Решение задач в среде Delphi

Тема: Использование алгоритмической конструкции
«Счетный цикл и цикл с предусловием»
в задачах на целочисленную арифметику.

Тюкавина Татьяна Михайловна

Задача №1. Написать программу, в которой к числу 1022 слева и справа приписываются по одной цифре так, чтобы полученное шестизначное число делилось на 7, 8 и 9.

Математическая модель.

В заданном числе a1022b вместо символа a надо подобрать цифру 1 ÷9 (для левого разряда). Вместо символа b подобрать цифру 0 ÷9 (разряда единиц). Полученное X должно делится на 7*8*9=504 без остатка:

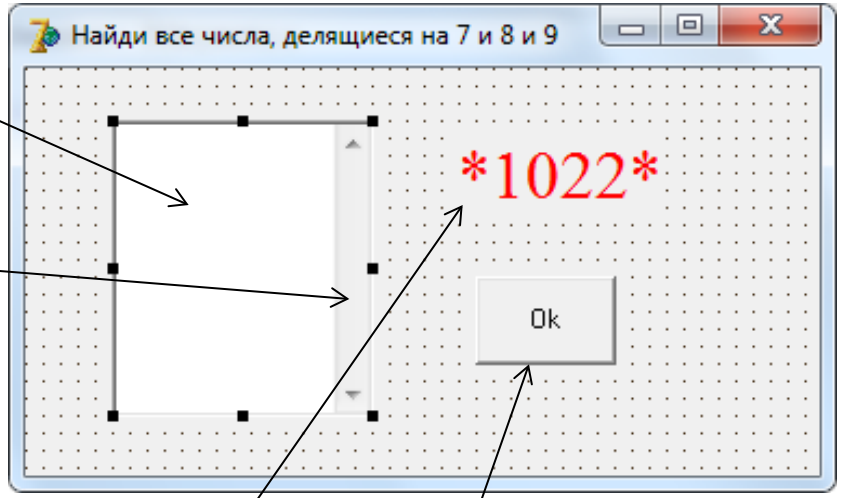
1. Искомое число X надо представить в десятичной позиционной системе счисления в виде X=100000*a + 10220 + b
2. Для получения всех вариантов числа X перебираем возможные значения a из диапазона [1; 9] и b из диапазона [0; 9]
3. В результат выводим только те X, которые делятся на 504 без остатка (используем операцию получение остатка от целочисленного деления X на 504)
4. Шаги 1 - 3 повторяем до тех пор, пока не переберём все варианты.

Описание данных

Название	Идентификатор в программе	Фактический смысл	Возможные значения Тип	Начальное значение
Исходные данные				
X	**x**	Искомое число	Integer	
Промежуточные данные (вспомогательные величины)				
Левая цифра	**a**	Множитель в 6-разряде числа X	Byte	
Правая цифра	**b**	Множитель в 1-разряде числа X	Byte	
Результат				
Вариант ответа на экране: 910224				

Конструирование окна программы и код программы

Многострочное текстовое поле Memo1 для ***вывода*** неизвестного количества вариантов искомого числа с вертикальной прокруткой, т.е. свойство ScrollBars задаем при конструировании ssVertical



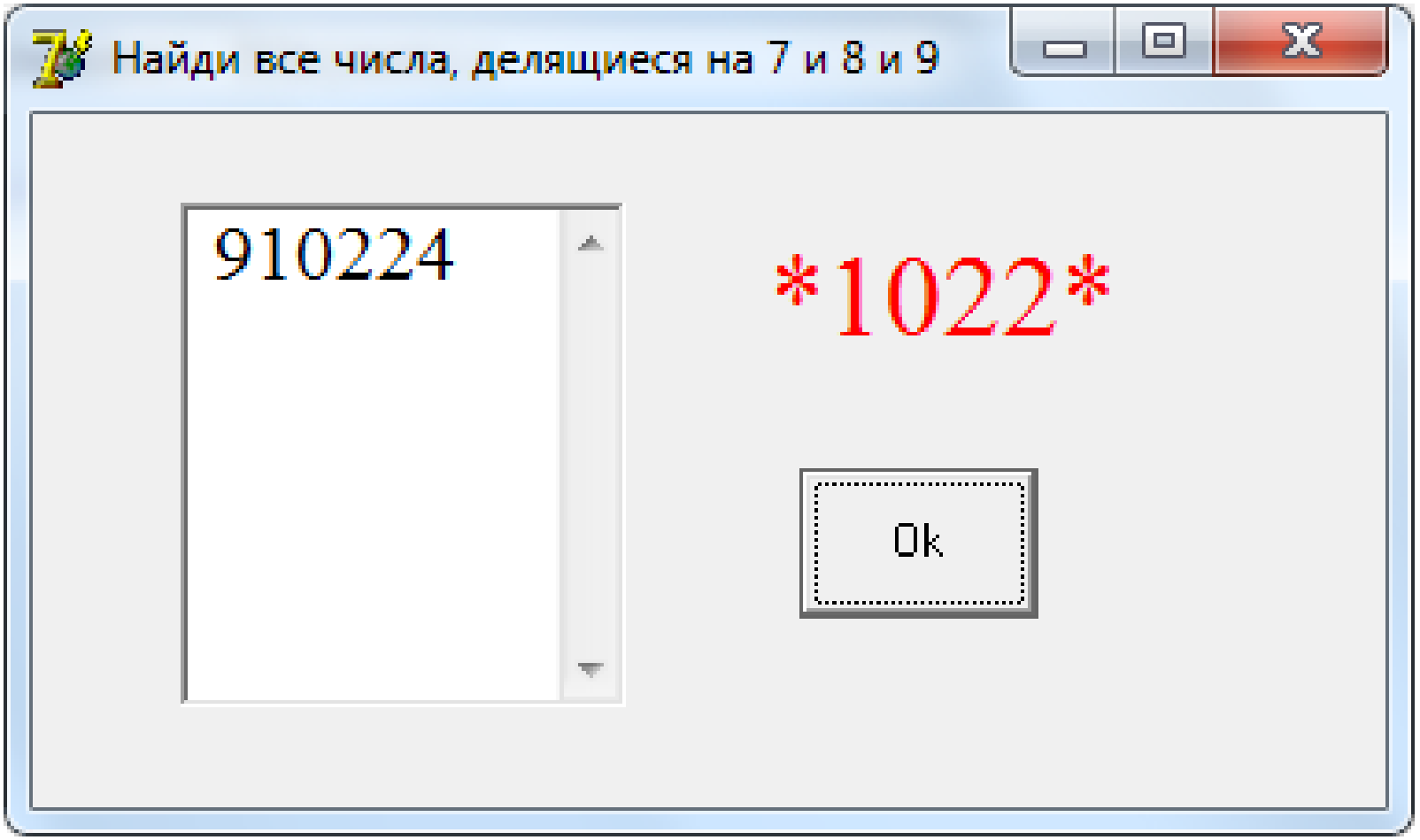

```
procedure TForm1.Button1Click(Sender: TObject);
Var a, b:  byte;
          x:  Integer;
Begin {1}
memo1.Clear; { очистка объекта Memo1}
 for  a:=1 to 9 do
    for b:=0 to 9 do
     begin {2}
     { 7*8*9=504 }
      x:=a*100000 + 10220 + b;
      if (x mod 504)=0 then   memo1.Lines.Add (IntToStr(x))
     end; {2}
end; {1}
```

