

Ravesli [Ravesli](#)


- [Уроки по C++](#)
- [OpenGL](#)
- [SFML](#)
- [Qt5](#)
- [RegExr](#)
- [Ассемблер](#)
- [Купить .PDF](#)


Урок №34. Логический тип данных bool

 [Юрий](#) |

- [Уроки C++](#)

|

 Обновл. 2 Сен 2020 |

 71850

[↑](#)  62

В реальной жизни перед нами очень часто возникают вопросы, на которые можно ответить однозначно: «Да» или «Нет». Яблоко является фруктом? Да! Вам нравится спаржа? Нет!

Рассмотрим утверждение: «Яблоко — это фрукт». Является ли это правдой? Да! Яблоко действительно является фруктом. Или как насчет: «Я люблю спаржу». Абсолютная ложь (тьфу!).

Подобные [стейтменты](#), которые имеют только два возможных исхода — да/правда или нет/ложь, настолько распространены, что многие языки программирования добавили специальный тип для работы с ними — логический тип данных. В языке C++ это **логический тип данных bool** (от англ. «*boolean*»).

Оглавление:

1. [Переменные логического типа данных](#)
2. [Использование логического типа данных в ветвлениях if](#)
3. [Возвращаемые значения логического типа данных](#)
4. [Тест](#)

Переменные логического типа данных

Логические переменные — это переменные, диапазон которых состоит только из двух возможных значений: true (1) и false (0).

Для объявления логической переменной используется **ключевое слово bool**:

```
1 bool b;
```

Инициализировать логическую переменную или выполнить операцию присваивания можно с помощью **ключевых слов** true или false:

```
1 bool b1 = true; // копирующая инициализация
2 bool b2(false); // прямая инициализация
3 bool b3 { true }; // uniform-инициализация (C++11)
4 b3 = false; // операция присваивания
```

Аналогично работе унарного оператора минус (-), с помощью которого мы можем сделать число отрицательным, с помощью логического оператора НЕ (!) мы можем изменить true на false и наоборот (false на true):

```
1 bool b1 = !true; // значение b1 - false
2 bool b2(!false); // значение b2 - true
```

На самом деле, логические значения не сохраняются как true или false. Они обрабатываются в виде целых чисел: вместо true — единица, вместо false — ноль.

Следовательно, если мы попытаемся вывести логические значения с помощью [std::cout](#), то увидим либо 0, либо 1:

```
1 #include <iostream>
2
3 int main()
4 {
5     std::cout << true << std::endl; // вместо true единица
6     std::cout << !true << std::endl; // вместо !true ноль
7
8     bool b(false);
9     std::cout << b << std::endl; // b - false (0)
10    std::cout << !b << std::endl; // !b - true (1)
11    return 0;
12 }
```

Результат выполнения программы:

```
1
0
0
1
```

Если вы хотите, чтобы std::cout выводил true или false (вместо целых чисел), то тогда используйте манипулятор форматирования **std::boolalpha**:

```
1 #include <iostream>
2
3 int main()
4 {
5     std::cout << true << std::endl;
6     std::cout << false << std::endl;
7
8     std::cout << std::boolalpha; // выводим логические значения как "true" или "false"
9
10    std::cout << true << std::endl;
11    std::cout << false << std::endl;
12    return 0;
```

```
13 | }
```

Результат выполнения программы:

```
1
0
true
false
```

Использование логического типа данных в ветвлениях if

Очень часто логические переменные используются в ветвлениях if. Ветвление if выглядит следующим образом:

```
if (выражение) стейтмент1;
```

Либо:

```
if (выражение) стейтмент1;
else стейтмент2;
```

(выражение) еще называется «условием», либо «условным выражением».

В обоих случаях, если результатом условия является ненулевое значение, то выполняется стейтмент1. Если же результатом условия является нулевое значение, то выполняется стейтмент2.

Помните, что true — это 1 (ненулевое значение), а false — это 0 (нулевое значение).

Теперь рассмотрим пример в коде:

```
1 if (true) // true - это условие
2     std::cout << "The condition is true!" << std::endl;
3 else
4     std::cout << "The condition is false!" << std::endl;
```

Результат:

```
The condition is true!
```

Что здесь делается? Во-первых, мы начинаем с условия if, которым является логическое значение true, т.е. 1 (ненулевое значение), что означает, что выполняться будет стейтмент1.

Следующая программа работает аналогично:

```
1 bool b(false);
2 if (b)
3     std::cout << "b is true!" << std::endl;
4 else
5     std::cout << "b is false!" << std::endl;
```

Результат:

```
b is false!
```

Здесь, при проверке условия, переменная `b` имеет значение `false`. `false` — это `0`. Следовательно, первый стейтмент под `if` (который `true`) пропускается, а второй, который под `else` (`false`) — выполняется.

А теперь рассмотрим пример посложнее. Оператор равенства (`==`) используется для сравнения двух чисел (являются ли они равными). Оператор `==` возвращает `true`, если операнды равны и `false`, если таковыми не являются:

```
1  #include <iostream>
2
3  int main()
4  {
5      std::cout << "Enter an integer: ";
6      int x;
7      std::cin >> x;
8
9      if (x == 0)
10         std::cout << "The value is zero" << std::endl;
11     else
12         std::cout << "The value is non-zero" << std::endl;
13     return 0;
14 }
```

Результат выполнения программы:

```
Enter an integer: 4
The value is non-zero
```

Давайте разберемся, что и как здесь работает. Во-первых, мы просим пользователя ввести целое число. После этого, с помощью оператора `==`, мы проверяем, является ли пользовательское число нулевым. В вышеприведенном примере `4` не равно `0`, поэтому оператор `==` определяет условие как `false`. Следовательно, выполняется стейтмент2 (тот, который под `else`), где мы выводим `The value is non-zero`.

