

# Программирование разветвляющихся алгоритмов

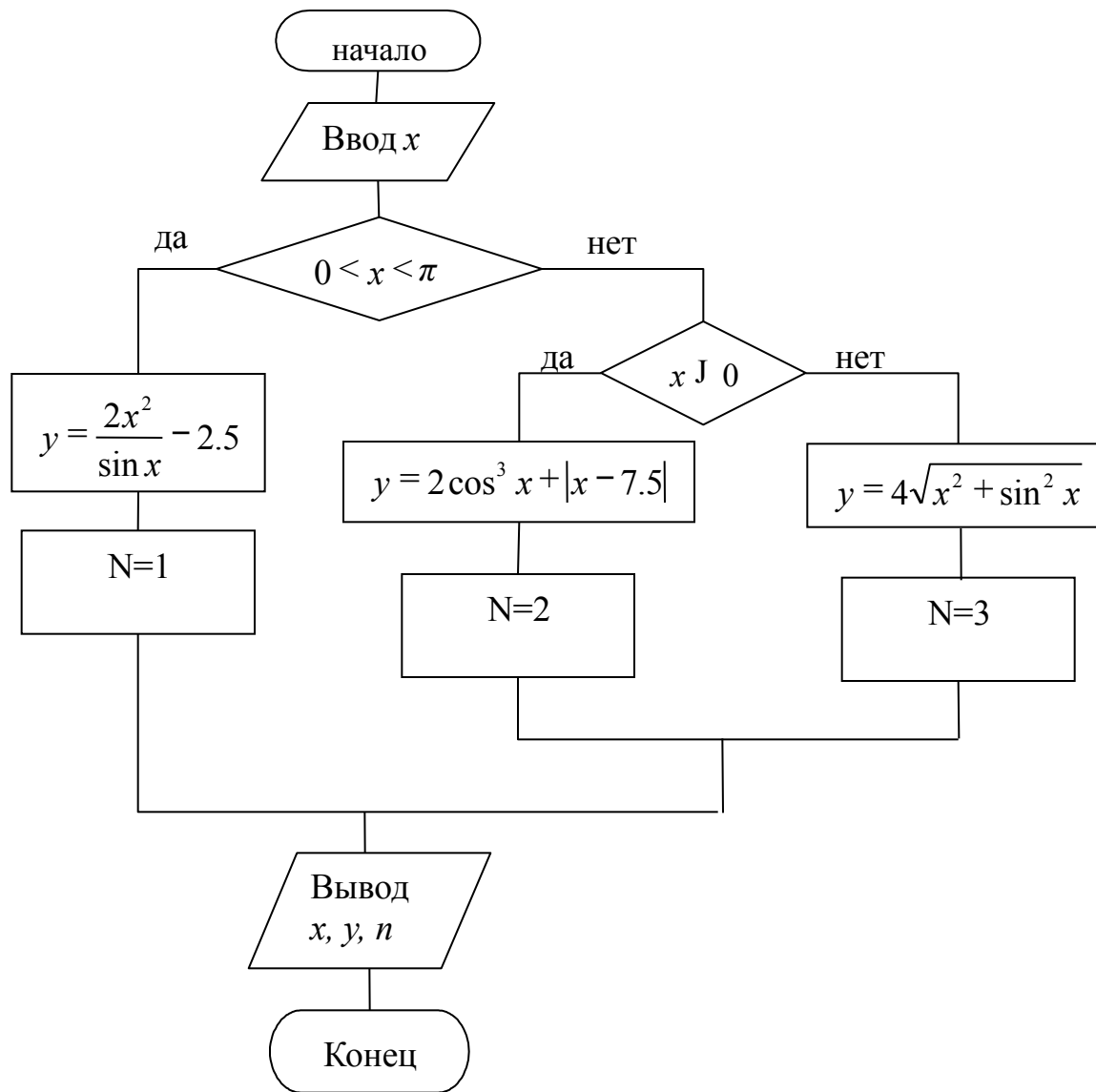
Лекция №5 по курсу «ОАИП»

**Разветвляющийся** - алгоритм, в котором некоторые действия выполняются один раз или не выполняются в зависимости от заданного условия.

**Пример 1.** Составить алгоритм вычисления функции.

$$y = \begin{cases} \frac{2x^2}{\sin x} - 2,5 & \text{если } 0 < x < \pi \\ 2 \cos^3 x + |x - 7,5| & \text{если } x \leq 0 \\ 4\sqrt{x^2 + \sin^2 x} & \text{в остальных случаях} \end{cases}$$

Предусмотреть вывод номера расчетной формулы.



## Операции отношения

Операции отношения позволяют сравнить два значения и возвращают 1 (если выражение истинно) или 0 (если выражение ложно).

- == «равно» Например,  $x < 5$
- > «больше»  $5 > z$
- < «меньше»  $x < y$
- >= «больше или равно»  $5 >= z$
- <= «меньше или равно»  $5 <= z$
- != «не равно»  $a != b$

Любое значение, не равное нулю (даже отрицательное), трактуется как true, а нуль считается false;

## *Логические операции*

Логические операции, как правило, объединяют несколько операций отношения.

### **! (операция отрицания)**

Если операнд равен 0, то возвращает 1, иначе возвращает 0.

a	!a
true	false
false	true

```
int c = !3; // 0  
int d = !0; // 1
```

### **&& (конъюнкция, логическое «и»)**

Возвращает 1, если оба операнда не равны 0. Возвращает 0, если хотя бы один операнд равен 0.

a	b	a&&b
true	true	true
true	false	false
false	true	false
false	false	false

```
int c = -2 && 5;    // 1
int d = 0 && 7;     // 0
```

## || (дизъюнкция, логическое «или»)

Возвращает 1, если хотя бы один операнд не равен 0.  
Возвращает 0, если оба операнда равны 0.

a	b	a  b
true	true	true
true	false	true
false	true	true
false	false	false

```
int c = -2 || 5;    // 1
int d = 0 || 7;     // 1
int e = 0 || 0;     // 0
```

Операции `&&` и `||` действуют по сокращенной схеме. Если значения первого операнда достаточно, чтобы определить результат *операции*, второй *операнд* не вычисляется.

*Логические операции* выполняются слева направо.

Наивысший приоритет имеет унарная операция `!` (НЕ), затем операция `&&` (И – логическое умножение), а затем `||` (ИЛИ – логическое сложение).

