

Поиск простых чисел

PascalABC, FreePascal

В.А. Жук, учитель информатики ГУО «Средняя школа №11 г.Слуцка»

Определение простого числа

Натуральное число, не равное **1**, называется **простым**, если оно делится только на себя и на **1**.

Простые числа:

**2, 3, 5, 7, 11, 13, 17, 19, 23,
29, 31, 37, 41, 43, 47, ...**

Составные числа

Натуральное число, не равное **1** и не являющееся простым, называется **составным**

Составные числа:

**4, 6, 8, 9, 10, 12, 14, 15, 16, 18,
20, 21, 22, 24, 25, 26, 27, 28, 30, 32,
33, 34, 35, 36, 38, 39, 40, 42, 44, 45,
46, 48, 49, 50, ...**

Что можно заметить?

1. Натуральное число **1** не относится ни к простым, ни к составным.
2. Среди простых чисел только одно чётное число **2**. Остальные числа нечетные.
3. Простых чисел бесконечно много.

Существует множество задач, связанных с простыми числами, и хотя формулируются они достаточно просто, решить их бывает очень трудно. Некоторые свойства простых чисел еще не открыты. Немецкий математик Германа Вейль (Weyl, 1885-1955) так охарактеризовал простые числа: **«Простые числа – это такие существа, которые всегда склонны прятаться от исследователя».**

Во все времена люди хотели найти как можно большее простое число. Пока люди считали только при помощи карандаша и бумаги, им нечасто удавалось обнаружить новые простые числа. До 1952 г. самое большое известное простое число состояло из 39 цифр. Теперь поиском все больших простых чисел занимаются компьютеры. Но для них нужно составлять программы.

Задача

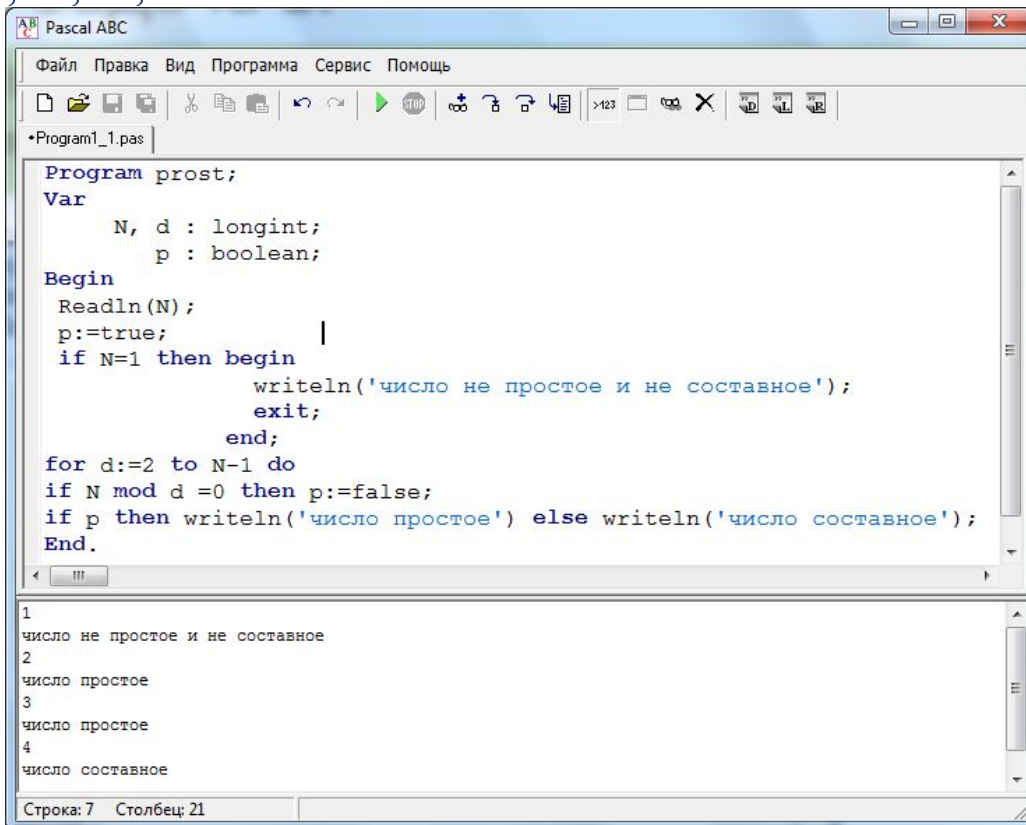
Вводится натуральное число N ($N \leq 2\,147\,483\,647$).