Возвращаемые значения логического типа данных

Логические значения часто используются в качестве возвращаемых значений в функциях. Названия таких функций очень часто начинаются со слов `is` (например, `isEqual`) или `has` (например, `hasCommonDivisor`).

Рассмотрим следующий пример:

```
1  #include <iostream>
2
3  // Возвращаем true, если x и y равны, в противном случае - возвращаем false
4  bool isEqual(int x, int y)
5  {
6      return (x == y); // оператор == возвращает true, если x равно y, в противном сл
7  }
8
9  int main()
10 {
11     std::cout << "Enter an integer: ";
```

```
12     int x;
13     std::cin >> x;
14
15     std::cout << "Enter another integer: ";
16     int y;
17     std::cin >> y;
18
19     if (isEqual(x, y))
20         std::cout << x << " and " << y << " are equal" << std::endl;
21     else
22         std::cout << x << " and " << y << " are not equal" << std::endl;
23
24     return 0;
25 }
```

Результат выполнения программы:

```
Enter an integer: 5
Enter another integer: 5
5 and 5 are equal
```

Как это работает? Во-первых, мы указываем значения переменным `x` и `y`. Затем проверяется условие, что приводит к вызову функции `isEqual(5, 5)`. Внутри этой функции наши два числа сравниваются между собой (`5 == 5`), что приводит к возврату значения `true` (так как `5 = 5`). Значение `true` возвращается обратно в `caller`. Так как условие истинно, то выполняется `стейтмент1`, который выводит `5 and 5 are equal`.

К логическим значениям нужно немного привыкнуть, но как только вы это сделаете, то сами удивитесь, насколько они удобны и просты.

Во всех примерах, приведенных выше, в наших условиях были либо логические значения (`true` или `false`), либо логические переменные, либо функции, которые возвращают логическое значение. А что произойдет, если мы не будем использовать логическое значение в условиях? Правильно! Если результатом условия будет любое ненулевое значение, то выполняться будет `стейтмент1`.

Поэтому, если попробовать сделать что-то вроде следующего:

```
1 if (4)
2     std::cout << "hi";
3 else
4     std::cout << "bye";
```

То результатом будет `hi`, так как `4` является ненулевым значением.

Тест

Что такое простое число? Правильно! Это целое положительное число больше единицы, которое делится без остатка либо на себя, либо на единицу. Напишите программу, которая просит пользователя ввести простое целое число, меньшее 10. Если пользователь ввел одно из следующих чисел: 2, 3, 5 или 7 — программа должна вывести `The digit is prime`, в противном случае — `The digit is not prime`.

Подсказка: Используйте ветвление if для сравнения чисел и логические значения для отслеживания того, является ли пользовательское число простым или нет.

Ответ

```
1  #include <iostream>
2
3  bool isPrime(int x)
4  {
5      if (x == 2)
6          return true;
7      if (x == 3)
8          return true;
9      if (x == 5)
10         return true;
11     if (x == 7)
12         return true;
13     return false;
14 }
15
16 int main()
17 {
18     std::cout << "Enter a single digit integer: ";
19     int x;
20     std::cin >> x;
21
22     bool prime = isPrime(x);
23
24     if (prime)
25         std::cout << "The digit is prime" << std::endl;
26     else
27         std::cout << "The digit is not prime" << std::endl;
28
29     return 0;
30 }
```

Оценить статью:



(364 оценок, среднее: 4,88 из 5)



[← Урок №33. Типы данных с плавающей точкой: float, double и long double](#)



[Урок №35. Символьный тип данных char →](#)

Комментариев: 62



1. *Дмитрий:*

[19 сентября 2020 в 16:44](#)

проверяет натуральное число ввел пользователь или нет.

```
1 #include <iostream>
2 int isPrime()
3 {
4     std::cout << "Enter a natural number: ";
5     int a;
6     std::cin >> a;
7
8     if (a > 0)
9         a = a / a;
10
11     else
12         a = 0;
13     return a;
14 }
15
16 int main()
17 {
18     int a = isPrime();
19     if (a == 1)
20         std::cout << "yes this is a natural number!";
21
22     if(a == 0)
23         std::cout << "no, this is not a natural number" << std::endl;
24
25     return 0;
26 }
```

[Ответить](#)



2. *aaa:*

[30 августа 2020 в 12:39](#)

Для конкретной задачи:

```
1 #include <iostream>
2
3 bool compare (int x)
4 {
5     return (x==2||x==3||x==5||x==7);
6 }
7
8 int main()
9 {
10     std::cout<<"Enter a number: ";
11     int num;
```

```
12     std::cin>>num;
13
14     if(compare(num))
15         std::cout<<"The digit is prime\n";
16     else
17         std::cout<<"The digit is not prime\n";
18
19 }
```

Для все чисел:

```
1  #include <iostream>
2
3  bool compare (int x)
4  {
5      if (x>10)
6      {
7          bool a=x%2;
8          bool b=x%3;
9          bool c=x%4;
10         bool d=x%5;
11         bool f=x%6;
12         bool g=x%7;
13         bool h=x%8;
14         bool j=x%9;
15         return (a==true && b==true && c==true && d==true && f==true && g==true
16             )
17     else
18     {
19         return (x==2||x==3||x==5||x==7);
20     }
21 }
22
23
24 int main()
25 {
26     std::cout<<"Enter a number: ";
27     int num;
28     std::cin>>num;
29
30     if(compare(num))
31         std::cout<<"The digit is prime\n";
32     else
33         std::cout<<"The digit is not prime\n";
34
35 }
```

Но я точно не уверен =))

[Ответить](#)



3. *Elaiten:*

[24 июля 2020 в 22:08](#)