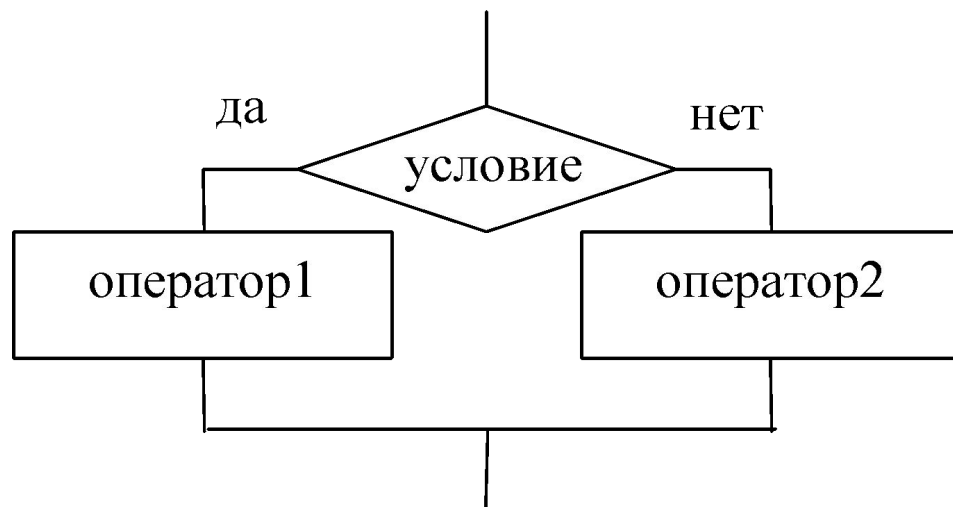
Операции сравнения имеют приоритет ниже, чем `!`, но выше, чем `&&` и `||`.

```
int d = -2 > 5 && 0 < 7 || 0 == 0;           // 1
```

## ***Условный оператор.***

Реализует алгоритмическую конструкцию **Ветвление** и изменяет порядок выполнения операторов в зависимости от истинности или ложности некоторого условия.

### а) Полная форма

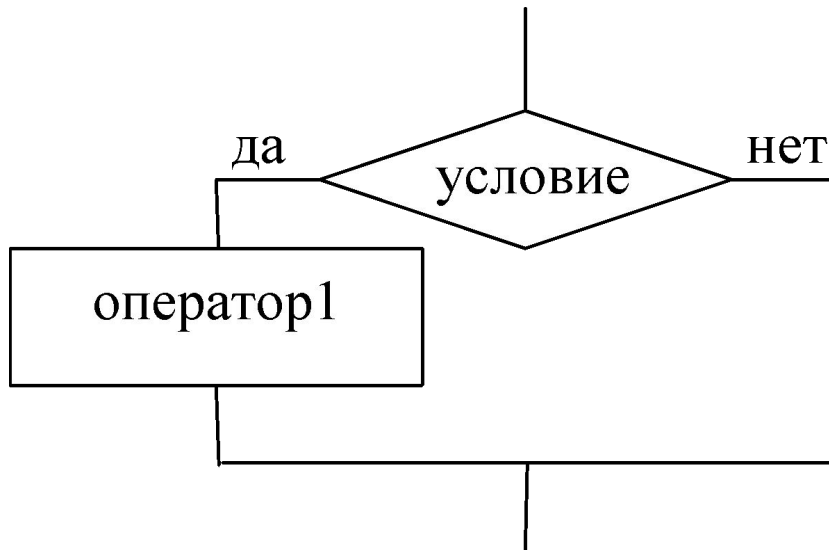


```
if ( выражение )  
    оператор_1;  
else  
    оператор_2;]
```

Сначала вычисляется значение выражения (логическое или арифметическое). Если оно не равно 0, то выполняется оператор 1. В противном случае выполняется оператор 2

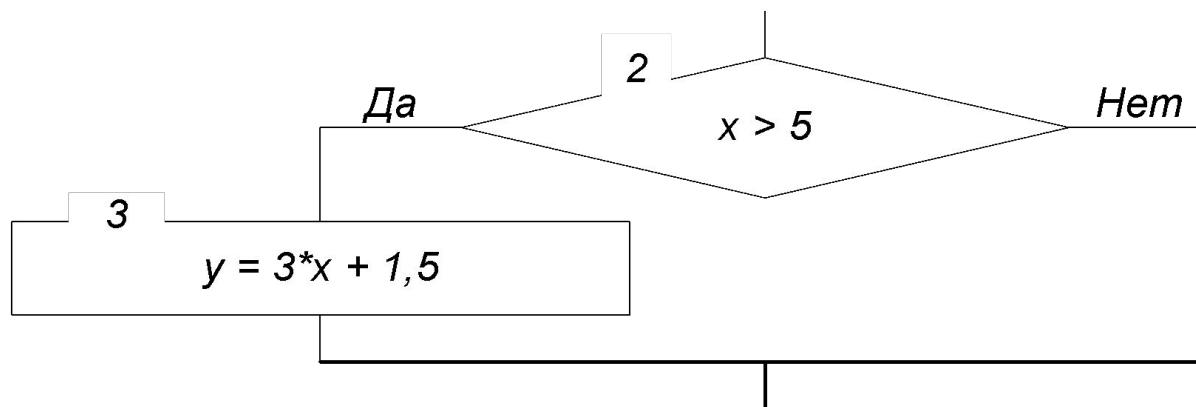


## а) Сокращенная форма

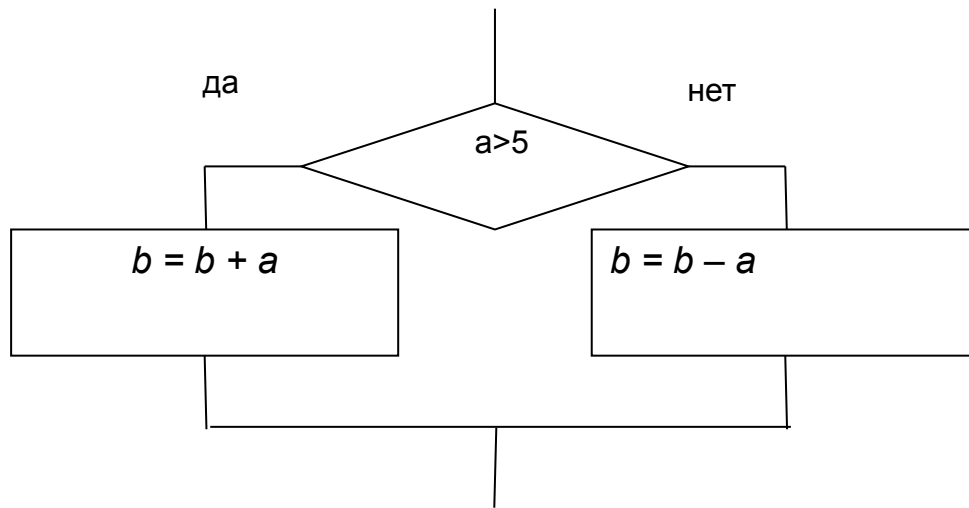


```
if (выражение)  
    оператор;
```