Определить, является ли оно **простым** или **составным**.

Решение:

Будем делить данное число N на все отличные от 1 числа, меньшие N . Если N не разделится ни на одно из этих чисел, то оно будет простым.

Например, введенное число 37. Следовательно, мы делим это число на числа 2, 3, 4, 5, 6, 7, 8, ..., 34, 35, 36.



```
Program prost;
Var
    N, d : longint;
    p : boolean;
Begin
    Readln(N);
    p:=true;
    if N=1 then begin
        writeln('число не простое и не составное');
        exit;
    end;
    for d:=2 to N-1 do
        if N mod d =0 then p:=false;
    if p then writeln('число простое') else writeln('число составное');
End.
```

```
1
число не простое и не составное
2
число простое
3
число простое
4
число составное
```

Строка: 7 Столбец: 21

Используем переменную p логического типа для определения простоты числа: первоначально p принимает значение **true**, но если число N имеет хотя бы один делитель, то p принимает значение **false**. d – для обозначения делителей от 2 до $N-1$

Задача

Вводится натуральное число N ($N \leq 2\,147\,483\,647$).
Определить, является ли оно **простым** или **составным**.

Выполним эту программу, при

1. $N=123$ Ответ: число составное
2. $N=131071$ Ответ: число простое
3. $N=524387$ Ответ: число простое
4. $N= 536870911$ Ответ: число составное
5. $N= 1999999991$ Ответ: ?
6. $N= 2147483647$ Ответ: ?

← Вы дождались ответа?

←
← Вы дождётесь ответа?

Для уменьшения времени работы программы нужно что-то придумать !

Что можно заметить?

В приведенной программе мы делили N на числа от 2 до $N-1$, т.е. мы искали делители числа N , не равные 1 и меньше N . На самом же деле можно число N проверять на делимость не всех чисел, а только от 2 до $N/2$. **Почему?**

команду **for d:=2 to N-1 do ...**

заменим на **for d:=2 to N div 2 do ...**

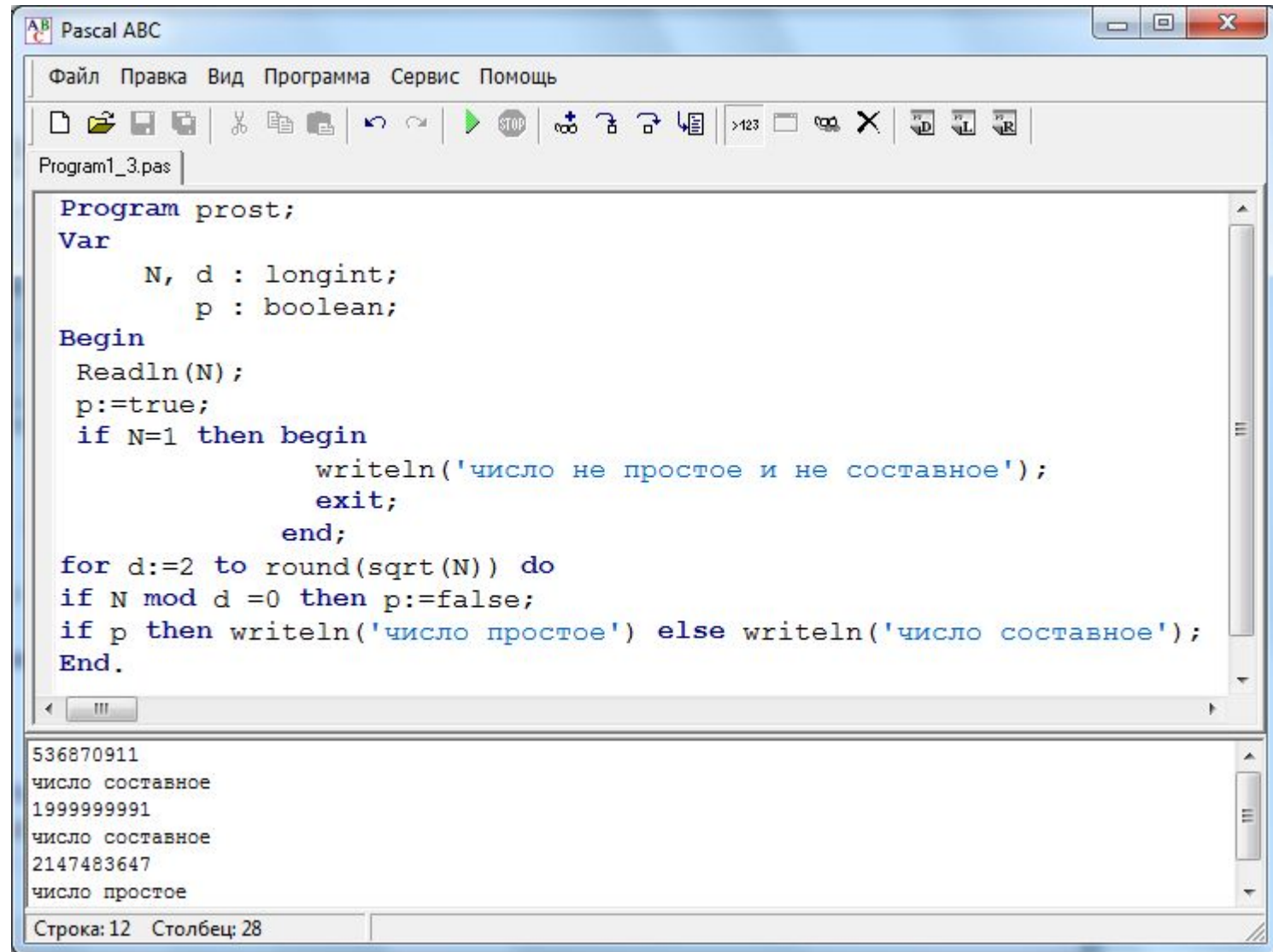
Таким образом, мы увеличили скорость выполнения программы в 2 раза, но тесты $4 - 6$ всё равно выполняются долго.

