

## Переменные. Типы данных

*Переменная* - объект программы, занимающий в общем случае несколько ячеек памяти, призванный хранить данные. Переменная, обладает именем, размером и рядом других атрибутов (таких как видимость, время существования и т.д.).

При объявлении переменной для нее резервируется некоторая область памяти, размер которой зависит от конкретного типа переменной. Здесь следует обратить внимание на то, что размер одного и того же типа данных может отличаться на компьютерах разных платформ, а также может зависеть от используемой операционной системы. Поэтому при объявлении той или иной переменной нужно четко представлять, сколько байт она будет занимать в памяти ЭВМ, чтобы избежать проблем, связанных с переполнением и неправильной интерпретацией данных.

Ниже приведен перечень базовых типов переменных и их размер в байтах. Следует учесть, что размер, указанный в табл. для каждого типа, должен быть проверен для конкретного ПК.

Имя объявления	Тип	Примерный диапазон значений
Char	Символьный	-128..127
unsigned char	Беззнаковый символьный	0..255
signed char	Знаковый символьный (такой же как char)	-128..127
int	Целый	-32868..32676
unsigned int	Беззнаковый целый	0..65535
signed int	Знаковый целый (такой же как int)	-32768..32767
short int	Короткий целый	-32768..32767
unsigned short int	Беззнаковый короткий целый	0..65535
signed short int	Знаковый короткий целый (такой же как short int)	-32768..32767
Long	Длинный целый	-2147483648..2147483847
long int	Длинный целый (такой же как long)	-2147483648..2147483847
signed long int	Знаковый длинный целый (такой же как long int)	-2147483648..2147483847
unsigned long int	Беззнаковый длинный целый	0..4294967295
Float	Длинный с плавающей точкой	-3.4E+38..3.4E+38
long double	Длинный с плавающей точкой двойной точности	-1.7E+308..1.7E+308

Объявление переменной начинается с ключевого слова, определяющего его тип, за которым следует собственно имя переменной и (необязательно) инициализация - присвоение начального значения.

Одно ключевое слово позволяет объявить несколько переменных одного и того же типа. При этом они следуют друг за другом через запятую (,). Заканчивается объявление символом точка с запятой (;).

Имя переменной (идентификатор) не должно превышать 256 символов (разные компиляторы накладывают свои ограничения на количество распознаваемых символов в идентификаторе). При этом важно учитывать регистр букв (Abe и abc - не одно и то же)! Конечно, имя должно быть достаточно информативным, однако не следует использовать слишком длинные имена, так как это приводит к опискам.

```
int a = 0, A = 1;
float aGe = 17.5;
double PointX;
bool bTheLightIsOn=false;
char LETTER = 'Z';
```

## Константы

*Константы*, так же как и переменные, представляют собой область памяти для хранения данных, с тем лишь отличием, что значение, присвоенное константе первоначально, не может быть изменено на протяжении выполнения всей программы. Константы бывают *литеральными* и *типизованными*, причем литеральные константы делятся на *символьные*, *строковые*, *целые* и *вещественные*.

*Символьные константы* представляются отдельным символом, заключенным в одинарные кавычки (апострофы): 'e', '@', '<'.

*Строковые константы* - это последовательность символов, заключенная в двойные кавычки: "Это пример не самой длинной строковой константы!".

*Целые константы* бывают следующих форматов:

- десятичные;
- восьмеричные;
- шестнадцатеричные.

*Десятичные* могут быть представлены как последовательность цифр, начинающаяся не с нуля, например: 123; 2384.

*Восьмеричные* константы - последовательность восьмеричных цифр (от 0 до 7), начинающаяся с нуля, например: 034; 047.

*Шестнадцатеричный* формат констант начинается с символов 0x или 0X с последующими шестнадцатеричными цифрами (0...9, A...F), например: 0xF4; 0X5D. Буквенные символы при этом могут быть представлены как в нижнем, так и в верхнем регистре.