Например,

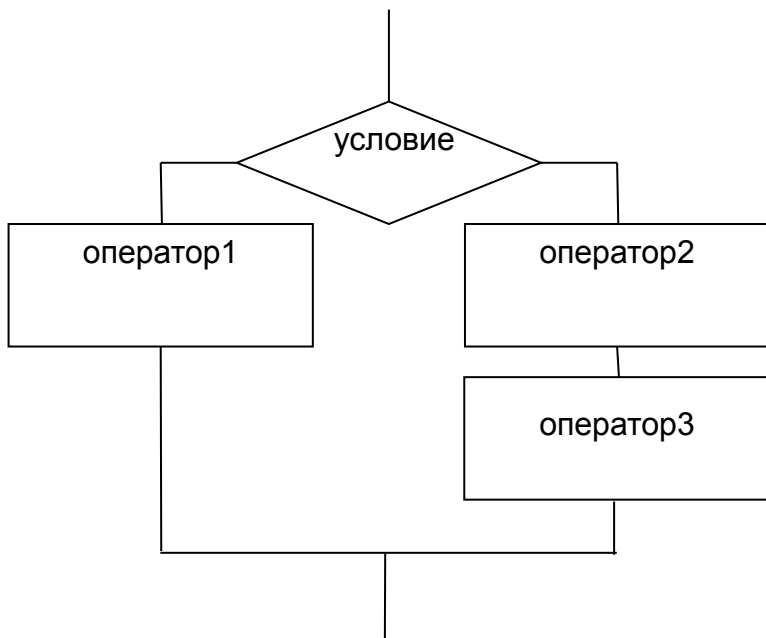


```
if (x > 5)  
    y = 3 * x + 1.5;
```

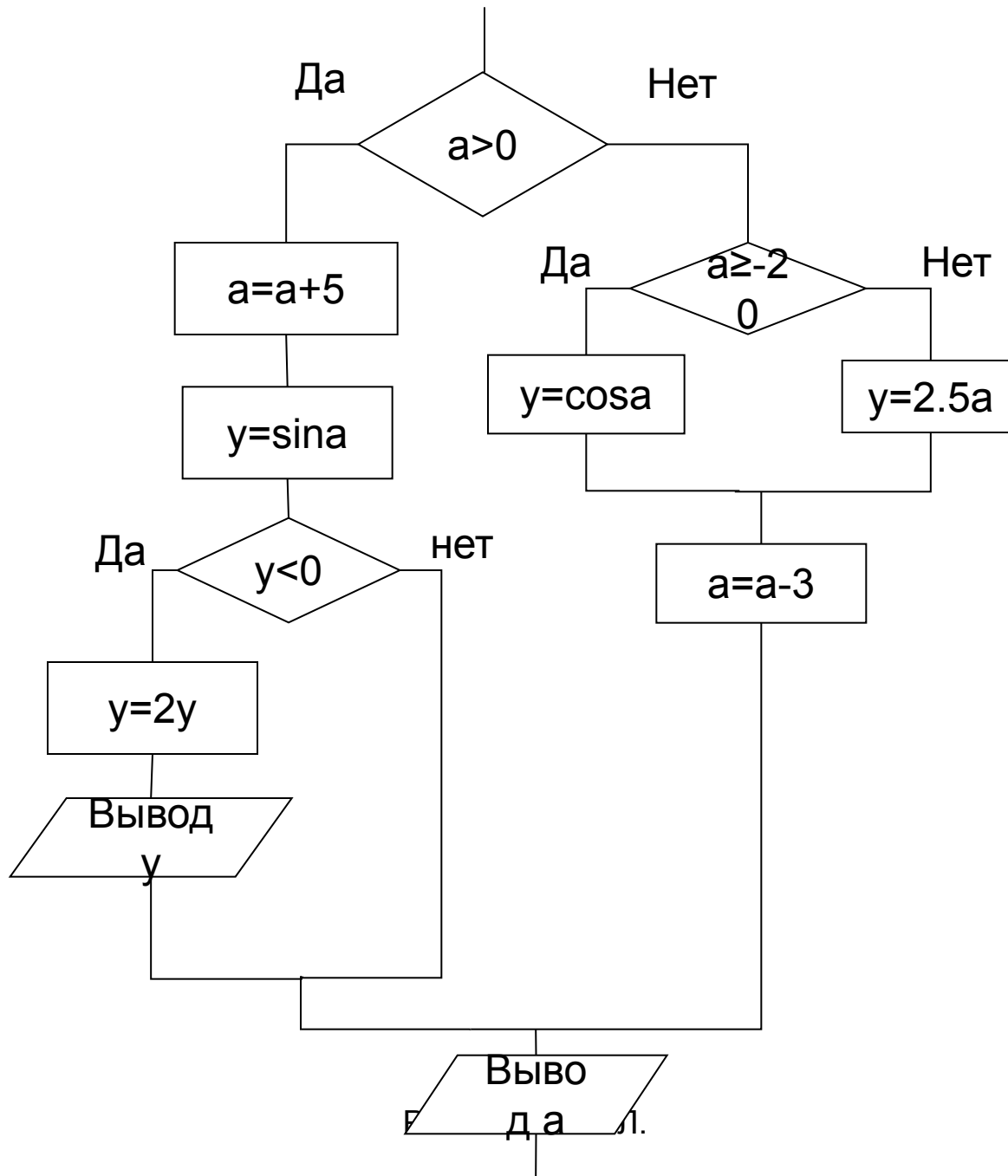


```
if (a > 5)
    b = b + a;
else
    b = b - a;
```

Если в конструкции **if** необходимо выполнить не одну, а несколько инструкций, то эти инструкции оформляются в **блок кода** с помощью фигурных скобок:



```
if (выражение)  
  оператор1 ;  
else  
  {  
    оператор2 ;  
    оператор3 ;  
  }
```



```
if (a > 0)
{
    a = a + 5;
    y = sin(a);
    if (y < 0)
    {
        y = 2 * y;
        printf("y=%f\n", y);
    }
}
else
{
    if (a >= -20)
        y = cos(a);
    else
        y = 2.5 * a;
    a = a - 3;
}
printf("a=%f\n", a);
```

не найдены. Стр: 35 Симв: 7 Столб: 10 Таблица

Из-за погрешности представления вещественных значений в памяти следует ее избегать, вместо этого лучше сравнивать модуль разности с некоторым малым числом.

```
float x, y; ...  
if ( x == y ) ...           // не рекомендуется!  
if ( abs(x - y) < 1e-7 ) ... // рекомендуется!
```

Значение величины, с которой сравнивается модуль разности, следует выбирать в зависимости от решаемой задачи и точности участвующих в выражении переменных.

