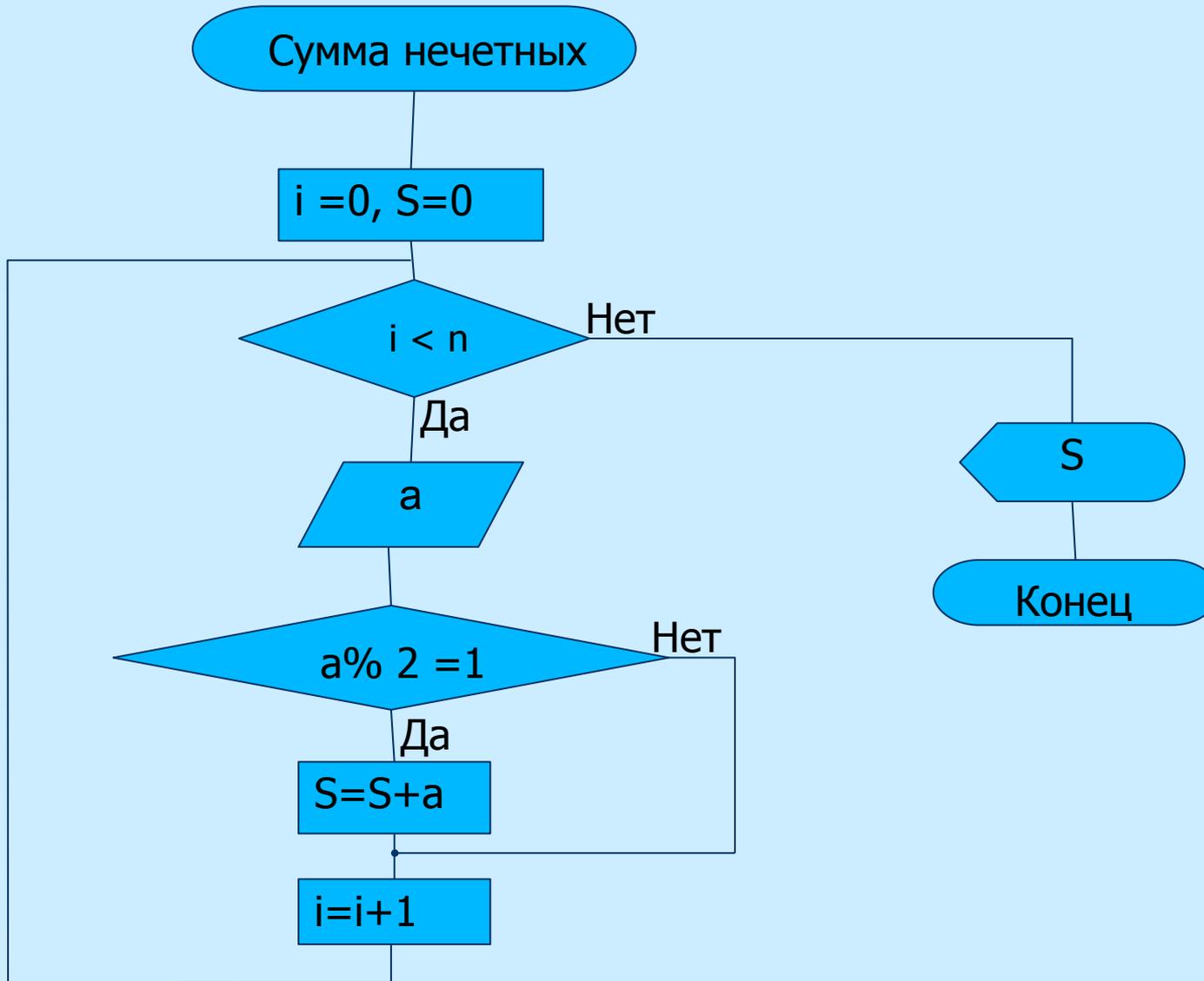
ввод k ;