Что ещё можно заметить?

Пусть число N – составное. Тогда его можно представить в виде $N=x \cdot y$, где x и y больше 1, а меньшее из них обязательно не больше \sqrt{N} . Понятно, что если N делится на x , то N делится и на y . Если N не делится ни на одно число от 2 до $\text{round}(\text{sqrt}(N))$, значит оно не делится ни на какое другое число, т.е. оно простое.

команду `for d:=2 to N div 2 do ...`
заменим на `for d:=2 to round(sqrt(N)) do ...`

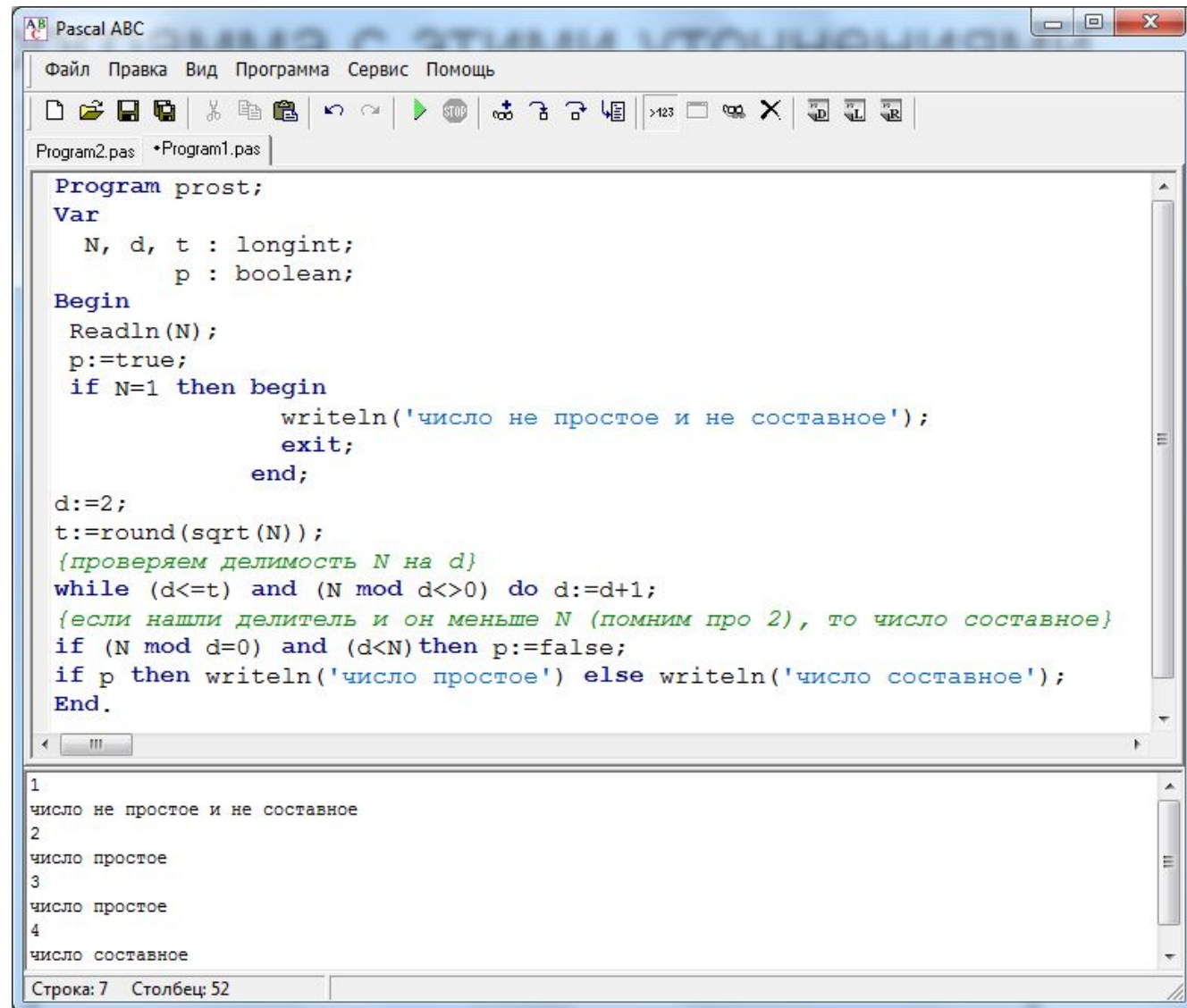
Выполнение программы для тестов 4, 5, 6



```
Program1_3.pas  
  
Program prost;  
Var  
    N, d : longint;  
    p : boolean;  
Begin  
    Readln(N);  
    p:=true;  
    if N=1 then begin  
        writeln('число не простое и не составное');  
        exit;  
    end;  
    for d:=2 to round(sqrt(N)) do  
        if N mod d =0 then p:=false;  
    if p then writeln('число простое') else writeln('число составное');  
End.  
  
536870911  
число составное  
1999999991  
число составное  
2147483647  
число простое  
Строка: 12 Столбец: 28
```