Label1 - метка для визуализации

Button1 – командная кнопка для запуска событийной процедуры перебора цифр a и b

{ Для добавления возможных вариантов чисел X в объект **Memo1** для свойства **Lines** (тип String) используем функцию **Add** }

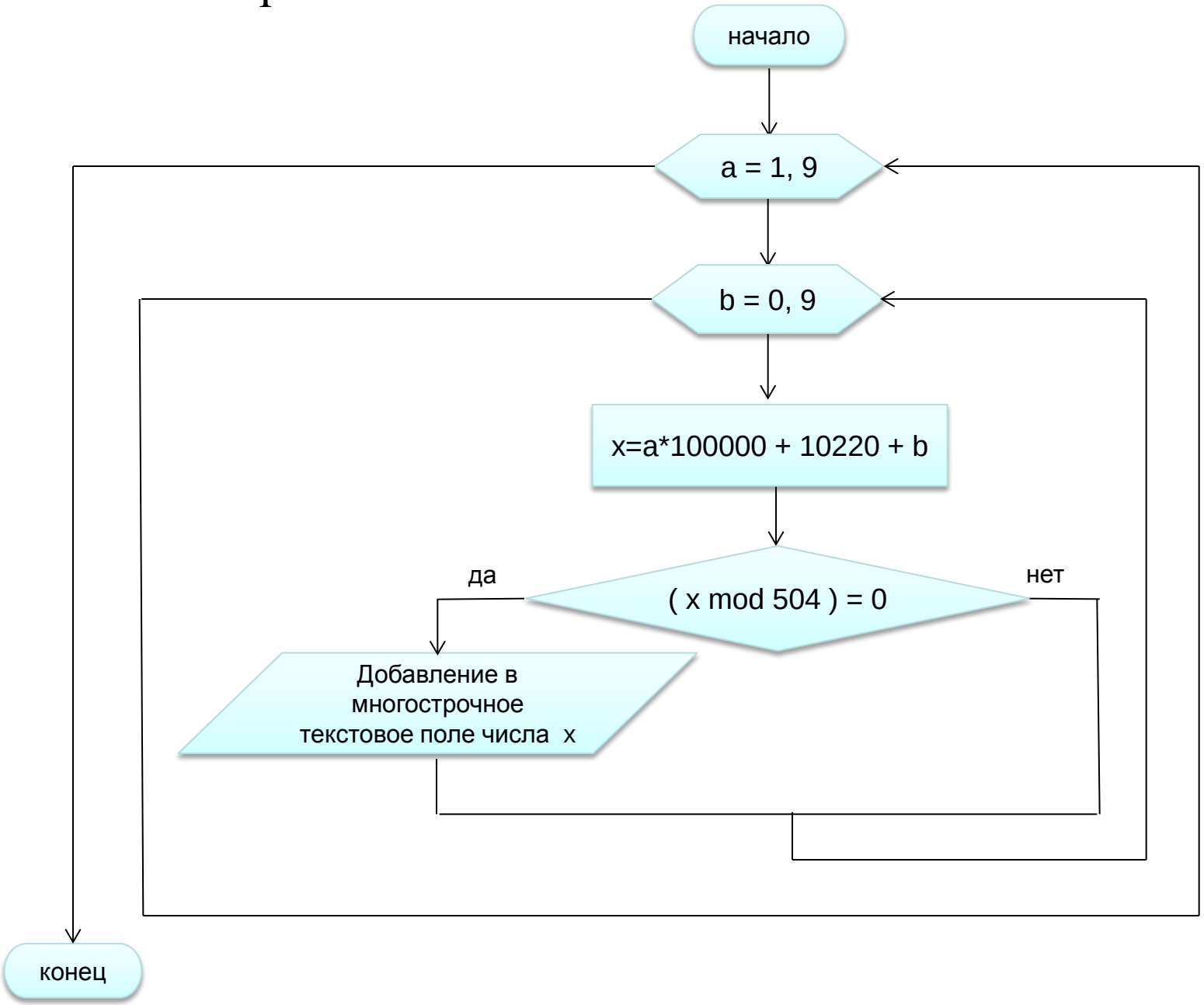
Тестовый пример



В результате работы программы

получено всего одно число=910224, удовлетворяющее условию делимости на 7 и 8 и 9

Схема алгоритма



Задача №2. Написать программу, определяющую, является ли дробь P/Q сократимой.

Математическая модель.

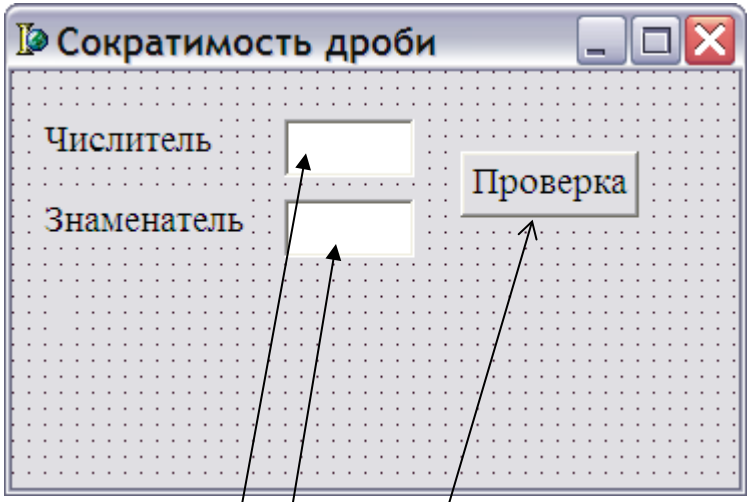
Дробь называется сократимой, если наибольший делитель числителя и знаменателя отличен от 1.