Константа  $\pi$  определена в файле math.h следующим образом:

```
#define M_PI 3.14159265358979323846 /*  
pi*/
```

Но не каждый компилятор распознает(

*Составим программу для примера 1.*

```
#define _CRT_SECURE_NO_WARNINGS  
#include <stdio.h>  
#include <math.h>  
#include <windows.h>  
  
#define PI 3.14159265358979323846
```



```
void main(void)
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    double x,y;      // декларируем необходимые переменные
    unsigned int n;
    //ввод переменных с клавиатуры
    printf("Введите x: ");
    int value_count=scanf("%lf", &x);

    //проверка введенных данных
    if (!value_count)//value_count==0
        puts("Вы ввели что-то не то :-(");
    else
    {
```

```
//рассчитываем результат
```

```
if (x > 0 && x < PI)
```

```
{
```

```
    y = 2 * x * x / sin(x) - 2.5;
```

```
    n = 1;
```

```
}
```

```
else
```

```
    if (x <= 0)
```

```
    {
```

```
        y = 2 * pow(cos(x), 3) + fabs(x - 7.5);
```

```
        n = 2;
```

```
    }
```

```
    else
```

```
    {
```

```
        y = 4 * sqrt(x * x + sin(x) * sin(x));
```

```
        n = 3;
```

```
    }
```

```
//вывод результата на экран
```

```
    printf("Вы ввели x=%.3f ,    Результат: %.3f\n", x, y);  
    printf("Расчет произведен по формуле № %i\n", n);  
}  
system("pause");
```

Тесты для проверки:

$x = -1$	$y = 8.815$	$n = 2$
$x = 0$	$y = 9.5$	$n = 2$
$x = 1$	$y = -0.123$	$n = 1$
$x = 3.1415926$	$y = 12.566$	$n = 3$
$x = 5$	$y = 20.032$	$n = 3$

```
Н:\TANJA\ОАИП\Примеры\Project5\Debug\...
Введите x: -1
Вы ввели x=-1.000 ,    Результат: 8.815
Расчет произведен по формуле № 2
Для продолжения нажмите любую клавишу . . .
```

```
Н:\TANJA\ОАИП\Примеры\Project5\...
Введите x: 0
Вы ввели x=0.000 ,    Результат: 9.500
Расчет произведен по формуле № 2
Для продолжения нажмите любую клавишу . . .
```

```
Н:\TANJA\ОАИП\Примеры\Project5\Debug\Pr...
Введите x: 3.1415926
Вы ввели x=3.142 ,    Результат: 12.566
Расчет произведен по формуле № 3
Для продолжения нажмите любую клавишу . . .
```

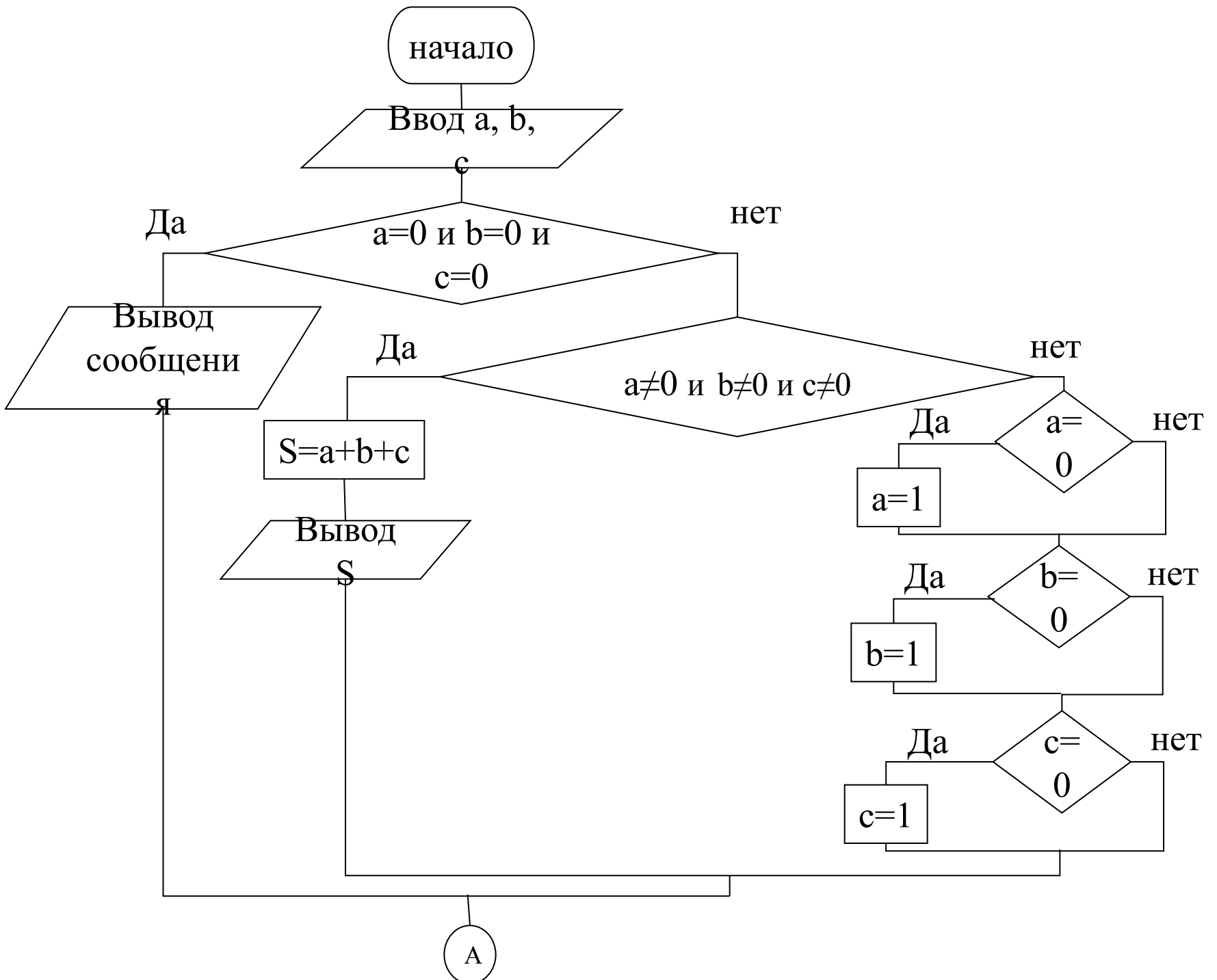
```
Н:\TANJA\ОАИП\Примеры\Project5\Debug\Projec...
Введите x: 3.14159265
Вы ввели x=3.142 ,    Результат: 540970000.000
Расчет произведен по формуле № 1
Для продолжения нажмите любую клавишу . . .
```