Программа которая будет проверять четное число или нет, если вы ввели отрицательное число или вместо числа букву, то программа выдаст ошибку и закроется:

```
1  #include <iostream>
2
3  using namespace std;
4
5  bool istrue(int x)
6  {
7      for (int i = 2; i < x; i++)
8      {
9          if (x % i == 0)
10         {
11             x = true;
12         }
13         else
14         {
15             x = false;
16         }
17     }
18     return x;
19 }
20
21 int main()
22 {
23     setlocale(LC_CTYPE, "Russian");
24     cout << "Введите простое простое положительное число" << endl;
25     int x;
26     bool fail = false;
27     cin >> x;
28     if (cin.fail())
29     {
30         fail = true;
31         cout << "Введите число которое будет соблюдать условие" << endl;
32         return (-1);
33     }
34     cin.clear();
35     cin.ignore();
36     cout << boolalpha;
37     if (x > 0)
38     {
39         if (istrue(x))
40             cout << "Ваше число является четным " << endl;
41         else
42             cout << "Ваше число является нечетным " << endl;
43     }
44     else
```

```

45         cout << "Введите положительное число" << endl;
46     return 0;
47 }

```

[Ответить](#)



4. Александр:

[12 июля 2020 в 16:14](#)

```

1  //Программа работает со всеми положительными и отрицательными числами кроме 2
2  #include <iostream>
3  #include <cmath>
4
5  using namespace std;
6
7  //Функция проверяет делится ли ЧИСЛО нацело на все числа от 2 до этого ЧИСЛА(не
8  //Если ЧИСЛО делится нацело хоть на одно число, то оно НЕ простое
9  //Каждый раз когда деление не дает нулевой результат к переменной y добавляется
10 bool isPrime(int x) {
11
12     //Переменная отвечает за к-тво чисел при деление на которых не будет целый
13     int y = 0;
14     for (int i = 2; i < abs(x); i++) {
15         if (x % i == 0) {
16             cout << "The number " << x << " is not prime" << endl;
17             exit(0);
18         }
19         else {
20             y += 1;
21         }
22     }
23
24     //Если к-тво чисел с не нулевой остаточей равно ЧИСЛУ -2, то оно ПРОСТОЕ
25     if (y == x - 2) {
26         cout << "The number " << x << " is prime" << endl;
27     }
28     return true;
29 }
30
31 int main() {
32     cout << "Enter the number" << endl;
33     int number;
34     cin >> number;
35     isPrime(number);
36 }

```

[Ответить](#)



5. Onium:

[13 июня 2020 в 00:20](#)

Так тоже работает.

```
1 #include <iostream>
2 using namespace std;
3
4 bool isPrime(int x)
5 {
6     if (x == 2)
7         return true;
8     if (x == 3)
9         return true;
10    if (x == 5)
11        return true;
12    if (x == 7)
13        return true;
14    else
15        return false;
16 }
17
18 int main()
19 {
20     int x;
21     cin >> x;
22
23     if (isPrime(x))
24         cout << "Prime!" << endl;
25     else
26         cout << "Not Prime..." << endl;
27
28     return 0;
29 }
```

Ответить



6. *Onium:*

13 июня 2020 в 00:14

Так программа сверяет число по всей ветке поочередно, что дольше. Зато сам и вроде всё правильно и без ошибок считает.

```
1 #include <iostream>
2 using namespace std;
3
4 bool isPrime(int x, int y, int z, int f)
5 {
6     if (x == 2)
7         return true;
8     else
9         if (y == 3)
10            return true;
11        else
12            if (z == 5)
```

```
13         return true;
14     else
15         if (f == 7)
16             return true;
17         else
18             return false;
19 }
20
21 int main()
22 {
23     int x;
24     cin >> x;
25
26     if (isPrime(x,x,x,x))
27         cout << "Prime!" << endl;
28     else
29         cout << "Not Prime..." << endl;
30
31     return 0;
32 }
```

Ответить



7. *Богдан:*

9 мая 2020 в 11:53

Немного заигрался и написал это. Были ниже некоторые более компактные варианты, но с некоторыми проблемами (ломаются если ввести число ≤ 0).

```
1  #include <iostream>
2  using namespace std;
3
4  void main()
5  {
6      int x;
7      int y;
8      cout << "Enter a positive number: \n";
9
10     for (int i = 0; ; i++)
11     {
12         cin >> x;
13         if (x > 0)
14             break;
15         else
16             cout << "Try again: \n";
17     }
18
19     if (x < 1)
20     {
21         cout << "Number " << x << " is not positive\n";
22     }
23     else if (x == 1)
```

```
24     {
25         cout << "Number " << x << " is prime\n";
26     }
27     else
28     {
29         for (int i = 2; i < x; i++)
30         {
31             if (x % i != 0)
32             {
33                 y = true;
34             }
35             else
36             {
37                 y = false;
38                 break;
39             }
40         }
41         if (y)
42         {
43             cout << "Number " << x << " is prime\n";
44         }
45         else
46         {
47             cout << "Number " << x << " is not prime\n";
48         }
49     }
50 }
51 }
```

[Ответить](#)8. *Sagynysh:*[3 мая 2020 в 12:29](#)

Тут я не поняла. Откуда компилятор знает, что `b = false`? То есть что имеется ввиду под оператором `if(b)`? Возможно туповатый вопрос 😊 Но моя башка что-то не понимает. Хэлп гайс

```
1 bool b(false);
2 if (b)
3     std::cout << "b is true!" << std::endl;
4 else
5     std::cout << "b is false!" << std::endl;
```

[Ответить](#)1. *Богдан:*[9 мая 2020 в 11:55](#)

Вы же в первой строчке присвоили значение `false`

[Ответить](#)



9. Андрей:

[9 марта 2020 в 15:42](#)

Я перечитал 4 раза этот урок, но так и не понял, зачем мы создавали переменную bool, я понял про if else, но момент с написанием bool и функции int main мне не понятен.