Описание данных

Название	**Идентификатор в программе**	**Фактический смысл**	**Возможные значения Тип**	**Начальное значение**
Исходные данные				
P	**P**	числитель	Integer	
Q	**Q**	знаменатель	≠0 Integer	
Промежуточные данные (вспомогательные величины)				
НОД	NOD	наибольший общий делитель	Integer	P
A	A		Integer	**Q**
Результат				
Вариант ответа на экране: при NOD<>1: Сократимая (NOD=2) при NOD=1: Не сократимая (NOD=1)				

Конструирование окна программы и код программы

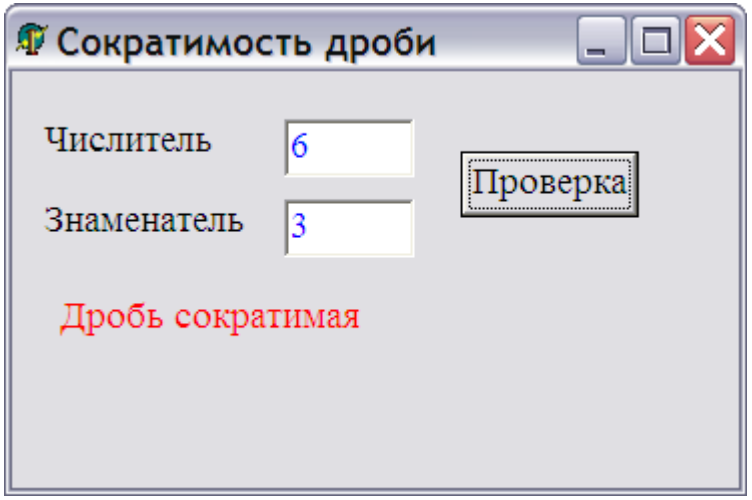
```
procedure TForm1.Button1Click(Sender: TObject);
Var P, Q, NOD, A : Integer;
begin
  P:=StrToInt(Edit1.Text);
  Q:=StrToInt(Edit2.Text);
  If  Q < > 0 Then
  begin  {ВЫЧИСЛЕНИЕ НОД}
       Label3.Caption:='Дробь сократимая';
       If  P < >Q  Then
       begin
          NOD:=P; A:=Q;
          While NOD<>A do
             If NOD <A Then A:=A-NOD  Else  NOD:=NOD-A;
             {ВЫВОД РЕЗУЛЬТАТОВ}
             If NOD=1 Then
                Label3.Caption:='Дробь не сократимая'
              Else
                Label3.Caption:='Дробь сократимая';
        end;
 end
 else
 Label3.Caption:='Введите знаменатель, не равный 0';
end;
```

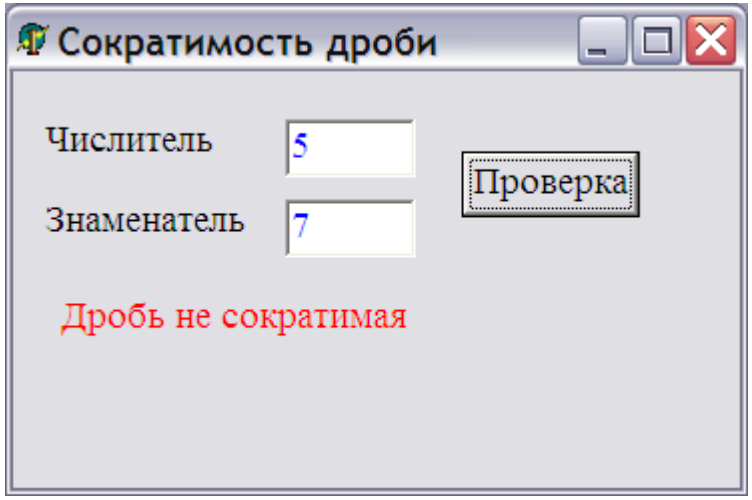


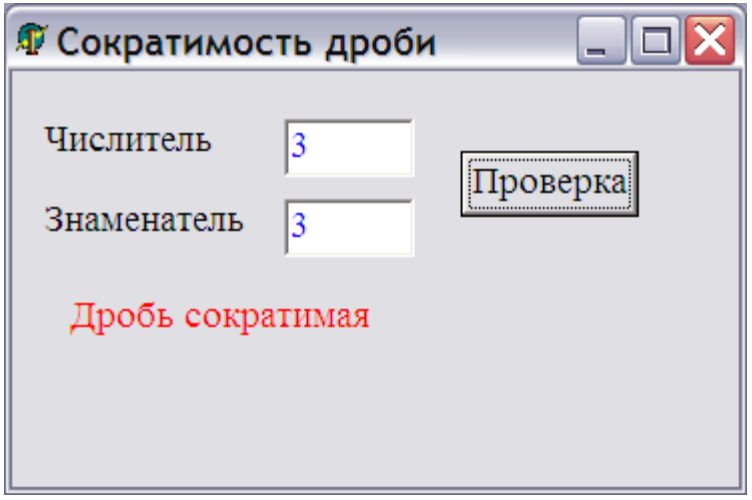
Edit1, Edit2 — Текстовые поля для ***ввода*** данных

Button1 - командная кнопка для запуска событийной процедуры проверки

Тестовые примеры







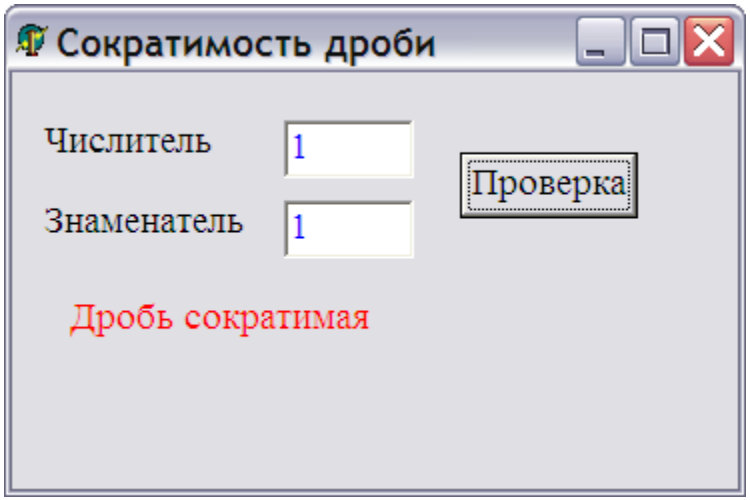
Нормальные ситуации:

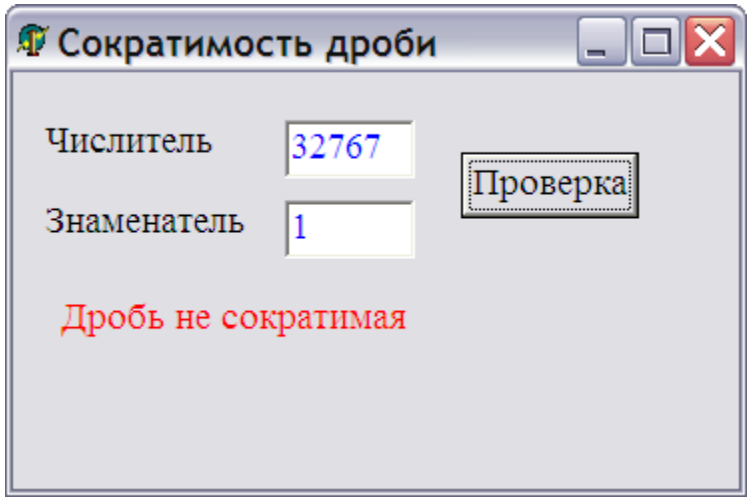
P=6, Q=3 Ответ: Сократимая.