Длинные целые константы, используемые в переменных типа long, определяются латинской буквой l или L сразу после константы без пробела: 36L, 012L, 0x52L.

*Вещественные константы* - числа с плавающей запятой могут быть записаны в десятичном формате (24.58; 13.0; 71) или в экспоненциальной форме (1e4; 5e+2; 2.2e-5, при этом в мантиссе может пропускаться целая или дробная часть: .2e4).

```
const double pi = 3.1415;  
const int Radius = 3;
```

## Преобразования типов

В C++ существует явное и неявное *преобразование типов*.

В общем случае *неявное преобразование типов* сводится к участию в выражении переменных разного типа (так называемая арифметика смешанных типов). Если подобная операция осуществляется над переменными базовых типов, она может повлечь за собой ошибки: в случае, например, если результат занимает в памяти больше места, чем отведено под принимающую переменную, неизбежна потеря значащих разрядов.

Для *явного преобразования* переменной одного типа в другой перед именем переменной в скобках указывается присваиваемый ей новый тип:

```
#include <iostream>  
using namespace std;  
  
int main()  
{  
    int Integer = 54;  
    float Floating = 15.854;  
    Integer = (int) Floating; // явное преобразование типов  
    cout << "New integer: ";
```

```
cout << Integer << '\n';  
return 0;  
}
```

В приведенном листинге после объявления соответствующих переменных (целочисленной Integer и вещественной Floating) производится явное преобразование типа с плавающей запятой (Floating) к целочисленному (Integer).

Пример неявного преобразования:

```
#include <iostream>  
using namespace std;  
  
int main()  
{  
int Integer = 0;  
float Floating = 15.854;  
Integer = Floating; // неявное преобразование типов  
cout << "New integer: ";  
cout << Integer << '\n';  
return 0;  
}
```

В отличие от предыдущего варианта программы, в данном случае после объявления и инициализации переменных осуществляется присваивание значения переменной с плавающей Floating целочисленной переменной Integer.

Результат работы обеих программ выглядит следующим образом:

New integer: 15

То есть произошло отсечение дробной части переменной Floating.

## Структура выбора if (ЕСЛИ)

Структура выбора используется для выбора среди альтернативных путей обработки информации. Например, предположим, что проходной балл на экзамене — 60. Предложение на псевдокоде

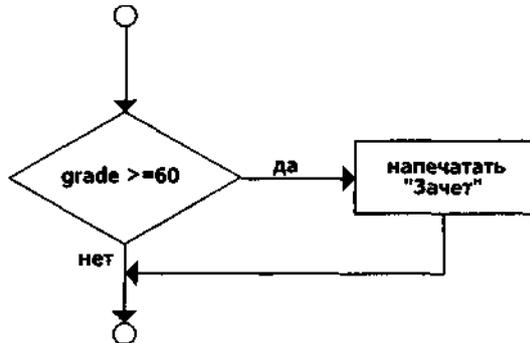
*ЕСЛИ оценка студента больше или равна 60  
Напечатать "Зачет"*

определяет, истинно или ложно условие «оценка студента больше или равна 60». Если это условие истинно, то печатается «Зачет» и «выполняется» следующее по порядку предложение псевдокода (напомним, что псевдокод — это в действительности не язык программирования). Если же данное условие ложно, то предложение печати игнорируется и сразу выполняется следующее по порядку предложение псевдокода. Заметьте, что вторая строка структуры выбора напечатана с отступом. Подобные отступы не обязательны, но их настоятельно рекомендуется делать, так как они подчеркивают структуры структурированных программ. Компилятор C++ игнорирует такие символы-разделители, как пробелы, символы табуляции, перевода строки, используемые для структурированного расположения текста и его вертикальной разрядки.