Спасибо заранее.ц

[Ответить](#)

10. Константин:

[13 февраля 2020 в 02:22](#)

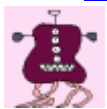
```
1  #include <iostream>
2
3  //возвращает true, если x и y равны, в противном случае возвращает false
4  bool isEqual(int x, int y)
5  {
6      // оператор == возвращает true? если x равно y, в противном случае - false
7      return (x == (2 or 3 or 5 or 7));
8  }
9  int main()
10 {
11     std::cout << "Enter first number an integer:";
12     int x;
13     std::cin >> x;
14
15
16     if (isEqual(x, (2 or 3 or 5 or 7)))
17         std::cout << "The digit is prime" << std::endl;
18     else
19         std::cout << " The digin is not prime " << std::endl;
20     return 0;
21 }
```

[Ответить](#)

1. Хей Лонг:

[10 апреля 2020 в 12:41](#)

Весьма интересный вариант, я даже свой тестовый код для проверки сделал. Для начала нужно понять, зачем в твоей функции isEqual нужны два параметра, когда применяется только один. Потом, логическое сложение (2+3+5+7) сразу вернёт единицу, поэтому твоя программа ответит digit is prime только при введённом x==1.

[Ответить](#)

11. cybersatori:

[8 января 2020 в 10:12](#)

подумал, как быть с ... 11, 13, 17, 19, 23, 29, 31, 37 и т.д. что бы уж до конца корректно.

собрал такой код, но при значении 111 пишет тоже прайм — хотя вроде как нет, подумаю ещё — но пример я так понял игровой на буль синтаксис, он впрочем у меня тоже с другой стороны.

```
1  #include <iostream>
2
3  int inputDigit() {
4      std::cout << "Input digit: ";
5      int x;
6      std::cin >> x;
7      return x;
8  }
9
10 void testNaturalValue(int y) {
11
12     if ((y > 1)&&(y%2!=0)) {
13         if ((y % y == 0)||(y % 1 == 0)) {
14
15             std::cout << "The digit is prime." << std::endl;
16         }
17     }
18     else
19     {
20         std::cout << "The digit is NOT prime." << std::endl;
21     }
22
23 }
24
25
26
27 int main()
28 {
29     int a = inputDigit();
30     testNaturalValue(a);
31
32     return 0;
33 }
```

Ответить



1. *cybersatori:*
[8 января 2020 в 11:52](#)

Вообщем предлагается использовать алгоритм решето Эратосфена, но это уже другая тема, а так интересно спасибо что натолкнули на размышления)

Ответить



12. *Valentin:*
[14 ноября 2019 в 17:48](#)

```
1 int enterNumber()
2 {
3     std::cout << "Enter the number: ";
4     int x;
5     std::cin >> x;
6     return x;
7 }
8
9 bool isPrime(int x)
10 {
11     return (x == x/1 and x>0);
12 }
13
14 int main()
15 {
16     int x = enterNumber();
17
18     if (isPrime(x))
19         std::cout << x << " The digit is prime." << std::endl;
20     else
21         std::cout << x << " The digit in not prime." << std::endl;
22
23     return 0;
24 }
```

Хотелось бы узнать это правильный ответ или я совсем не так понял задачу. И как далеко я от истины. Спасибо

[Ответить](#)



1. *Nikita:*

[17 ноября 2019 в 09:45](#)

Не совсем понятно что делает ваша функция . Она в идеале должна проверять число . Если число делится на цело только на 1 и на саму себя , тогда это простое число. Вот это условие должно быть в функции.

[Ответить](#)



1. *Nikita:*

[17 ноября 2019 в 09:58](#)

Хотя легче сказать чем сделать.

Лично я вижу решение по такому алгоритму:

Функция принимает введенное число — x . Далее, в итеративном цикле, это число , начинает делится на все Числа по порядку от 2 и до $x-1$. Если ни на одно число Z которое находится в этом диапазоне ($1 < Z < x$) наш x не поделится нацело, значит число простое.

[Ответить](#)



1. *Алексей:*
[14 января 2020 в 18:07](#)

Проходить стоит даже не до $x-1$, а до $x/2$.

На числа, которые больше половины нашего числа, оно не поделится нацело



13. *Максим:*
[29 августа 2019 в 08:46](#)

```
1 #include <iostream>
2
3 int main()
4 {
5     std::cout << "write digit:  ";
6     int x;
7     std::cin >> x;
8
9     if (x == 2 | x == 3 | x == 5 | x == 7)
10         std::cout << "The digit is prime" << std::endl;
11     else
12         std::cout << "The digit is not prime" << std::endl;
13     return 0;
14 }
```

[ОТВЕТИТЬ](#)



14. *Проггер:*
[24 августа 2019 в 21:32](#)

```
1 #include <iostream>
2
3 bool isPrime(int x)
4 {
5     if (x == 2, x == 3, x == 5, x == 7)
6         return true;
7 }
8
9 int main()
10 {
11     std::cout << "Enter prime integer: ";
12     int x;
13     std::cin >> x;
14
15     x = isPrime(x);
16
17     if (x)
18         std::cout << "The digit is prime" << std::endl;
19     else
```

```
20         std::cout << "The digit is not prime" << std::endl;
21
22     return 0;
23 }
```

Можно ли писать условие if через запятую(у меня скомпилировалось и все работает)???