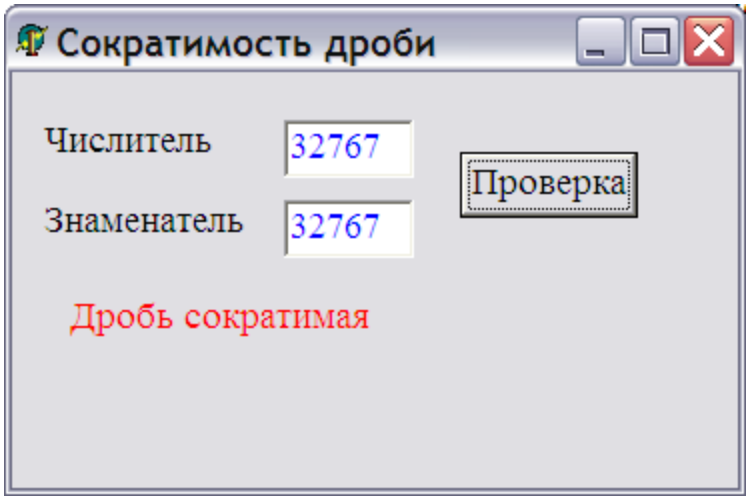
P=3, Q=3 Ответ: Сократимая.

P=5, Q=7 Ответ: Не сократимая.

Тестовые примеры







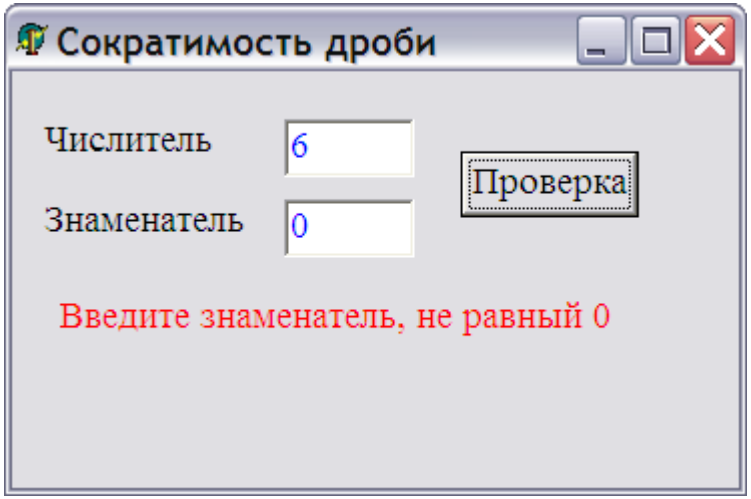
Экстремальные ситуации:

P=1, Q=1 Ответ: Сократимая.

P=32767, Q=32767 Ответ: Сократимая.

P=32767, Q=1 Ответ: Не сократимая.

Тестовые примеры

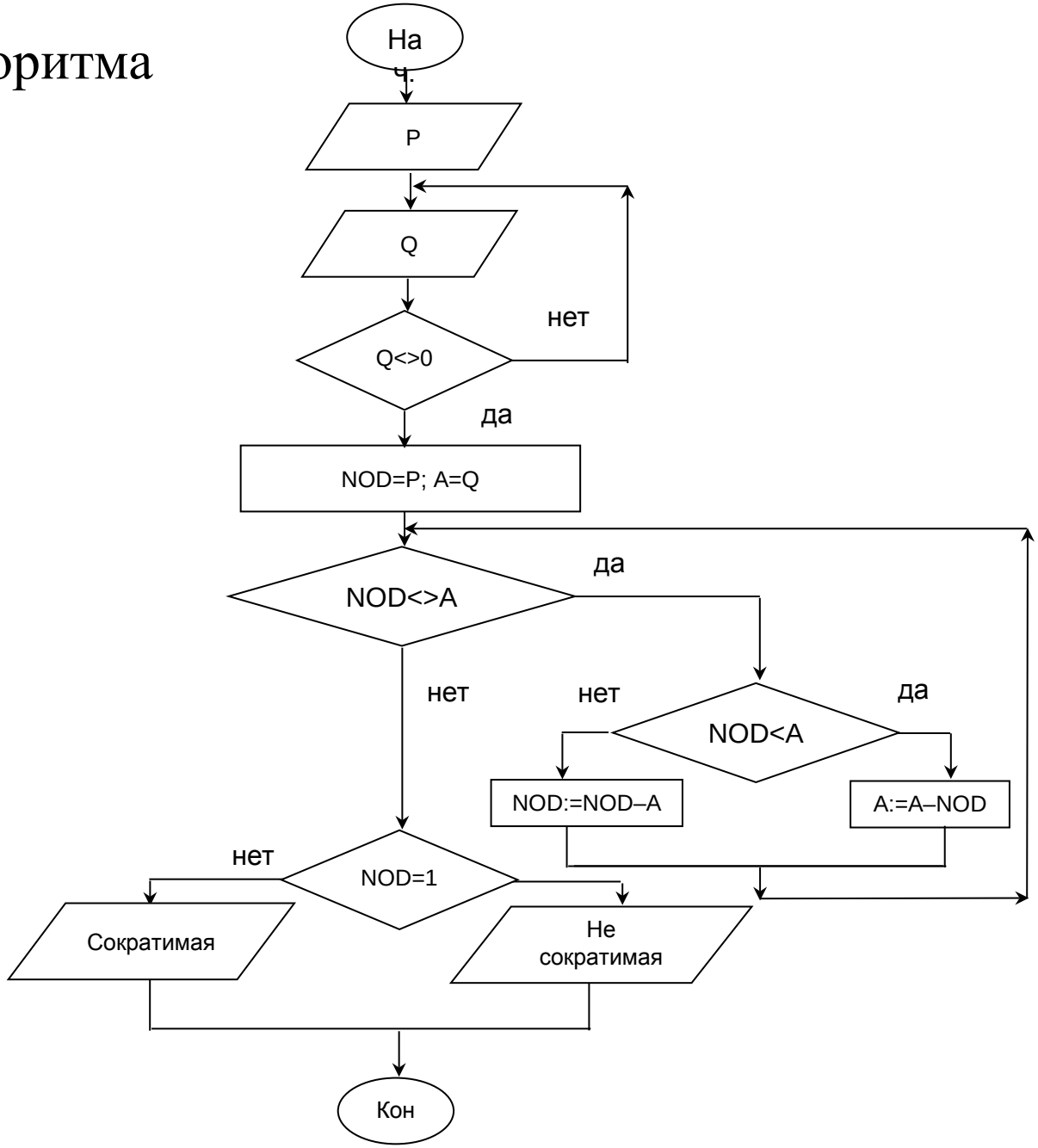


Исключительные ситуации:

P=6, Q=0

Должен быть повторный запрос на ввод.