Соответствующий приведенному псевдокоду оператор if может быть записан на языке C++ как

```
if (grade >= 60)
    cout << "Зачет" << endl;
```

Блок-схема:



### Структура выбора if/else (ЕСЛИ-ИНАЧЕ)

Структура выбора if выполняет указанное в ней действие только, если условие истинно, и пропускает его в ином случае. Структура выбора if/else позволяет программисту определить различные действия, которые должны выполняться в случаях, если условие истинно или ложно. Например, предложение псевдокода

```
ЕСЛИ оценка студента больше или равна 60
    Напечатать "Зачет" ИНАЧЕ
    Напечатать "Незачет"
```

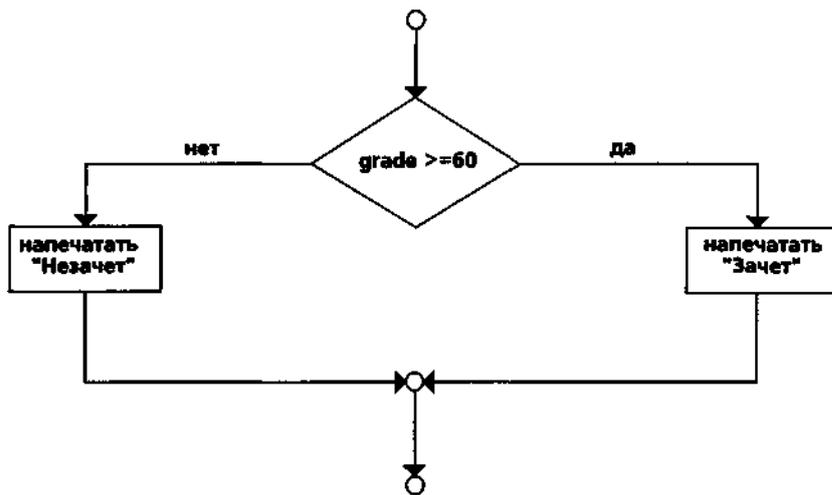
печатает *Зачет*, если оценка студента больше или равна 60, и печатает *Незачет*, если оценка меньше 60. В обоих случаях после печати «выполняется» следующее по порядку предложение псевдокода. Отметим, что тело ИНАЧЕ также записывается с отступом.

Какой бы стиль отступов вы ни приняли, необходимо строго придерживаться его во всех программах. Трудно читать программы, в которых в отступах не поддерживается постоянное количество пробелов.

Рассмотренный псевдокод структуры if/else может быть записан на C++ следующим образом:

```
if (grade >= 60)
    cout << "Зачет" << endl;
    else cout << "Незачет" << endl;
```

Блок-схема:



Для множественного выбора можно использовать вложенные структуры if/else, помещая одну структуру if/else внутрь другой. Например, следующее предложение псевдокода будет печатать A при экзаменационной оценке больше или равной 90, B — при оценке, лежащей в диапазоне от 80 до 89, C — при оценке в диапазоне 70-79, D — при оценке в пределах 60-69 и F — при других оценках.

*ЕСЛИ оценка студента больше или равна 90  
 Напечатать "A" ИНАЧЕ  
 ЕСЛИ оценка студента больше или равна 80 Напечатать "B"  
 ИНАЧЕ  
 ЕСЛИ оценка студента больше или, равна 70  
 Напечатать "C" ИНАЧЕ  
 ЕСЛИ оценка студента больше или равна 60  
 Напечатать "D" ИНАЧЕ  
 Напечатать "F"*

Этот псевдокод может быть записан на языке C++ в виде:

```

if (grade >= 90)
  cout << "A" << endl; else
  if (grade >= 80)
    cout << "B" << endl; else
    if (grade >= 70)
      cout << "C" << endl; else
      if (grade >= 60)
        cout << "D" << endl; else
        cout << "F" << endl;
  
```

Структура выбора if обычно предполагает наличие в своем теле только одного оператора. Чтобы включить несколько операторов в тело структуры, заключите их в фигурные скобки: «{» и «}». Множество операторов, заключенных в фигурные скобки, называется *составным оператором*.

```

if (grade >= 60)
  cout << "Зачет." << endl;
else {
  cout << "Незачет." << endl;
  cout << "Вы должны изучить этот курс снова." << endl;
}
  