Улучшение программы

Алгоритм можно ещё улучшить, если останавливаться сразу после того, когда установили, что число N разделилось на какое-либо число d . Для этого заменим цикл **for** на **while**.



```
Program2.pas • Program1.pas
Program prost;
Var
  N, d, t : longint;
  p : boolean;
Begin
  Readln(N);
  p:=true;
  if N=1 then begin
    writeln('число не простое и не составное');
    exit;
  end;
  d:=2;
  t:=round(sqrt(N));
  {проверяем делимость N на d}
  while (d<=t) and (N mod d<>0) do d:=d+1;
  {если нашли делитель и он меньше N (помним про 2), то число составное}
  if (N mod d=0) and (d<N) then p:=false;
  if p then writeln('число простое') else writeln('число составное');
End.
```

```
1
число не простое и не составное
2
число простое
3
число простое
4
число составное
Строка: 7 Столбец: 52
```

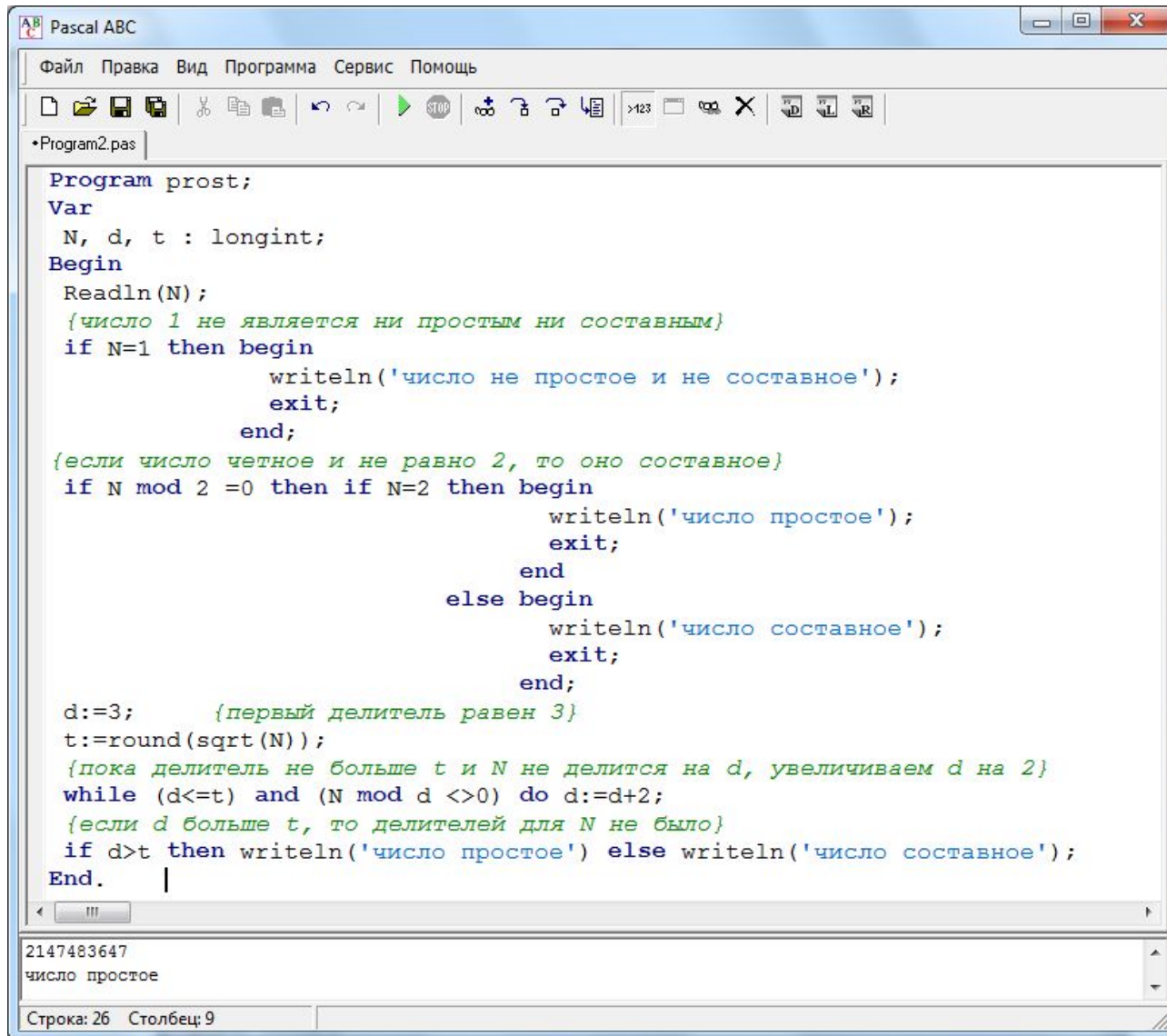
А можно ли ещё улучшить время выполнения программы?

Мы уменьшили количество проверок делимости сначала в 2 раза, а затем приблизительно в \sqrt{N} раз.

Можно ли ещё улучшить алгоритм?

Ещё в самом начале было замечено, что «среди простых чисел только одно чётное число **2**. Остальные числа нечетные». Ясно также, что достаточно проверить делимость только на нечетные **d**.

Программа с проверкой на делимость d нечетных



```
Program prost;
Var
  N, d, t : longint;
Begin
  Readln(N);
  {число 1 не является ни простым ни составным}
  if N=1 then begin
    writeln('число не простое и не составное');
    exit;
  end;