[Ответить](#)



15. Юрий:

[26 июля 2019 в 19:10](#)

```
1  #include <iostream>
2
3  using namespace std;
4
5  int readNumber()
6  {
7      cout << "Please enter prime: " << endl;
8
9      int number;
10     cin >> number;
11
12     return number;
13 }
14
15 bool checkNumber(int number)
16 {
17     return number == 2 || number == 3 || number == 5 || number == 7;
18 }
19
20 int main()
21 {
22     int number = readNumber();
23
24     if (checkNumber(number))
25         cout << "the digit is prime" << endl;
26     else
27         cout << "the digin is not prime" << endl;
28
29     return 0;
30 }
```

[Ответить](#)



16. Михалыч:

[5 июля 2019 в 23:55](#)

```
1  #include <iostream>
2
3  using namespace std;
```

```

4
5 int main()
6 {
7     std::cout << "enter an enteger";
8
9     int x;
10    cin >> x;
11
12    if (x==2||x==3||x==5||x==7)
13        cout << "The digit is prime";
14    else
15        cout << "The digit is not prime";
16
17    return 0;
18
19 }

```

таким кодом меньше писать. а работает также.

[Ответить](#)



17. Анастасия:

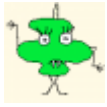
[3 июля 2019 в 09:49](#)

```

1 //Чуть-чуть усложним задачу для полного понимания)))
2
3 //Если говорить о всех числах в общем (кроме 0), и при этом не
4 //использовать циклы(что было бы гораздо удобней)(то есть
5 //опираться чисто на тему урока и изученное ранее), то получается
6 //вот такая каша.
7
8 #include <iostream>
9
10 bool isPrime(int x)
11 {
12     if ((x % 2 == 0) or (x % 3 == 0) or (x % 5 == 0) or (x % 7 == 0))
13         if ((x == 2) or (x == 3) or (x == 5) or (x == 7))
14             return true;
15     else return false;
16     else return true;
17 }
18
19
20 int main()
21 {
22     std::cout << "Enter a digit integer ";
23     int x;
24     std::cin >> x;
25
26     if (isPrime(x))
27         std::cout << "\nYour digit is prime!" << std::endl;
28

```

```
29     else
30         std::cout << "\nYour digit is not prime!" << std::endl;
31
32     return 0;
}
```

[Ответить](#)18. *Алексей:*[25 июня 2019 в 16:46](#)

Немного по-другому сделал.

```
1  #include <iostream>
2
3  bool check(int a)
4  {
5      if (a == 2 or a == 3 or a == 5 or a == 7) return true;
6      else return false;
7  }
8
9  int main()
10 {
11     std::cout << "Enter number:";
12     int a;
13     std::cin >> a;
14     bool prime = check(a);
15     if(prime) std::cout << "True!" << std::endl;
16     else std::cout << "False!" << std::endl;
17 }
```

Тупил конечно, но тут вовсе ничего сложного.

[Ответить](#)19. *Nikita:*[16 мая 2019 в 22:19](#)

А с циклом нельзя?

```
1  bool isPrime(int p)
2  {
3      for (int i = 2; i < p; i++)
4      {
5          if (p%i == 0)
6              return 0;
7      }
8      return 1;
9  }
10
11 int main()
12 {
```

```
13     int x;
14     cout << "Vvedite prostoe celoe chislo: ";
15     cin >> x;
16
17     if (isPrime(x))
18         cout << "The digit is prime\n";
19     else
20         cout << "The digit is not prime\n";
21
22
23 }
```

[Ответить](#)

20. Vladislav:

[3 мая 2019 в 00:59](#)

Объясните мне — Нахрена использовать "std::cout/cin", если у Вас подключена библиотека?

[Ответить](#)

1. Роман:

[17 июня 2019 в 20:06](#)

Это необходимо использовать, так как не подключено пространство имён std

[Ответить](#)

21. Denis:

[17 апреля 2019 в 13:19](#)

```
1  #include "pch.h"
2  #include <iostream>
3
4  using namespace std;
5
6  int getNumber() // ВВОДИМ ЧИСЛО
7  {
8      cout << "Введите число: ";
9      int num;
10     cin >> num;
11     return num;
12 }
13
14 bool isPrimeDigit(int a) // Is prime digit?
15 {
16     return ((a % 1 == 0) && (a % a == 0) && (a > 1));
17 }
18
19 void result(bool value) // Result
20 {
```

```
21     if (value)
22         cout << "The prime is digit!";
23     else
24         cout << "The prime is not digit!";
25 }
26
27 int main()
28 {
29     setlocale(LC_ALL, "ru");
30
31     bool value = isPrimeDigit(getNumber());
32     result(value);
33
34     return 0;
35 }
```

[Ответить](#)



22. *Владимир:*

[12 февраля 2019 в 04:24](#)

Сделал вариант через три функции.

```
1  #include <iostream>
2  using namespace std;
3
4  int getNumber() // Вводим число
5  {
6      cout << "Введите число: " << endl;
7      int num;
8      cin >> num;
9      return num;
10 }
11
12 bool isEven(int a) // Проверяем четность
13 {
14     if (a % 2 == 0)
15     {
16         return true;
17     }
18     else
19     {
20         return false;
21     }
22 }
23
24 void showResult(bool value) // Выводим результат
25 {
26     if (value == true)
27     {
28         cout << "Число чётное." << endl;
29     }
```

```

30     }
31     if (value == false)
32     {
33         cout << "Число нечетное." << endl;
34     }
35 }
36
37 int main()
38 {
39     setlocale(LC_ALL, "ru");
40
41     bool value = isEven(getNumber());
42
43     showResult(value);
44
45     system("pause");
46     return 0;
}

```

Ответить1. *Nikita:*[16 мая 2019 в 16:20](#)

Нам ведь нужно не четность проверить. А простое ли число

Ответить23. *Айзек:*[10 февраля 2019 в 21:51](#)