```

}

### Операции присваивания

В C++ имеется несколько операций присваивания, позволяющих сокращать запись присваиваемых выражений. Например, оператор

```
c = c + 3;
```

может быть сокращен применением *составной операции сложения* += :

```
c += 3;
```

Операция += прибавляет значение выражения, записанного справа от операции, к величине переменной, указанной слева, и сохраняет результат в этой переменной. Любой оператор вида

```
переменная = переменная операция выражение;
```

где операция — одна из бинарных операций +, -, \*, / или % (или иные операции, которые будут рассмотрены позднее), может быть записан в виде

```
переменная операция = выражение;
```

Таким образом, присваивание `c += 3` добавляет 3 к `c`. Рис. 2.12 показывает арифметические операции присваивания, примеры выражений с этими операциями и их расширенное толкование.

Операция присваивания	Пример	Пояснение	Результат присваивания
<b>Предположим: int c = 3, d = 5, e = 4, f = 6, g = 12;</b>			
+=	<b>c += 7</b>	<b>c = c + 7</b>	<b>c = 10</b>
-=	<b>d -= 4</b>	<b>d = d - 4</b>	<b>d = 1</b>
*=	<b>e *= 5</b>	<b>e = e * 5</b>	<b>e = 20</b>
/=	<b>f /= 3</b>	<b>f = f / 3</b>	<b>f = 2</b>
%=	<b>g %= 9</b>	<b>g = g % 9</b>	<b>g = 3</b>

### Операции инкремента и декремента

В C++ имеется унарная *операция инкремента* ++ (увеличение на 1) и унарная *операция декремента* - (уменьшение на 1). Если переменная `c` должна быть увеличена на 1, лучше применить оператор ++, чем выражения `c=c+1` или `c+=1`. Если операция инкремента или декремента помещена перед переменной, говорят о *префиксной форме записи* инкремента или декремента. Если операция инкремента или декремента записана после переменной, то говорят о *постфиксной форме записи*. При префиксной форме переменная сначала увеличивается или уменьшается на единицу, а затем это ее новое значение используется в том выражении, в котором она встретилась. При постфиксной форме в выражении используется текущее значение переменной, и только после этого ее значение увеличивается или уменьшается на единицу.

Программа демонстрирует различие между префиксной и постфиксной формами операции инкремента ++. Постфиксная форма записи операции инкремента вызывает увеличение переменной `c` после того, как она использована в операторе вывода. При префиксной форме переменная `c` изменяется до того, как будет использована в операторе вывода.

Программа выводит на экран значения `c` до и после применения операции ++. Операция -- действует аналогично.

Операция	Название операции	Пример выраже	Пояснение
+	префиксная форма инкремента	++a	Величина <i>a</i> увеличивается на 1 и это новое значение <i>a</i> используется в выражении, в котором оно встретилось
+	постфиксная форма инкремента	a++	В выражении используется текущее значение <i>a</i> , а затем величина, <i>a</i> увеличивается на 1
-	префиксная форма декремента	--b	Величина <i>b</i> уменьшается на 1 и это новое значение <i>b</i> используется в выражении, в котором оно встретилось
-	постфиксная форма декремента	b--	В выражении используется текущее значение <i>b</i> , а затем величина <i>b</i> уменьшается на 1

Различие между префиксой и постфиксной формами операции инкремента:

```
int Main()
{
int c;
c=5;
cout << c << endl;
cout << c++ << endl;
cout << c << endl;

// Постфиксная форма инкремента

c=5;
cout << c << endl;
cout << ++c << endl;      // Префиксная форма инкремента
cout << c << endl;
return 0;  // успешное окончание

5
5
6
5
6
6
```

Три оператора присваивания:

```
passes = passes + 1;
failures = failures + 1;
student = student + 1;
```

могут быть записаны более кратко в виде:

```
passes += 1;
failures += + 1;
student += 1;
```

или с использованием операции инкремента в префиксной форме:

```
++passes;  
++failures;  
++student;
```

или в постфиксной форме:

```
passes++;  
failures++;  
student++;
```

Важно отметить, что когда инкремент или декремент переменной осуществляется в виде отдельного оператора, то префиксная и постфиксная формы приводят к одинаковому результату. И только если переменная появляется в контексте более сложного выражения, тогда префиксная и постфиксная формы приводят к разным результатам.