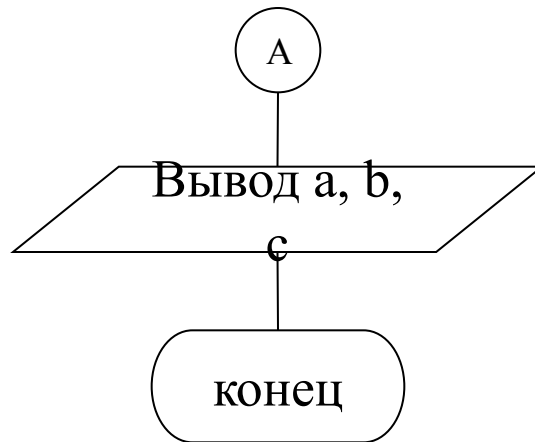
Что-то не то (

Нужно: `if (x > 0 && x < PI-1e-6 )`

## Пример 2.

Даны числа a, b, c. Если все они равны нулю, вывести об этом сообщение, если среди чисел есть нули, заменить их единицами, в противном случае найти и вывести сумму исходных чисел.





```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <windows.h>

void main(void) {

    SetConsoleCP(1251); //на ВВОД
    SetConsoleOutputCP(1251); //на ВЫВОД
    int a,b,c;
```

```
puts("Введите три целых числа через пробел или через ENTER");
int value_count = scanf("%d %d %d", &a,&b,&c);
//проверка введенных данных
if (value_count<3)
    puts("Вы ввели не все или что-то не то :-(");
else
{
    system("cls");//очистка экрана
    printf("Исходные числа:\na=%d b=%d c=%d\n", a, b, c);
    if (a==0 && b==0 && c==0)
    {
        puts("Все числа равны нулю");
    }
    else
        if (a!=0 && b != 0 && c != 0)
        {
            int sum=a+b+c;
            printf("Сумма чисел равна %d\n", sum);
        }
}
```

```

else
{
    if (a == 0) a = 1;
    if (b == 0) b = 1;
    if (c == 0) c = 1;
    printf("Числа после изменения:\na=%d b=%d c=%d\n", a, b, c);
}
}
system("pause");
}

```

```

Исходные числа:
a=0 b=0 c=0
Все числа равны нулю
Для продолжения нажмите любую клавишу . . .

```

```

Исходные числа:
a=1 b=3 c=5
Сумма чисел равна 9
Для продолжения нажмите любую клавишу . . .

```

```

Исходные числа:
a=5 b=0 c=6
Числа после изменения:
a=5 b=1 c=6
Для продолжения нажмите любую клавишу . . .

```

```

Введите три целых числа через пробел или через ENTER
sd 2 4
Вы ввели не все или что-то не то :-(.
Для продолжения нажмите любую клавишу . . .

```



## Условная операция «? :»

Синтаксис:

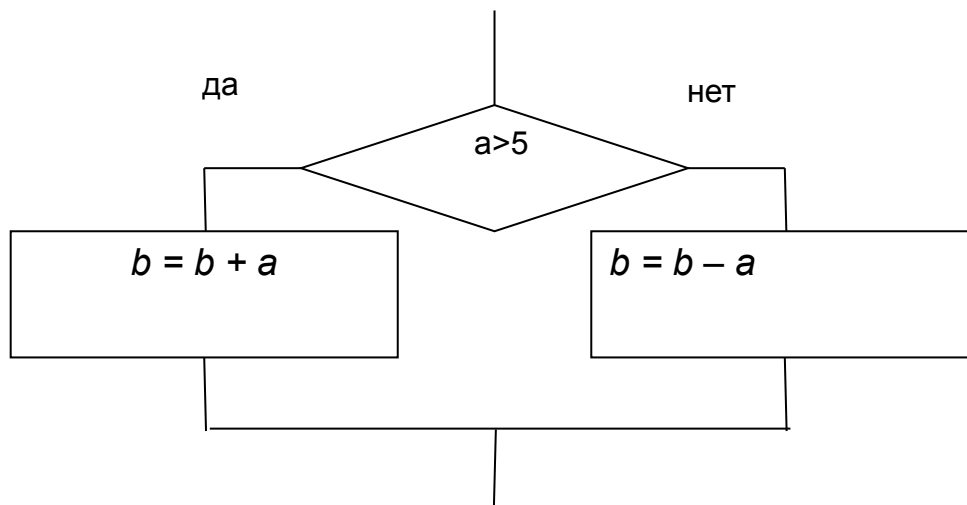
**Выражение 1 ? выражение 2 : выражение 3**

Порядок выполнения:

если *выражение 1* (условие) отлично от нуля (истинно), то результатом операции является значение *выражения 2*, в противном случае – значение *выражения 3*.

```
if (a > 5)
    b = b + a;
else
    b = b - a;
```

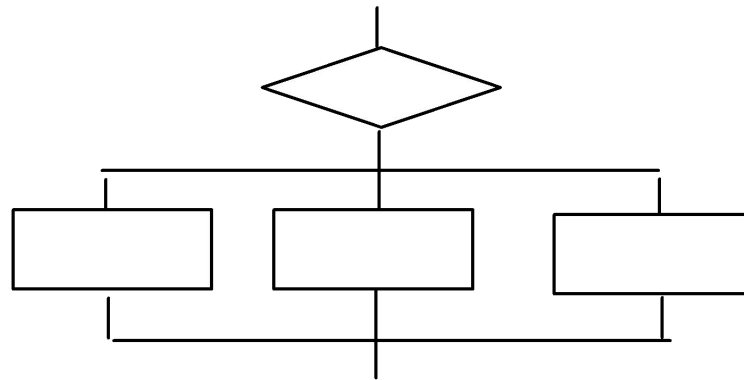
```
b = (a > 5) ? b+a : b-a;
```



## *Оператор выбора (переключатель)*

Оператор выбора позволяет выбрать один из нескольких возможных вариантов продолжения программы.