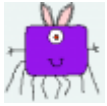
Блин, а я всё размышлял, как не проходя циклов решить это задание
Оказалось, что нужно внимательно читать ТЗ 😊

```

1 bool isPrime(int digit) {
2
3     if (digit < 1)
4         return false;
5
6     bool b(true);
7     for (int i(digit - 1); i > 1; i--) {
8         if (digit % i == 0)
9             b=false;
10
11     }
12     return b;
13 }
14
15 int main() {
16     std::cout << "Please, enter a digit: ";
17

```

```
18     int digit;
19     std::cin >> digit;
20
21     if (isPrime(digit))
22         std::cout << "The digit is prime";
23     else
24         std::cout << "The digit is not prime";
25 }
```

[Ответить](#)

24. Лёха:

[1 февраля 2019 в 19:47](#)

```
1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5      int a;
6      cout<<"Enter a digit = ";
7      cin>>a;
8
9      if(a==2||a==3||a==5||a==7)
10     {
11         cout<<"The digit is prime";
12     }
13     else
14         cout<<"The digit is not prime";
15     return 0;
16
17 }
```

[Ответить](#)

25. Вячеслав:

[15 января 2019 в 23:52](#)

```
1  #include "pch.h"
2  #include <iostream>
3  using namespace std;
4
5  bool isPrimeInteger(int x)
6  {
7      if (x = 2)
8          return true;
9      if (x = 3)
10         return true;
11     if (x = 5)
12         return true;
13     if (x = 7)
14         return true;
```



```
15     return false;
16 }
17
18 int main()
19 {
20     cout << "Enter a prime integer:";
21     int x;
22     cin >> x;
23
24     bool prime = isPrimeInteger(x);
25
26     if(prime)
27         cout << "The digit is prime" << endl;
28     else
29         cout << "The digit is non prime" << endl;
30
31     return 0;
32 }
```

так будет правильно, у меня работает?

[Ответить](#)



1. *Tosha:*

[29 января 2019 в 20:00](#)

Не-а, тут в условиях используется присваивание(=) вместо сравнения(==), благодаря чему любое число, отличное от нуля будет воспринято, как true в первом же выражении

[Ответить](#)



26. *Вячеслав:*

[15 января 2019 в 23:17](#)

Юрий, доброго времени суток. У меня стоит MVS 2017 community, как с нее можно скопировать код, чтобы отправить? Заранее спасибо.

[Ответить](#)



1. *Юрий:*

[15 января 2019 в 23:22](#)

Привет. Просто Ctrl+C и Ctrl+V — дальше я уже самостоятельно редактирую код, чтобы было так красиво, как в статьях 😊

[Ответить](#)



27. *Виктор:*

[11 декабря 2018 в 02:05](#)

Мой вариант:

```
1 #include "stdafx.h"
2 #include <iostream>
3 #include <Windows.h>
4
5 using namespace std;
6 int inputUserNumber()
7 {
8     cout << "Введите простое целое число: ";
9     int x;
10    cin >> x;
11    return x;
12 }
13 bool comparingUserNumber(int x)
14 {
15     if (x == 2 or
16         x == 3 or
17         x == 5 or
18         x == 7)
19         return true;
20     return false;
21 }
22 void prinResult(bool result)
23 {
24     if (result)
25         cout << "Цыфра проста!" << endl;
26     else
27         cout << "Цыфра не простая!" << endl;
28 }
29 int main()
30 {
31     SetConsoleCP(1251);
32     SetConsoleOutputCP(1251);
33     int x = inputUserNumber();
34     bool result = comparingUserNumber(x);
35     prinResult(result);
36     return 0;
37 }
```

[Ответить](#)



28. *katerinkaj:*

[12 ноября 2018 в 00:49](#)

Единственно смущает, что простых чисел куда больше, чем эти 3 числа. Так что задача и ее решение не совсем корректны. Как написать формулу для всех простых чисел так сразу не дотумкала, решила на этом не заикливаться и не тратить время. Наверняка как-то в цикле можно это условие описать...

```
1 setlocale(LC_ALL, "RUS");
2
3 float x{};
4
```

```

5  std::cout << "Введите простое число: ";
6  std::cin >> x;
7
8  if ((x / 2 == 1) || (x/3==1)|| (x / 5 == 1) || (x / 7 == 1))
9  {
10
11      std::cout << "Вы действительно ввели простое число!";
12  }
13  else
14  {
15      std::cout << "Вы нихренашечки не знаете о простых числах";
16  }

```

[Ответить](#)29. *subj:*[25 октября 2018 в 15:50](#)

```

1  int main()
2  {
3      int x;
4      std::cout << "Enter a simple number";
5      std::cin >> x;
6
7      if (x == 2)
8          std::cout << "The digit is prime";
9      else
10         if (x == 3)
11             std::cout << "The digit is prime";
12         else
13             if (x == 5)
14                 std::cout << "The digit is prime";
15             else
16                 if (x == 7)
17                     std::cout << "The digit is prime";
18                 else
19                     std::cout << "The digit is not prime";
20
21         return 0;
22     }
23 }

```

Сделал не правильно, но работает.