Схема алгоритма



Задача №3. Написать программу, определяющую Наибольший Общий Делитель 2-х целых чисел.

Математическая модель.

Наибольшим общим делителем (**НОД**) для двух целых чисел ***m*** и ***n*** называется наибольший из их общих делителей.

У каждого натурального числа >1 имеются по крайней мере два натуральных делителя: 1 и само это число. При этом натуральные числа, имеющие ровно два делителя, называются простыми, а имеющие больше двух делителей — составными. Единица имеет ровно один делитель и не является ни простым, ни составным.

Пример: для чисел 70 и 105 наибольший общий делитель равен 35.

Наибольший общий делитель существует и однозначно определён, если числа ***m*** и ***n*** не равны нулю.

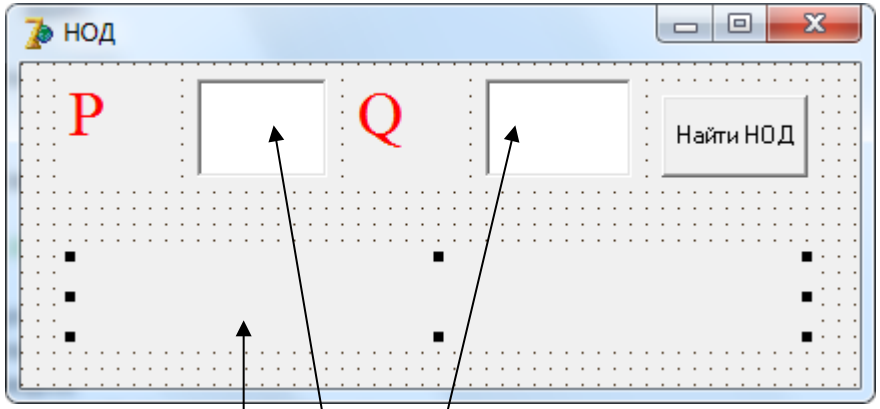
Алгори́тм Евкли́да — эффективный алгоритм для нахождения *НОД* двух целых чисел. В самом простом случае алгоритм Евклида применяется к паре положительных целых чисел и формирует новую пару, которая состоит из меньшего числа и разницы между большим и меньшим числом. Процесс повторяется, пока числа не станут равными. Найденное число и есть наибольший общий делитель исходной пары.

Описание данных

Название	**Идентификатор в программе**	**Фактический смысл**	**Возможные значения Тип**	**Начальное значение**
Исходные данные				
P	**P**	1- е число	Integer ≠0	**P**
Q	**Q**	2- е число	Integer ≠0	**Q**
Результат				
Вариант ответа на экране: для P=70 , Q= 105 НОД=35 или Введите числа, не равные 0				

Конструирование окна программы и код программы

```
procedure TForm1.Button1Click(Sender: TObject);
Var P, Q, : Integer;
begin
  P:=StrToInt(Edit1.Text);
  Q:=StrToInt(Edit2.Text);
  If  (Q < > 0) OR (P < > 0) Then
  begin  {ВЫЧИСЛЕНИЕ НОД}
    while P<>Q do
    begin
        if P>Q then P:=P-Q
               else Q:=Q-P;
    end;
 {ВЫВОД РЕЗУЛЬТАТОВ}
  label3.Caption:='НОД=' + IntToStr(P);
  end
  else
  Label3.Caption:='Введите чисела, не равное 0';
end;
```