Реализует алгоритмическую структуру «**Выбор**»



***switch*** ( выражение )

```
{  
    case константа1: список операторов 1  
    case константа2: список операторов 2  
        ...  
    case константаN: список операторов N  
    default: список операторов N+1  
}
```

***switch*** ( выражение )

```
{  
    case константа1: список операторов 1  
    case константа2: список операторов 2  
        ...  
    case константаN: список операторов N  
}
```

Значение *выражения* должно быть целого или символьного типа.

Все константы должны иметь разные значения, но быть одного и того же типа.

Порядок выполнения:

Если значение выражения совпало с одной из перечисленных констант, то управление передается в соответствующую ветвь.

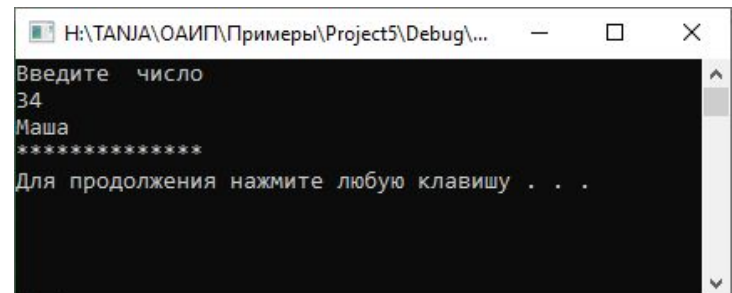
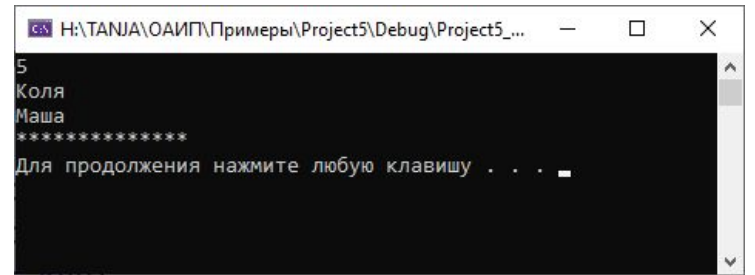
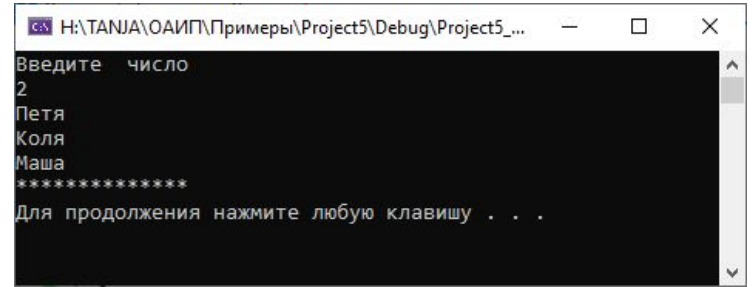
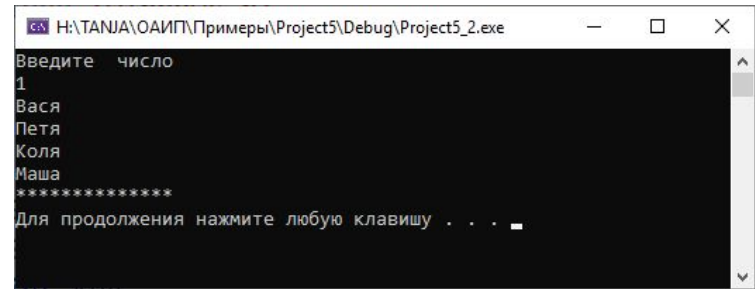
После этого, если выход из переключателя в данной ветви явно не указан, последовательно выполняются все остальные ветви.

В случае несовпадения значения выражения ни с одной из констант выбора происходит переход на метку *default* либо, при ее отсутствии, к оператору, следующему за оператором *switch*.

```

int key;
puts("Введите число");
scanf("%d", &key);
switch (key)
{
    case 1:puts("Вася");
    case 2:puts("Петя");
    case 5:puts("Коля");
    default:
        puts("Маша");
        break;
}
puts("*****");

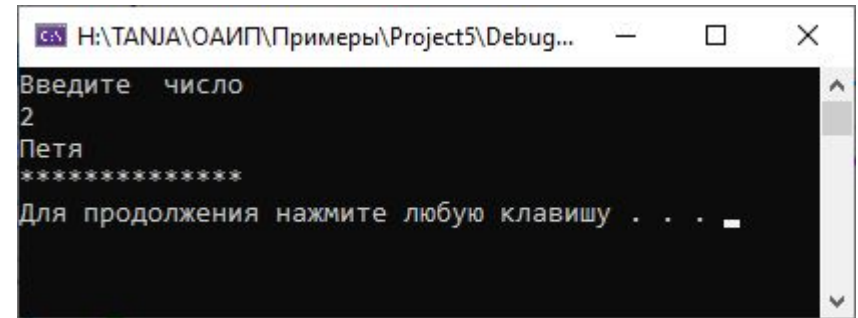
```



Для выхода из оператора *switch* следует использовать управляющий оператор *break* .

После операторов последней группы (после default или после последнего case) оператор break можно не указывать.

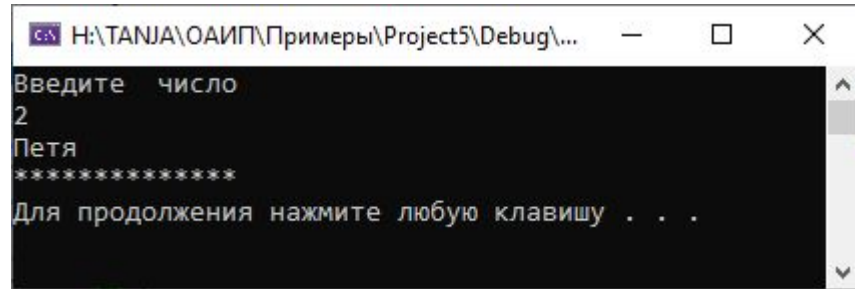
```
switch (key)
{
    case 1:puts("Вася"); break;
    case 2:puts("Петя"); break;
    case 5:puts("Коля"); break;
    default:
        puts("Маша");
}
puts("*****");
```



```
Н:\TANJA\ОАИП\Примеры\Project5\Debug...
Введите число
2
Петя
*****
Для продолжения нажмите любую клавишу . . .
```

Оператор break не является частью конструкции выбора. Он может использоваться и в других контекстах (например, для экстренного выхода из цикла)

```
switch (key)
{
case 1:puts("Вася"); break;
case 2:
case 3:
case 4:puts("Петя"); break;
case 5:puts("Коля"); break;
default:puts("Маша");
}
```



```
cmd H:\TANJA\ОАИП\Примеры\Project5\Debug\...
Введите число
2
Петя
*****
Для продолжения нажмите любую клавишу . . .
```