[Ответить](#)30. *Алексей:*[23 сентября 2018 в 07:40](#)

```

1  #include <iostream>
2

```

```
3 using namespace std;
4
5 bool numCheck (int x);
6 int getNum ();
7 void showRes (bool x);
8
9 int main()
10 {
11     int x = getNum ();
12     bool y = numCheck (x);
13     showRes (y);
14     //showRes (numCheck (x));
15
16     return 0;
17 }
18 bool numCheck (int x)
19 {
20     if (x==2)
21         return true;
22     if (x==3)
23         return true;
24     if (x==5)
25         return true;
26     if (x==7)
27         return true;
28     return false;
29 }
30 int getNum ()
31 {
32     cout << "Enter a single digit integer pls" <<endl;
33     int x;
34     cin >> x;
35     return x;
36 }
37 void showRes (bool x)
38 {
39     if (x)
40         cout << "The digit is prime" << endl;
41     else
42         cout << "The digit is not prime" << endl;
43 }
```

ОТВЕТИТЬ



31. *Максим:*

[4 сентября 2018 в 00:18](#)

```
1 #include <iostream>
2
3 bool isPrime(int x)
4 {
```

```
5     if ((x-2)*(x-3)*(x-5)*(x-7)==0)
6         return true;
7     else
8         return false;
9 }
10
11
12 int main()
13 {
14     int x;
15     std::cout << "Enter an integer from 1 to 10: ";
16     std::cin >> x;
17     if (isPrime(x))
18         std::cout<<"The digit is prime\n";
19     else
20         std::cout<<"The digit is not prime\n";
21
22     return 0;
23 }
```

Ответить



32. Коля:

[1 августа 2018 в 18:18](#)

я правильно написал?

```
1 #include <iostream>
2 using namespace std;
3 int chcker(int x) {
4     if (x == 2 || x == 5 || x== 3 || x == 7) {
5         return(true);
6     }
7     else {
8         return(false);
9     }
10 }
11
12
13 int main() {
14     int x = 0;
15     cin >> x;
16     chcker(x);
17     if (chcker(true)) {
18         cout << "The digit is prime" << endl;
19     }
20     else {
21         cout << "The digit is not prime" << endl;
22     }
23
24
25 system("pause");
```

```
26 | return 0;  
27 | }
```

[Ответить](#)33. *artem:*[5 июня 2018 в 11:48](#)

```
1 | #include "stdafx.h"  
2 | #include <iostream>  
3 |  
4 | using namespace std;  
5 |  
6 | int main()  
7 | {  
8 |     cout << "Enter a number" << endl;  
9 |     int a;  
10 |    cin >> a;  
11 |    if (a == 2 || a == 3 || a == 5 || a == 7)  
12 |    {  
13 |        cout << "The digit is prime" << endl;  
14 |    }  
15 |    else  
16 |    {  
17 |        cout << "The digit is not prime" << endl;  
18 |    }  
19 | }
```

[Ответить](#)34. *Андрей:*[24 марта 2018 в 00:34](#)

Спасибо за статьи!

Добавлю свое решение

```
1 | #include <iostream>  
2 |  
3 | bool isSimple(int x)  
4 | {  
5 |     return (x == 2 || x == 3 || x == 5 || x == 7);  
6 | }  
7 |  
8 | int main()  
9 | {  
10 |    std::cout << "Enter integred number: ";  
11 |    int y;  
12 |    std::cin >> y;  
13 |  
14 |    if (isSimple(y))  
15 |        std::cout << "The number is simple";
```

```
16
17     else
18         std::cout << "The number is not simple";
19
20
21     return 0;
22 }
```

[Ответить](#)

1. Юрий:

[24 марта 2018 в 01:37](#)

Привет, пожалуйста 😊

[Ответить](#)

1. Роман:

[18 июля 2018 в 13:09](#)

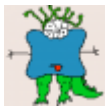
помогите пожалуйста как в комментариях так красиво код выкладывать со строчками? если что у меня IDE dev c++

[Ответить](#)

1. Юрий:

[18 июля 2018 в 19:55](#)

Это стили, которые прописываются на моей стороне (стороне админа), а не на вашей. Вы вставляете свой код как есть, дальше я его редактирую.

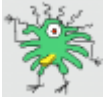


35. Кирилл:

[3 марта 2018 в 15:23](#)

```
1  #include <iostream>
2
3  bool isPrime()
4  {
5      using namespace std;
6      cout << "enter a number " << endl;
7      int x;
8      cin >> x;
9      return (x == 2 || x == 3 || x == 5 || x == 7);
10 }
11
12 int main()
13 {
14     using namespace std;
15     if (isPrime())
```

```
16         cout << "prime" << endl;
17     else
18         cout << "not prime" << endl;
19 }
```

[Ответить](#)36. *aleksandr:*[19 февраля 2018 в 16:48](#)

А так грамотно ?

```
1  #include "stdafx.h"
2  #include <iostream>
3
4  using namespace std;
5
6  bool isNumberPrime(int x);
7
8  int enterNumber();
9
10 int main() {
11
12     int x = enterNumber();
13
14     if (isNumberPrime(x)) {
15         cout << "Yes " << x << " are prime." << endl;
16     }
17     else {
18         cout << "No " << x << " are not prime." << endl;
19     }
20
21     system("pause");
22
23     return 0;
24 }
25
26 bool isNumberPrime(int x) {
27     if (x == 2 || x == 3 || x == 5 || x == 7) {
28         return true;
29     }
30 }
31 else {
32     return false;
33 }
34 }
35
36 int enterNumber() {
37     cout << "Enter number : ";
38     int x;
39     cin >> x;
40     return x;
}
```


41 | }

[Ответить](#)

1. Юрий:

[19 февраля 2018 в 21:34](#)

Как вариант, может быть.

[Ответить](#)

37. Андрей:

[16 января 2018 в 13:03](#)

При подключении Windows.h, надо в консоли (командной строке) установить по умолчанию шрифт Lucida Console.

У меня кстати и ввод и вывод работают.
Автору спасибо!!!