### Математические функции:

Функция	Описание
sqrt(x)	корень квадратный из x
exp(x)	экспоненциальная функция $e^x$
log(x)	логарифм натуральный x (по основанию e)
log10(x)	логарифм десятичный x (по основанию 10)
fabs(x)	абсолютное значение x
ceil(x)	округление x до наименьшего целого, не меньшего чем x
floor(x)	округление x до наибольшего целого, не большего чем x
pow{x, y}	x в степени y
fmod(x, y)	остаток от x/y, как число с плавающей точкой
sin(x)	синус x (x в радианах)
cos(x)	косинус x (x в радианах)
tan(x)	тангенс x (x в радианах)

```
#include <iostream>  
#include <math.h>  
using namespace std;
```

```
int main()  
{  
    float x=sin(10.7);  
    cout << x << endl;  
}
```

### Упражнения:

1. Записать следующие формулы:

- a)  $\frac{-b + \sqrt{b^2 - 4ac}}{2a}$ ; ▲
- b)  $\left\{ \left[ (ax - b)x + c \right] x + d \right\} x - e$ ; ▲
- c)  $\frac{1}{3} \left( \left( \frac{\sin^2(x) - \cos^2(x)}{\sin \frac{x+y}{2}} - e^{|\cos x| + \sin x} \right) \ln x - \sqrt{x-1} \right)$ ; ▲
- d)  $\left( 1 + \frac{x}{2!} + \frac{y}{3!} + \frac{z}{4!} \right)$ ; ▲
- e)  $\frac{1 + \sin^2(x+y)}{2 + \sqrt{\frac{x-2x}{(1+x^2y^2)}}} + x$ ; ▲
- f)  $\sqrt{p(p-a)^2(p-b)(p-c)^2}$ ; ▲
- g)  $a + \frac{1}{b+c+\frac{1}{d}} - \frac{a}{2d + \frac{c}{2b + \frac{1}{f}}}$ ; ▲
- h)  $\operatorname{tg}^2 \left( \frac{x^2}{2} - 1 \right) + \frac{2 \cos(x - \pi/6)}{1/2 + \sin^2 y}$ ; ★
- i)  $\frac{xyz - 3.3 \sqrt{x + \sqrt[4]{y}}}{10^7 + \sqrt{\operatorname{tg} 4!}}$ ; ★
- j)  $e^{|x-y|} + \ln(1+e) \log_2 \operatorname{tg} 2$ . ★

2. Записать следующие формулы:

- a)  $2 \log_2 \frac{x}{15}$ ;
- b)  $\sin 15^\circ \cos 15^\circ$ ;
- c)  $x^8$ ;
- d)  $x^{250}$ ;
- e)  $10^3 + \frac{\beta}{x_2 - y\delta}$ ;

3. Найти сумму  $n$  членов арифметической прогрессии, первый член которой равен  $a$ , разность равна  $d$ . ▲
4. Вычислить значение первой производной функции  $x^n$  в заданной точке  $a$  при заданном значении  $n$ . ▲
5. Ввести положительное число  $a$ . Вычислить ▲
- площадь равностороннего треугольника со стороной  $a$ ;
  - площадь квадрата со стороной  $a$ ;
  - площадь круга, радиус которого равен  $a$ .
6. Вычислить длину окружности, площадь круга, объем шара заданного радиуса. ▲
7. Даны числа  $a, b, \gamma$ . Найти площадь треугольника, две стороны которого равны  $a$  и  $b$ , а угол между этими сторонами равен  $\gamma$ . Считать, что  $\gamma$  - это ★
- радианная мера угла;
  - градусная мера угла.
8. Дано натуральное число  $n$ , состоящее из 6 цифр. Определить число сотен и тысяч в нем. ★