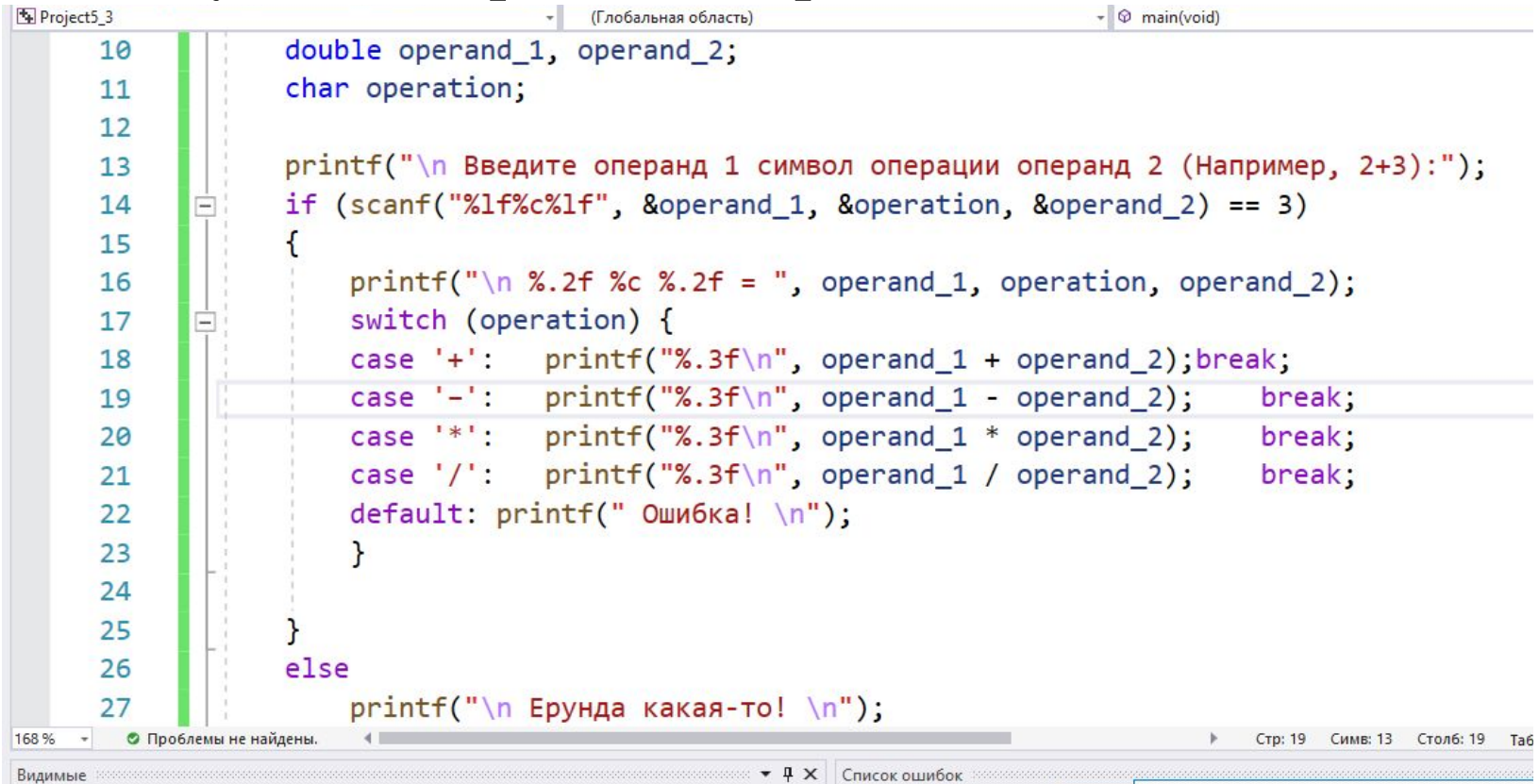
**Пример 3.** Требуется вывести на экран оценку, введенную цифрой, прописью.

```
int rezult;
printf("Введите оценку ");
if (scanf("%d", &rezult))
    switch (rezult)
    {
        case 5: printf("Оценка – отлично.\n"); break;
        case 4: printf("Оценка – хорошо.\n"); break;
        case 3: printf("Оценка – удовлетворительно.\n"); break;
        case 2: printf("Оценка – неудовлетворительно.\n"); break;
        case 1: printf("Оценка – полный отстой.\n"); break;
        default: printf("Неверное значение.\n");
    }
else
    puts("Это бред какой-то");
```



**Пример 4.** Реализовать простейший калькулятор на четыре действия с контролем правильности ввода символа нужной операции.

Ввод данных осуществляется следующим образом: **операнд 1**  
**символ нужной операции** **операнд 2.**



```
10 double operand_1, operand_2;
11 char operation;
12
13 printf("\n Введите операнд 1 символ операции операнд 2 (Например, 2+3):");
14 if (scanf("%lf%c%lf", &operand_1, &operation, &operand_2) == 3)
15 {
16     printf("\n %.2f %c %.2f = ", operand_1, operation, operand_2);
17     switch (operation) {
18     case '+': printf("%.3f\n", operand_1 + operand_2); break;
19     case '-': printf("%.3f\n", operand_1 - operand_2); break;
20     case '*': printf("%.3f\n", operand_1 * operand_2); break;
21     case '/': printf("%.3f\n", operand_1 / operand_2); break;
22     default: printf(" Ошибка! \n");
23     }
24
25 }
26 else
27     printf("\n Ерунда какая-то! \n");
```

```
с:\ H:\TANJA\ОАИП\Примеры\Project5\Debug\Project5_3.exe
Введите операнд 1 символ операции операнд 2 (Например, 2+3):4+5
4.00 + 5.00 = 9.000
Для продолжения нажмите любую клавишу . . .
```

```
с:\ H:\TANJA\ОАИП\Примеры\Project5\Debug\Project5_3.exe
Введите операнд 1 символ операции операнд 2 (Например, 2+3):7.5-2.123
7.50 - 2.12 = 5.377
Для продолжения нажмите любую клавишу . . .
```

```
с:\ H:\TANJA\ОАИП\Примеры\Project5\Debug\Project5_3.exe
Введите операнд 1 символ операции операнд 2 (Например, 2+3):5^9
5.00 ^ 9.00 = Ошибка!
Для продолжения нажмите любую клавишу . . .
```

```
с:\ H:\TANJA\ОАИП\Примеры\Project5\Debug\Project5_3.exe
Введите операнд 1 символ операции операнд 2 (Например, 2+3):7/0
7.00 / 0.00 = inf
Для продолжения нажмите любую клавишу . . .
```

```
с:\ H:\TANJA\ОАИП\Примеры\Project5\Debug\Project5_3.exe
Введите операнд 1 символ операции операнд 2 (Например, 2+3):vasja
Ерунда какая-то!
Для продолжения нажмите любую клавишу . . .
```