[Ответить](#)

38. Александр:

[16 января 2018 в 09:16](#)

А может быть лучше вот так?

```
1  #include <iostream>
2
3
4  int main(int x)
5  {
6
7      std::cout << "Enter simple integer: ";
8      std::cin >> x;
9
10     if (x == 2 || x == 3 || x == 5 || x == 7)
11     {
12         std::cout << "The digit is prime." << std::endl;
13     }
14
15     else
16     {
17         std::cout << "The digit is not prime." << std::endl;
18     }
19     return 0;
20 }
```

[Ответить](#)



1. *Юрий:*

[18 января 2018 в 00:02](#)

Нет, так не лучше. Во-первых, в функции `main()` — параметры не объявляются. Во-вторых, вы ведь просто переместили код функции в `main()`, если работать по такой логике, то зачем вообще придумали функции, если весь код можно разместить в `main`-е?

Код выносится в функции, чтобы его можно было повторно легко использовать и легко поддерживать (модифицировать).

Да, всё будет работать и так в `main`-е, это задание не большое — кода мало, но мы ведь учимся писать код правильно, не так ли? И здесь лучше использовать отдельную функцию.

[Ответить](#)



39. *Андрей:*

[3 января 2018 в 14:48](#)

У меня вопрос, нашел в инете способ вывода русских букв в консоль. Он нормальный или есть какие-то подводные камни.

Еще пишут можно сохранять файл в другой кодировке, но я не понял какой файл, `src` ? и как это сделать...

[Ответить](#)



1. *Юрий:*

[4 января 2018 в 13:47](#)

Для корректного отображения кириллицы в консольном окне — добавьте заголовочный файл:

```
1 #include <locale>
```

и пропишите следующую строку кода в `main`:

```
1 setlocale(LC_CTYPE, "rus");
```

Либо, если вы еще планируете вводить данные на кириллице, тогда подключите заголовочный файл:

```
1 #include <Windows.h>
```

и пропишите в `main`:

```
1 SetConsoleCP(1251); SetConsoleOutputCP(1251);
```

[Ответить](#)



1. *Андрей:*

[4 января 2018 в 17:04](#)

Спасибо, `setlocale(LC_STYPE, «rus»);` работает.

Но с `#include`

почему-то не пошло...

В настройках стоит Windows 1251...

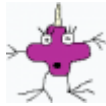
Ответить



1. *Юрий:*

[4 января 2018 в 17:57](#)

С `include` ошибка выбивает?



2. *Михаил:*

[14 января 2018 в 22:55](#)

`SetConsoleCP(1251)` и `SetConsoleOutputCP(1251)` не работают у меня на MVS Community 2017: ввод и вывод на кириллице все равно обрабатывает крокозябликами



3. *Юрий:*

[16 января 2018 в 01:07](#)

Попробуйте запустить у себя вот этот код:

```
1  #include "stdafx.h"
2  #include <iostream>
3  #include <Windows.h>
4  using namespace std;
5
6  int main(int argc, char* argv[])
7  {
8      SetConsoleCP(1251);
9      SetConsoleOutputCP(1251);
10
11     char string[20];
12     cin >> string; // вводим строку, используя "Кириллицу"
13     cout << "\nРезультат: " << string << endl; // вывод строки
14     system("pause");
15     return 0;
16 }
```

У меня здесь всё заработало.



4. *Роман:*

[18 июля 2018 в 13:07](#)

ещё работает просто `setlocale(0, "");` для вывода любых символов например вот

```

1 using namespace std;
2 #include <iostream>
3 #include <conio.h>
4
5 int main()
6 {
7     setlocale (0, "");
8     int ten = 0;
9     double drob = 0;
10    int ot;
11    int r;
12    int s; //счётчик цикла
13    int crat; //выделить числа кратные ...
14
15    //записываю число от и до
16    cout << "Введите первое число ";
17    cin >> ot;
18    cout << "Введите последнее число " ;
19    cin >> r;
20
21    //выбираю кратность
22    cout << "Выберите число на которое должны делиться отфильтро
23    cin >> crat;
24
25    for (s = ot; s <= r; s++)
26    {
27        ten = s * 1.0 / crat;
28        drob = s * 1.0 / crat;
29        if (ten == drob)
30        {
31            cout << s << " ";
32        }
33    }
34
35    cout << endl << "Конец программы.";
36    getch();
37    return 0;
38
39
40 }

```



40. **Арте́м:**

[16 декабря 2017 в 23:50](#)

как вариант

```

1 #include <iostream>
2
3 void isSimpleInt(int x)
4 {

```

```
5     if (x == 2 || x == 3 || x == 5 || x == 7)
6     {
7         std::cout << "The digit is prime." << std::endl;
8     }
9     else
10    {
11        std::cout << "The digit is not prime." << std::endl;
12    }
13 }
14
15 int main()
16 {
17     int value;
18     std::cout << "Enter simple integer: ";
19     std::cin >> value;
20     isSimpleInt(value);
21
22     return 0;
23 }
```

Спасибо за статьи! Очень доходчиво и читабельно.

[Ответить](#)



1. *Юрий:*
[17 декабря 2017 в 00:00](#)

Да, так тоже можно 😊

[Ответить](#)

Добавить комментарий

Ваш E-mail не будет опубликован. Обязательные поля помечены *

Имя *

Email *

Комментарий

☐ Сохранить моё Имя и E-mail. Видеть комментарии, отправленные на модерацию

☐ Получать уведомления о новых комментариях по электронной почте. Вы можете [подписаться](#) без комментирования.






TELEGRAM  КАНАЛ

Электронная почта



ПАБЛИК 

ТОП СТАТЬИ

-  [Словарь программиста. Сленг, который должен знать каждый кодер](#)
-  [Урок №1. Введение в программирование](#)
-  [70+ бесплатных ресурсов для изучения программирования](#)
-  [Урок №1: Введение в создание игры «Same Game»](#)
-  [Урок №4. Установка IDE \(Интегрированной Среды Разработки\)](#)

- [Ravesli](#)
- - [О проекте](#) -
- - [Пользовательское Соглашение](#) -
- - [Все статьи](#) -
- Copyright © 2015 - 2020