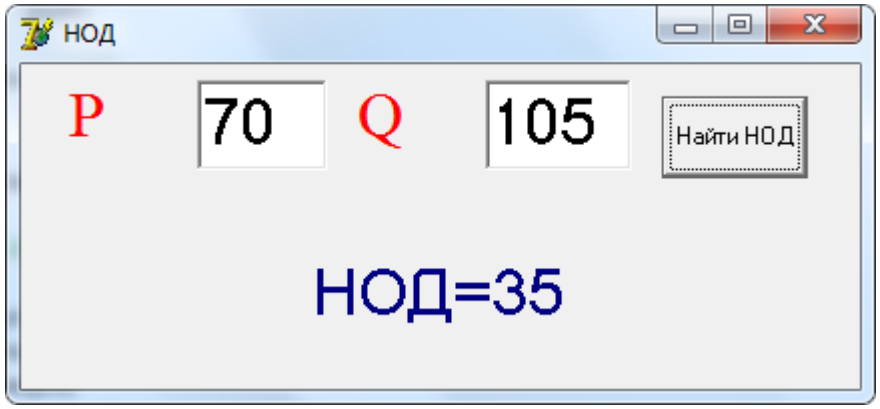
Edit1

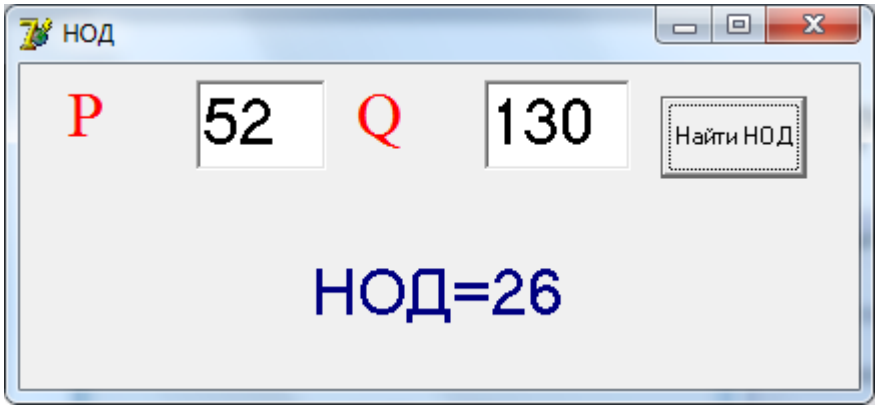
Edit2

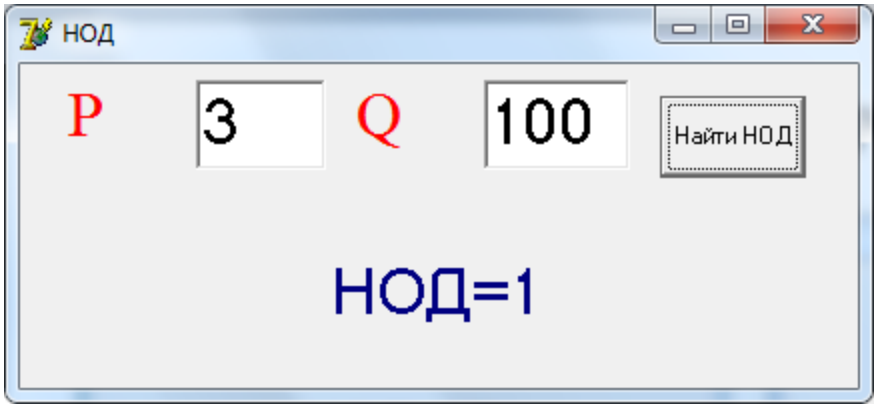
Текстовые поля для ***ввода*** данных

Label3 – метка для вывода результата

Тестовые примеры







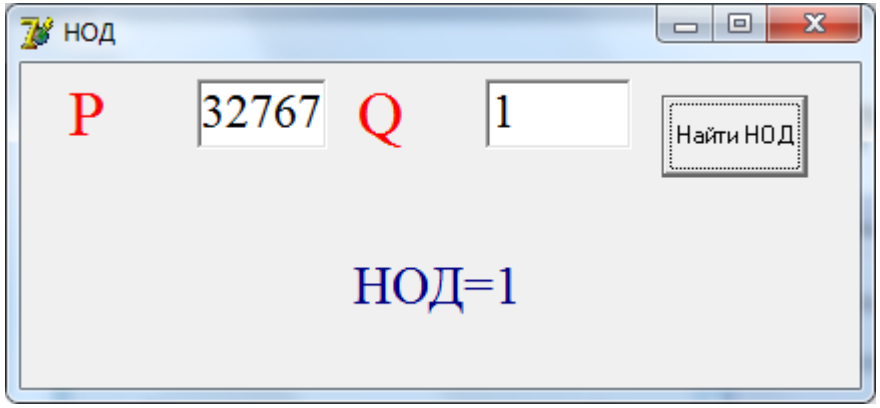
Нормальные ситуации:

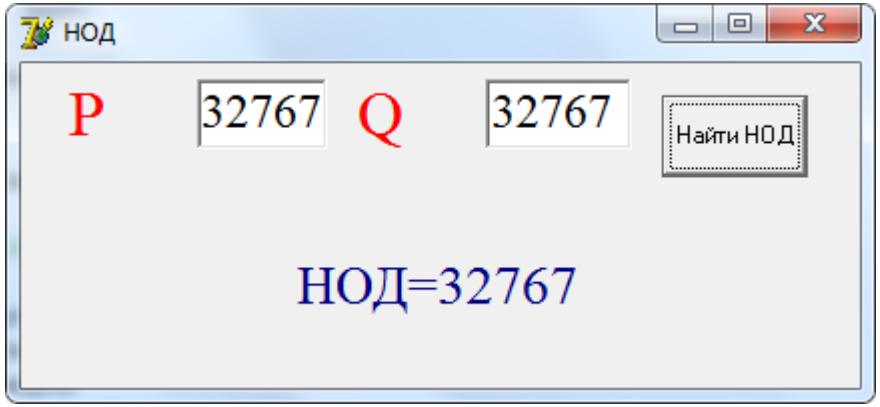
P=70, Q=105 Ответ: НОД=35

P=52, Q=130 Ответ: НОД=26

P=3, Q=100 Ответ: НОД=1

Тестовые примеры



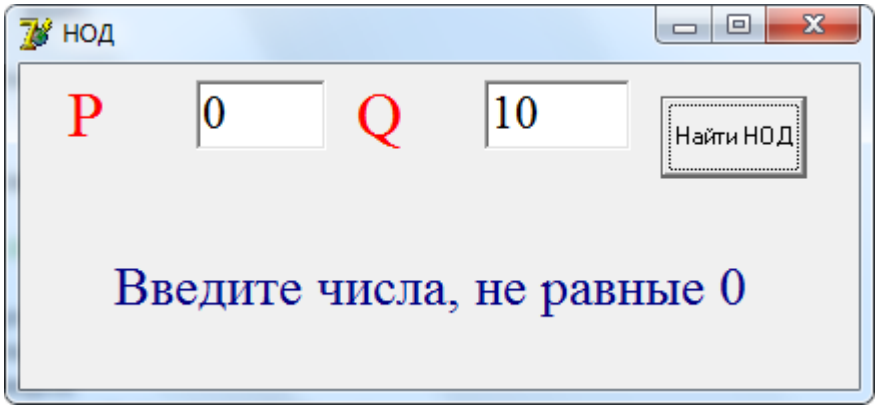


Экстремальные ситуации:

P=32767, Q=1 Ответ: НОД=1

P=32767, Q=32767 Ответ: НОД=32767

Тестовые примеры

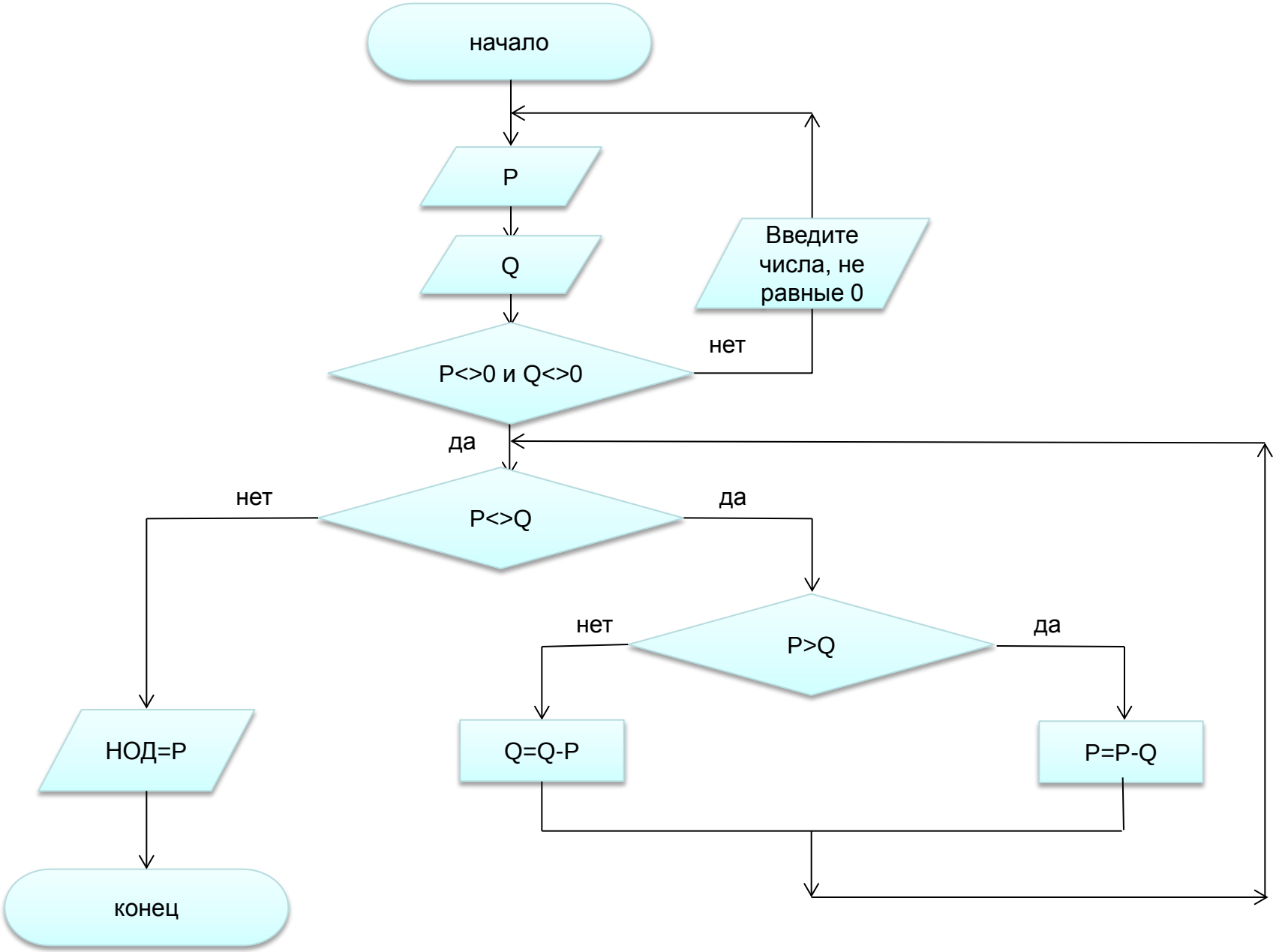


Исключительные ситуации:

P=0, Q=10

Ответ: Введите числа, не равные 0

Схема алгоритма



Задача №4. Написать программу, определяющую, сумму цифр натурального числа, заданного пользователем.

Математическая модель.

В заданном натуральном числе надо выделить каждую цифру и добавить её в общую сумму:

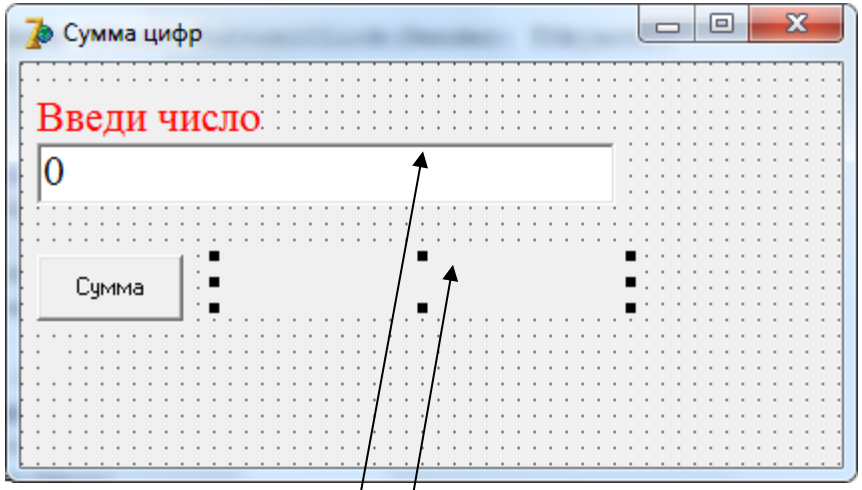
1. Для выделения последней цифры числа используем операцию получение остатка от целочисленного деления на 10.
2. Для отбрасывания последней цифры числа используем операцию целочисленного деления на 10
3. Шаги 1 и 2 повторяем до тех пор, пока результат 2 шага ≠0

Описание данных

Название	**Идентификатор в программе**	**Фактический смысл**	**Возможные значения Тип**	**Начальное значение**
Исходные данные				
X	**x**	Вводимое число	Integer <10 цифр Int64 <19 цифр	**X**
S	**s**	Сумма цифр	Byte	
Промежуточные данные (вспомогательные величины)				
N	n	Копия числа X	Integer <10 цифр Int64 <19 цифр	**X**
c	c	Выделенная цифра	Byte	
Результат				
Вариант ответа на экране: при X=12345: Сумма цифр=15				

Конструирование окна программы и код программы

```
procedure TForm1.Button1Click(Sender: TObject);
Var s, c:  Byte;
     x, n: Integer;
Begin {1}
   if length(edit1.Text)<10 then {результат
   компьютерного эксперимента}
   Begin {2}
        x:=StrToInt(edit1.Text);
        n:=x;
        s:=0;
        while n<>0 do
        begin {3}
              c:=n mod 10;
              s:=s+c;
              n:=n div 10;
        end; {3}
        label2.Caption:= ‘Сумма цифр=’ + IntToStr(s);
    End {2}
    else label2.Caption:=‘ Слишком длинное число';
end; {1}
```



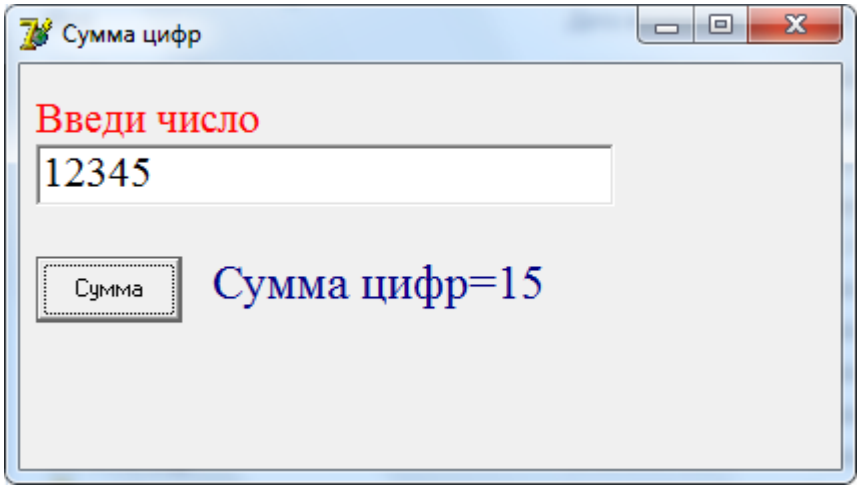
Edit1 — Текстовое поля для ***ввода*** числа