9. Ввести два вещественных числа. Напечатать коэффициенты приведенного квадратного уравнения, корнями которого являются эти числа. ▲
10. Дано число  $f$  – угол в градусах. Определить смежный к нему угол в радианах. ▲
11. В бригаде, работающей на уборке сена, имеется  $n$  косилок. Первая из них работала  $m$  часов, а каждая следующая на 10 мин больше, чем предыдущая. Сколько часов работала вся бригада. ▲
12. Даны целые числа  $m, n$  (часы, минуты),  $0 \leq m \leq 11, 0 \leq n \leq 59$ , определяющие время суток. Определить наименьшее время (число полных минут), которое должно пройти до того момента, когда часовая и минутная стрелка на циферблате ★
  - а) совпадут;
  - б) расположатся перпендикулярно друг другу.
13. Сколько времени в минутах затратит школьник на дорогу от школы до стадиона, если известна длина этого расстояния  $S$  км и средняя скорость движения школьника  $V$  км/ч.
14. В квадрат вписана окружность (Рис. 3.1). Определить площадь заштрихованной части фигуры, если известна длина стороны квадрата.
15. В квадрат вписана окружность (Рис. 3.1). Определить площадь заштрихованной части фигуры, если известен радиус окружности.
16. В квадрат вписана окружность (Рис. 3.2). Определить площадь заштрихованной части фигуры, если известна длина стороны квадрата.
17. В квадрат вписана окружность (Рис. 3.3). Определить площадь заштрихованной части фигуры, если известна длина стороны квадрата.

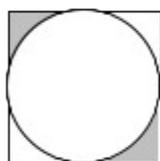


Рис. 3.1

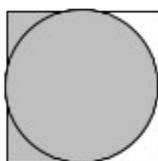


Рис. 3.2

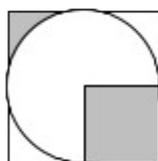


Рис. 3.3

18. Определить, есть ли среди цифр заданного трехзначного числа повторяющиеся. ▲
19. Определить, равна ли сумма крайних цифр заданного четырехзначного числа сумме его средних цифр. ▲
20. Пусть  $(k, l), (m, n)$  – поля шахматной доски:  $k, m$  – номера горизонтали;  $l, n$  – номера вертикали;  $(1 \leq k, l, m, n \leq 8)$ . Определить, ✦
- можно ли с поля  $(k, l)$  попасть на поле  $(m, n)$  одним ходом пешки;
  - можно ли с поля  $(k, l)$  попасть на поле  $(m, n)$  одним ходом ладьи;
  - можно ли с поля  $(k, l)$  попасть на поле  $(m, n)$  одним ходом слона;
  - можно ли с поля  $(k, l)$  попасть на поле  $(m, n)$  одним ходом ферзя.
21. Дано вещественное число  $a$ . Вычислить  $f(a)$ , где  $f$  – периодическая функция с периодом 2, совпадающая на отрезке  $[-1, 1]$  с функцией  $(-x^2+1)$ .
22. Написать оператор присваивания, в результате выполнения которого логическая переменная  $t$  принимает значение true, если выполняется указанное условие, и значение false в противном случае:
- числа  $x, y, z$  равны между собой;
  - из чисел  $x, y, z$  только два равны между собой;
  - целые числа  $m$  и  $n$  имеют одинаковую чётность;
  - только одна из логических переменных имеет значение true;
  - $x$  – положительное число;
  - цифра 7 входит в десятичную запись трехзначного целого числа  $k$  раз;
  - из чисел  $x, y, z$  хотя бы два числа положительные;
  - точка  $(x1, y1)$  попадает внутрь круга радиуса  $r$ ;
  - даны  $l$  и  $r$  ( $1 \geq r \geq 0$ ). Точка  $(x1, y1)$  попадает внутрь кольца, внешний радиус которого равен  $l$ , а внутренний –  $r$ ;
  - поля шахматной доски  $(z1, v1)$  и  $(z2, v2)$  имеют одинаковый цвет. ( $z1, z2, v1, v2$  – целые числа от 1 до 8);
  - ферзь, расположенный на поле  $(z1, v1)$  шахматной доски, “бьет” фигуру, расположенную на поле  $(z2, v2)$ ;
  - конь за один ход может перейти с поля  $(z1, v1)$  на поле  $(z2, v2)$ .
23. Составить программу вычисления функции  $f(x)$  вида ✦
- $f(x) = \begin{cases} x^2 - 3x + 8, & x^2 - x \leq 1 \\ \frac{1}{x^2 - 3x + 8}, & x^2 - x > 1 \end{cases}$
  - $f(x) = \begin{cases} x^2 + 3x + 4, & x \leq 0 \\ (x^2 + 3x + 4)^{-1}, & 0 < x \leq 1 \\ x^2 - x, & x > 1 \end{cases}$
  - $f(x) = \begin{cases} x^2 - x, & 0 \leq x \leq 1 \\ x^2 - \sin(x^2), & x > 1 \end{cases}$
24. Для заданных значений  $x, y$  записать программу вычисления  $Z = \frac{\min(x, y) + 0,5}{1 + \max^2(x, y)}$ . ▲