$S := S * k$;

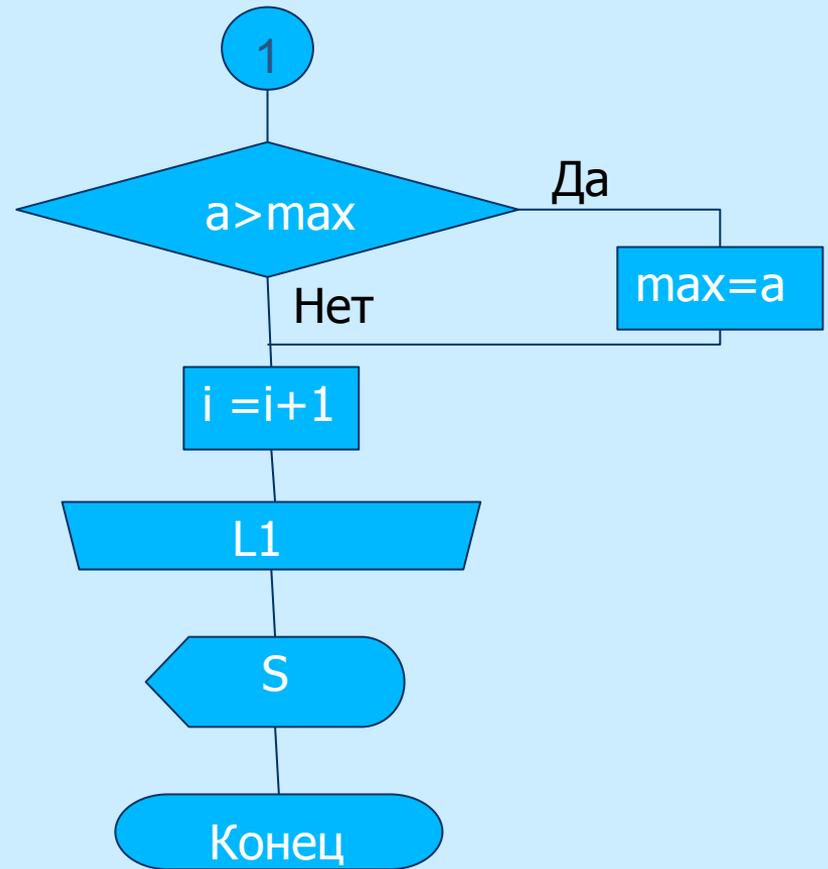
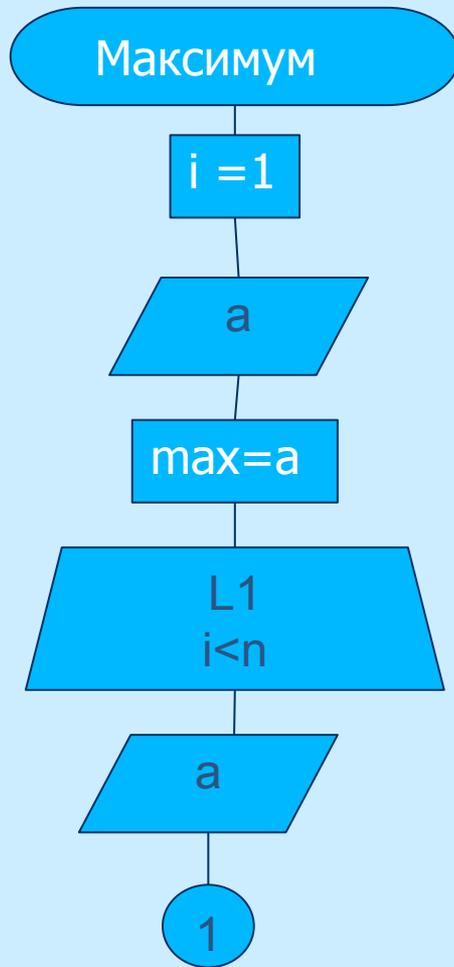
кц

рез S .