  {если число четное и не равно 2, то оно составное}
  if N mod 2 = 0 then if N=2 then begin
    writeln('число простое');
    exit;
  end
  else begin
    writeln('число составное');
    exit;
  end;
  d:=3;      {первый делитель равен 3}
  t:=round(sqrt(N));
  {пока делитель не больше t и N не делится на d, увеличиваем d на 2}
  while (d<=t) and (N mod d <>0) do d:=d+2;
  {если d больше t, то делителей для N не было}
  if d>t then writeln('число простое') else writeln('число составное');
End.
```

2147483647
число простое

Строка: 26 Столбец: 9

Вопросы по теме:

1. Какое число называется простым?
2. Есть ли среди простых чисел четные?
3. Назовите самое маленькое простое число?
4. Сколько однозначных простых чисел?
5. К каким числам относится число 1?
6. Число 29 является простым? Какое самое близкое к нему будет простым?
7. Число 49 является простым? Какое самое близкое к нему будет простым?
8. Запишите в тетрадь следующие числа:

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37

38 39 40 41 42 43 44 45 46 47 48 49 50

Зачеркните все составные числа.

9. Какой алгоритм Вы будете использовать для определения является ли введенное натуральное число простым или составным? Опишите его.

Литература:

1. В.М. Котов, И.А. Волков, А.И. Лапо «Информатика. Методы алгоритмизации», учебное пособие 8-9 классов, Минск «Народная асвета», 2000
2. И.Н. Порублев, А.Б. Ставровский «Алгоритмы и программы. Решение олимпиадных задач», Москва ООО «И.Д.Вильямс», 2007