*Замечание.* Здесь и далее  $\min(x, y)$  – обозначает минимальное из чисел  $x$  и  $y$ ,  $\max(x, y)$  – максимальное из чисел  $x$  и  $y$ .

25. По заданным значениям  $x, y, z$  вычислить значение  $u$ . ✦
- $u = \max(x + y + z, xyz)$ ;
  - $u = \begin{cases} ax + by & , ax + by \in [c, d) \\ x + y & , ax + by < c \\ 1 - x - y & , ax + by \geq d \end{cases}$ , где  $a, b, c, d$  – известные величины;
  - $u = \min(x, \max(y, z))$ ;
  - $u = \frac{\min\left(\frac{x + y + z}{3}, xyz\right)}{1 + \min^2\left(\frac{x + y + z}{3}, xyz\right)}$ .

26. Даны вещественные числа  $x_1, x_2, x_3, y_1, y_2, y_3$ . Принадлежит ли начало координат треугольнику с вершинами  $(x_1, y_1), (x_2, y_2), (x_3, y_3)$ ? ★
27. Сколько общих точек у прямой  $y=kx+b$  и окружности  $x^2+y^2=R^2$ ? ▲
28. Даны вещественные положительные числа  $a, b, c, d$ . Выяснить, можно ли прямоугольник со сторонами  $a, b$  уместить внутри прямоугольника со сторонами  $c, d$  так, чтобы каждая из сторон одного прямоугольника была параллельна или перпендикулярна каждой стороне второго прямоугольника. ▲
29. Составить программу, которая проверяет, пройдет ли кирпич с ребрами  $a, b, c$  в прямоугольное отверстие со сторонами  $x$  и  $y$ . Просовывать кирпич в отверстие разрешается только так, чтобы каждое из его ребер было параллельно или перпендикулярно каждой из сторон отверстия. †
30. Дано целое  $k, 1 \leq k \leq 180$ . Определить, какая цифра находится в  $k$ -ой позиции последовательности 101112131415...9899. ★
31. Составить программу нахождения корней биквадратного уравнения  $ax^4+bx^2+c=0$ . ★
32. Ввести три числа. Если они могут быть длинами сторон прямоугольного треугольника, вывести их в порядке возрастания, вычислить площадь полученного треугольника.
33. Ввести три числа. Если они могут быть длинами сторон остроугольного треугольника, вывести их в порядке убывания, вычислить площадь полученного треугольника.
34. Ввести три числа. Если они могут быть длинами сторон тупоугольного треугольника, вывести их в порядке убывания, вычислить площадь полученного треугольника.
35. Ввести три числа. Если они могут быть сторонами равностороннего треугольника, вычислить его площадь и длину высоты. Вывести стороны, площадь и длину высоты в порядке возрастания.
36. Ввести три числа. Если они могут быть длинами сторон равнобедренного треугольника, вычислить длины его высот. Вывести длину основания и длины высот в порядке возрастания.
37. Ввести три числа. Если они могут быть длинами сторон разностороннего тупоугольного треугольника, вывести их в порядке возрастания, вычислить площадь полученного треугольника.
38. Ввести три числа. Если они могут быть длинами сторон равнобедренного тупоугольного треугольника, вычислить его площадь. Вывести длины сторон и площадь в порядке возрастания.
39. Ввести три числа. Если они могут быть длинами сторон равнобедренного остроугольного треугольника, вычислить его площадь. Вывести длины сторон и площадь в порядке возрастания.
40. Ввести три числа. Если они могут быть длинами сторон разностороннего остроугольного треугольника, вывести их в порядке возрастания, вычислить площадь полученного треугольника.
41. Даны координаты трех точек на плоскости. Если они могут быть вершинами прямоугольного треугольника, вывести их в порядке возрастания, вычислить площадь полученного треугольника.
42. Даны координаты трех точек на плоскости. Если они могут быть вершинами остроугольного треугольника, вывести их в порядке убывания, вычислить площадь полученного треугольника.
43. Даны координаты трех точек на плоскости. Если они могут быть вершинами тупоугольного треугольника, вычислить его площадь. Вывести длины сторон в порядке убывания.
44. Даны координаты трех точек на плоскости. Если они могут быть вершинами равностороннего треугольника, вычислить его площадь и длину высоты. Вывести стороны, площадь и длину высоты в порядке возрастания.
45. Даны координаты трех точек на плоскости. Если они могут быть вершинами равнобедренного треугольника, вычислить длины его высот. Вывести длину основания и длины высот в порядке возрастания.