Дана последовательность из n произвольных чисел.
Найти сумму нечетных элементов



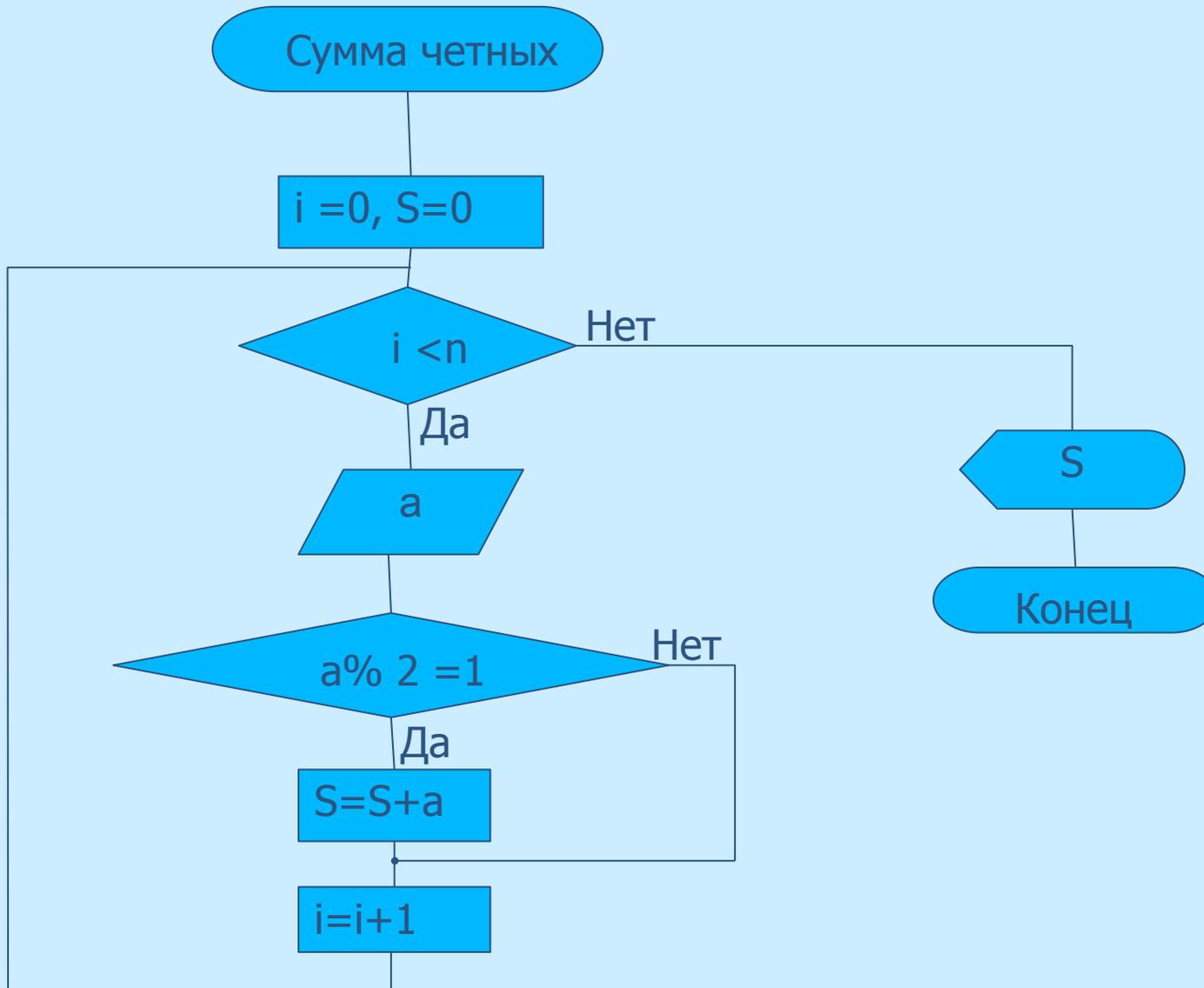
Дана последовательность из n произвольных чисел.
Найти значение самого большого элемента