**Пример 5.** Программа выводит на экран меню, в котором пользователь может выбрать одно из следующих действий введя соответствующее число: подсчитать сумму двух чисел, подсчитать разность двух чисел, выйти из программы.

```
int n, a, b;  
printf("Меню:\n");  
printf("1 - подсчет суммы двух чисел\n");  
printf("2 - подсчет разности двух чисел\n");  
printf("3 - выход из программы\n");  
printf("Ваш выбор ? \n ");  
scanf("%d", &n);
```

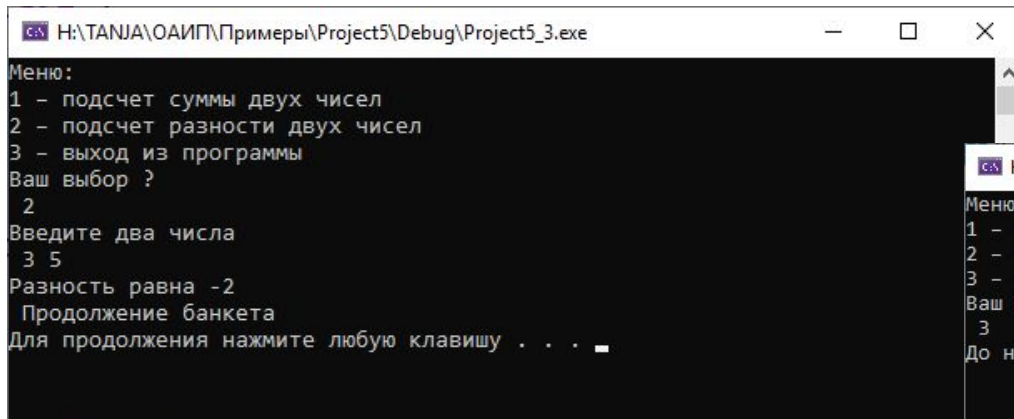
```
switch (n)
{
case 1:
    printf("Введите два числа\n ");
    scanf("%d %d", &a, &b);
    printf("Сумма равна % d\n ", a + b);
    break;
case 2:
    printf("Введите два числа\n ");
    scanf("%d %d", &a, &b);
    printf("Разность равна % d\n ", a - b);
    break;
```

case 3:

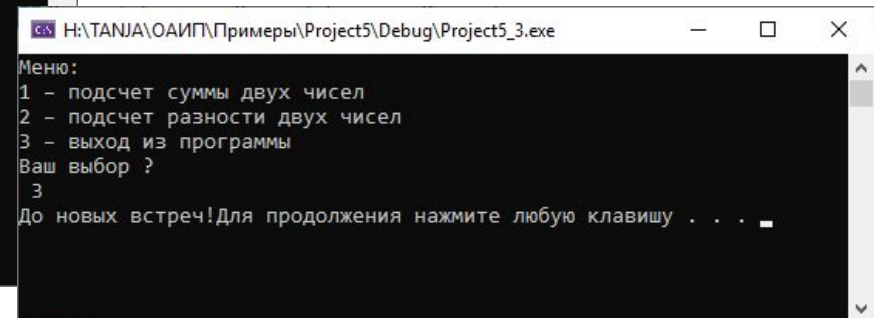
```
printf("До новых встреч!");  
system("pause");  
return; //выход  
break;
```

default:

```
printf("Неверное значение.");  
}  
puts("Продолжение банкета");  
system("pause");
```



```
cs H:\TANJA\ОАИП\Примеры\Project5\Debug\Project5_3.exe  
Меню:  
1 - подсчет суммы двух чисел  
2 - подсчет разности двух чисел  
3 - выход из программы  
Ваш выбор ?  
2  
Введите два числа  
3 5  
Разность равна -2  
Продолжение банкета  
Для продолжения нажмите любую клавишу . . .
```



```
cs H:\TANJA\ОАИП\Примеры\Project5\Debug\Project5_3.exe  
Меню:  
1 - подсчет суммы двух чисел  
2 - подсчет разности двух чисел  
3 - выход из программы  
Ваш выбор ?  
3  
До новых встреч!  
Для продолжения нажмите любую клавишу . . .
```

**Пример 6.** Программа-конвертор валют. Пользователь вводит сумму в белорусских рублях, а программа рассчитывает соответствующую сумму в долларах, евро или российских рублях. Валюту выбирает пользователь.

```
#include <stdio.h>
#include <locale.h>
#define DOLLAR 2.62
#define EURO 3.07
#define RUB 0.034
```

```
void main()
{
    setlocale(LC_ALL, "rus");
    float sum;
    puts("Введите сумму в бел. рублях: ");
    scanf_s("%f", &sum);
    puts("Введите D, если хотите получить сумму в долларах  
США");
    puts("Введите E, если хотите получить сумму в евро");
    puts("Введите R, если хотите получить сумму в рос.  
рублях");
    puts("Ваш выбор: ");
}
```

```
char v=getch();
switch (v)
{
case 'D':
case 'd':
    printf("Вам следует получить %.2f $\n", sum / DOLLAR);
    break;
case 'E':
case 'e':
    printf("Вам следует получить %.2f EUR\n", sum / EURO);
    break;
case 'R':
case 'r':
    printf("Вам следует получить %.2f RUB\n", sum / RUB);
    break;
default: puts("Неверный выбор\n");
}
system("pause");
}
```



```
cs H:\TANJA\ОАИП\Примеры\Project5\Debug\Project5_4.exe
Введите сумму в бел. рублях:
345
Введите D, если хотите получить сумму в долларах США
Введите E, если хотите получить сумму в евро
Введите R, если хотите получить сумму в рос. рублях
Ваш выбор:
Вам следует получить 112,38 EUR
Для продолжения нажмите любую клавишу . . .
```

```
cs H:\TANJA\ОАИП\Примеры\Project5\Debug\Project5_4.exe
Введите сумму в бел. рублях:
120
Введите D, если хотите получить сумму в долларах США
Введите E, если хотите получить сумму в евро
Введите R, если хотите получить сумму в рос. рублях
Ваш выбор:
Вам следует получить 45,80 $
Для продолжения нажмите любую клавишу . . .
```