46. Даны координаты трех точек на плоскости. Если они могут быть вершинами разностороннего тупоугольного треугольника, вывести их в порядке возрастания, вычислить площадь полученного треугольника.
47. Даны координаты трех точек на плоскости. Если они могут быть вершинами равнобедренного тупоугольного треугольника, вычислить его площадь. Вывести длины сторон и площадь в порядке возрастания.
48. Даны координаты трех точек на плоскости. Если они могут быть вершинами равнобедренного остроугольного треугольника, вычислить его площадь. Вывести длины сторон и площадь в порядке возрастания.
49. Даны координаты трех точек на плоскости. Если они могут быть вершинами разностороннего остроугольного треугольника, вывести их в порядке возрастания, вычислить площадь полученного треугольника.
50. Даны три числа. Если они могут быть длинами сторон треугольника, определить его вид (разносторонний, равнобедренный, равносторонний). Вычислить длины его высот и напечатать их в порядке убывания.
51. Даны три числа. Если они могут быть длинами сторон треугольника, определить его вид (прямоугольный, тупоугольный, остроугольный). Вычислить длины его высот и напечатать их в порядке убывания.
52. Даны координаты трех точек на плоскости. Если они могут быть вершинами треугольника, определить его вид (разносторонний, равнобедренный, равносторонний). Вычислить длины его высот и напечатать их в порядке убывания.
53. Даны координаты трех точек на плоскости. Если они могут быть вершинами треугольника, определить его вид (прямоугольный, тупоугольный, остроугольный). Вычислить длины его высот и напечатать их в порядке убывания.
54. Составить программу, которая определяла бы вид треугольника (равносторонний, равнобедренный, разносторонний, прямоугольный, тупоугольный, остроугольный), если по данным трём отрезкам его можно построить.
55. Даны координаты трех точек на плоскости. Составить программу, которая определяла бы вид треугольника (равносторонний, равнобедренный, разносторонний, прямоугольный, тупоугольный, остроугольный), если данные координаты вершин позволяют его построить.
56. Даны координаты вершин четырехугольника. Составить программу, которая определяла бы, является ли этот четырехугольник прямоугольником.
57. Даны координаты трех вершин прямоугольника. Определить координаты четвертой вершины.