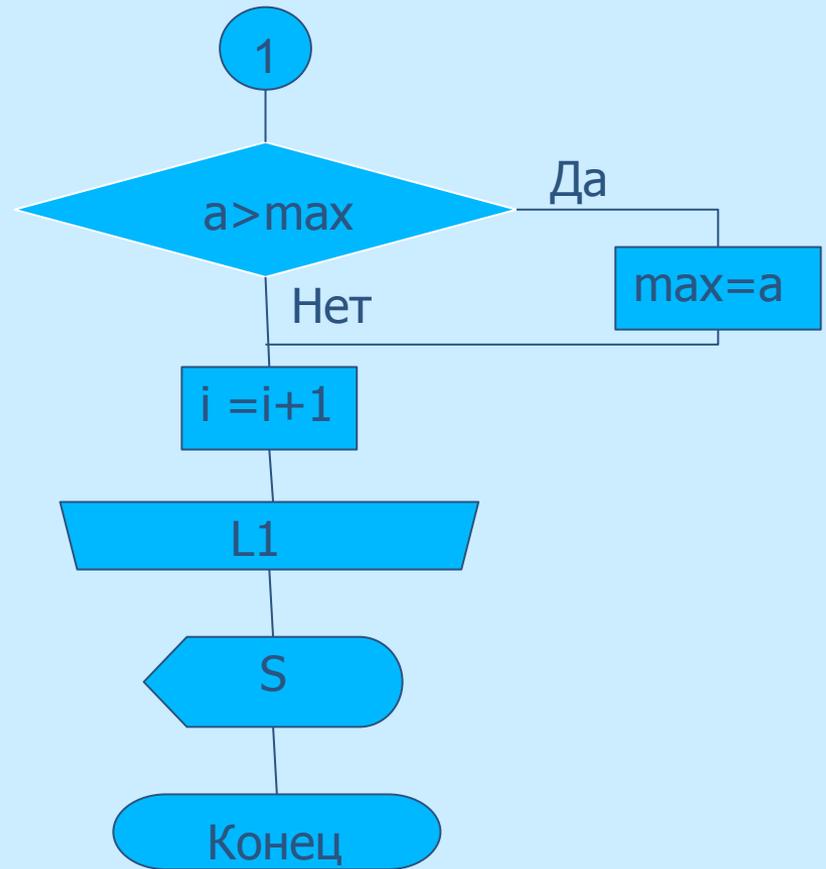
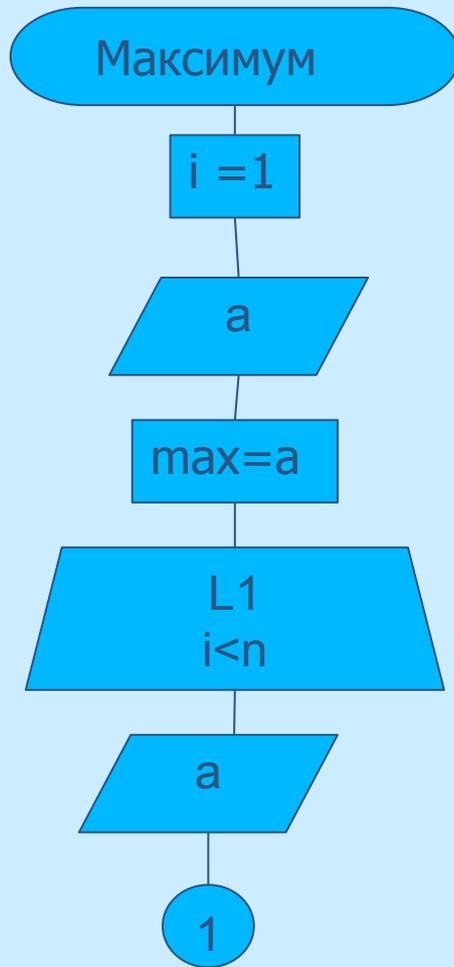
Вывести на экран таблицу значений функции $x^2 + 2$ для $x = 1, 1.2, 1.4, \dots 3$.

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

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	

Вычислить сумму $1 + 1/2 + 1/3 + 1/4 + \dots + 1/n$

```
int main(int argc, char *argv[])
{
    int n,i;
    printf("n = ");
    scanf("%d",&n);
    float S = 0;
    for(i=1;i<=n;i++)
        S+=1./i;
    printf("S = %f\n",S);
    system("PAUSE");
    return 0;
}
```

Рекуррентные соотношения

Рекуррентными соотношениями называются соотношения в которых задано начальное значение (a_0). Все остальные значения a_1, a_2, \dots, a_k вычисляются с использованием предыдущего.

Последовательность чисел a_0, a_1, a_2, \dots образуется по закону: $a_0 = 1$,
 $a_k = ka_{k-1} + 1/k$ ($k = 1, 2, \dots$). Дано натуральное число n . Получить a_1, a_2, \dots, a_n .

алг Формула нач

ВВОД n

вещ $a_0 = 1, a_i$

ВЫВОД a_0

для i от 1 до n

$a_i = i * a_0 + 1/i$

ВЫВОД a_i

$a_0 = a_i$

КЦ

КОН

```
int main(int argc, char *argv[])
{
    float a0=1.,ai;
    int n=10,i;
    printf("a0 = %f\n",a0);
    for(i=1;i<=n;i++){
        ai= i*a0+1./i;
        printf("a%d = %f\n",i,ai);
        a0=ai;
    }
    system("PAUSE");
    return 0;
}
```

Последовательность Фибоначчи образуется так: первый и второй члены последовательности равны 1, каждый следующий равен сумме двух предыдущих (1, 1, 2, 3, 5, 8, 13, ...). *Фибоначчи.*

алг ФИБОНАЧЧИ нач

ВВОД n

вещ $a0 = 1, a1 = 1$

ВЫВОД $a0, a1$

для i от 2 до n

$a_i = a0 + a1$

ВЫВОД a_i

$a0 = a1$

$a1 = a_i$

КЦ

КОН

```
int main(int argc, char *argv[])
{
    int a0=1,a1 = 1, ai;
    int n=6,i;
    printf("a0 = %d\n a1 = %d\n ",a0,a1);
    for(i=2;i<=n;i++){
        ai= a0+a1;
        printf("a%d = %d\n",i,ai);
        a0=a1;
        a1=ai;
    }
    system("PAUSE");
    return 0;
}
```

Найти сумму $2^1 + 2^2 + \dots + 2^{10}$. Операцию возведения в степень не использовать.

$$a_0 = 2, a_k = a_{k-1} * 2$$

```
алг СУММА нач  
ввод 10  
вещ a0 = 2, S=a0  
для i от 2 до n  
    a0 = a0*2  
    S = S + a0  
кц  
вывод S  
кон
```

Возможности цикла *for*:

- Уменьшение счетчика – *for(i = 10 ;i>=0;i- -)*
- Изменение шага - *for(i = 1,i<=10,i+=4)*
for(i = 1,i<=10,i=2)*
- Использование вещественных переменных в качестве счетчика - *for(x=0;x<10;x+=0.5)*
- Работа с символами - *for(c = 'A'; c < 'Я'; c++).*
- Возможность записывать несколько действий в одной секции -
for (i=1,j=1;i<10,j<10;i++,j+=4)

- Возможность опускать любое из выражений заголовка

for(;;) – бесконечный цикл, пустое условие всегда считается истинным;

4.3.2. Операторы безусловной передачи управления *continue* и *break*

- ***break*** досрочно завершает выполнение цикла. Управление передается оператору, следующему за циклом.

```
int n =15;  
for(int i=0;i<n;i++)  
{ int z = rand()%200;  
if (z>100) break;  
}
```

- ***continue*** пропускает все последующие операторы тела цикла и передает управление на в начало цикла.

```
int f = 1;  
do  
{  
int z = rand()%100;  
if (z>30) continue;  
if (z<10) f = 0;  
printf("%d",z);  
} while(f);
```

Ввести с клавиатуры 5 произвольных чисел,
найти и вывести на экран их сумму.

...

```
printf (" Вводите числа: \n");  
float S = 0,c;  
int n;  
for (int i=0;i<5;i++) {  
    printf("%d==> ",i+1);  
    n = scanf("%f",&c);  
    if(n!=1) {fflush(stdin);  
        i--;  
        continue;}  
    S+=c;  
}
```

G:\Пермякова\2014\Dev\Лекция 3\lec3.exe

Текущая кодовая страница: 1251

Вводите числа:

1==> 3

2==> 4

3==> и

3==> ы

3==> 5

4==> 6

5==> е

5==> е

5==> 2

Сумма чисел = 20.000

Для продолжения нажмите любую клавишу . . .

4.3.3. Циклы *while* и *do while*

Для решения задач, при выполнении которых необходимо проводить циклические действия до тех пор, пока истинно какое-либо условие

while() и *do while()*.

Синтаксис:

while (*условное выражение*)

{ *тело цикла*

}

Синтаксис: **do** {

тело цикла

} **while** (*условное выражение*)

- **while** - цикл с предусловием

(может ни разу не выполниться)

- **do while** - цикл с постусловием

(обязательно выполниться хотя бы один раз)

Ввести с клавиатуры произвольное количество чисел и найти сумму введенных чисел. Ввод продолжать до первого отрицательного числа.

```
printf ("Вводите числа: \n");  
float S = 0, c=0;  
int n;  
int i = 1;  
do {  
    printf("%d==> ", i);  
    i++;  
    n = scanf("%f", &c);  
    if(n!=1) {fflush(stdin);  
              i--;  
              continue;}  
    S+=c;  
}while (c>=0);  
printf("\n Сумма - %f\n", S);
```

C:\ F:\ИИ3\ИИ3.exe



Active code page: 1251

Вводите числа:

1==> 5

2==> 6

3==> t

3==> 0

4==> -12

Сумма - -1.000000

Press any key to continue . . .

```
...printf (" Вводите числа: \n");
float S = 0,c=0;
int n;
int i = 1;
printf("%d==> ",i);
    n = scanf("%f",&c);
    if(n!=1) {fflush(stdin);
                i--;
            }
    else {S+=c;}
    i++;
while (c>=0) {
    printf("%d==> ",i);
    i++;
```

```
n = scanf("%f",&c);  
if(n!=1) {fflush(stdin);  
        i--;  
        continue;}  
S+=c;  
}  
printf(" CyMMA - %f\n", S);  
system("PAUSE");
```

E:\13\13.exe

Текущая кодовая страница: 1251

Вводите числа:

1==> p

1==> 3

2==> 4

3==> -1

Сумма - 6.000000

Для продолжения нажмите любую клавишу . . .

Основные логические ошибки при использовании циклов.

- После заголовка цикла ставиться точка с запятой. Такой цикл считается компилятором пустым Например:

```
for(int i=0;i<10;i++);  
{ n+=10;  
  y-=15;  
}
```

Увеличение переменной *n* и уменьшение переменной *y* происходит за циклом, ровно один раз.

- Условие цикла заведомо ложно (цикл никогда не выполнится)
- Условие цикла никогда не станет ложным (цикл будет бесконечным)