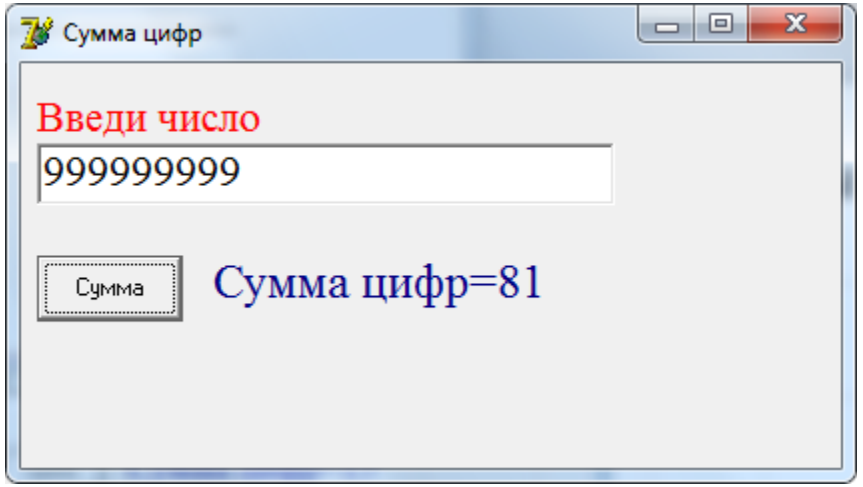
Label2 — метка для вывода результата

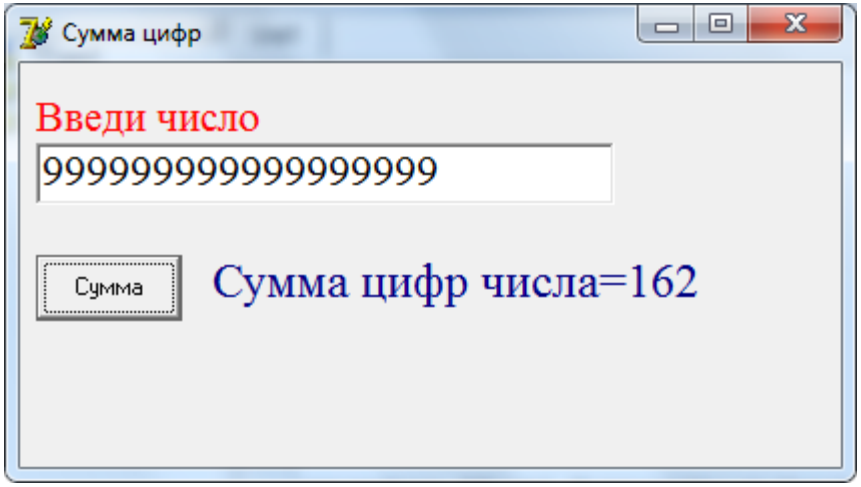
Код программы для типа Int64

```
procedure TForm1.Button1Click(Sender: TObject);
Var s, c:  Byte;
     x, n: Int64;

Begin {1}
   if length(edit1.Text)<19 then {результат компьютерного эксперимента}
   Begin {2}
        x:=StrToInt64(edit1.Text);
        n:=x;
        s:=0;
        while n<>0 do
        begin {3}
              c:=n mod 10;
              s:=s+c;
              n:=n div 10;
        end; {3}
        label2.Caption:= ‘Сумма цифр=’ + IntToStr(s);
     End {2}
     else label2.Caption:=‘ Слишком длинное число';
end; {1}
```

Тестовые примеры







Нормальные ситуации:

X=12345 Ответ: Сумма цифр=15

для варианта Integer

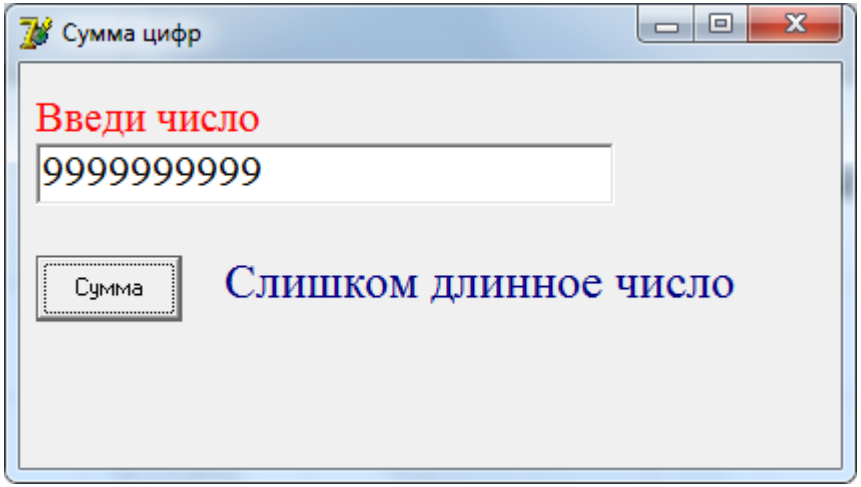
X=999999999 Ответ: Сумма цифр=81

для варианта Int64

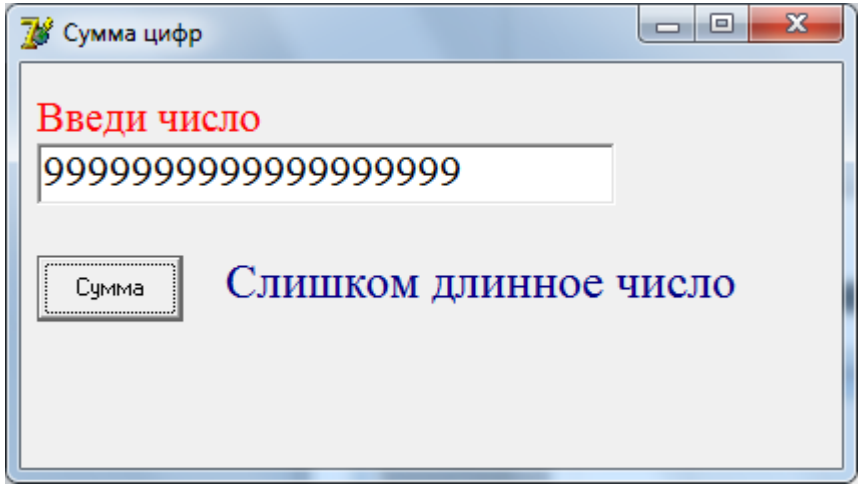
X=9999999999999999

Ответ: Сумма цифр=162

Тестовые примеры



вариант Integer



вариант Int64

Экстремальные ситуации:

для варианта Integer

X=9999999999

Ответ: Слишком длинное число

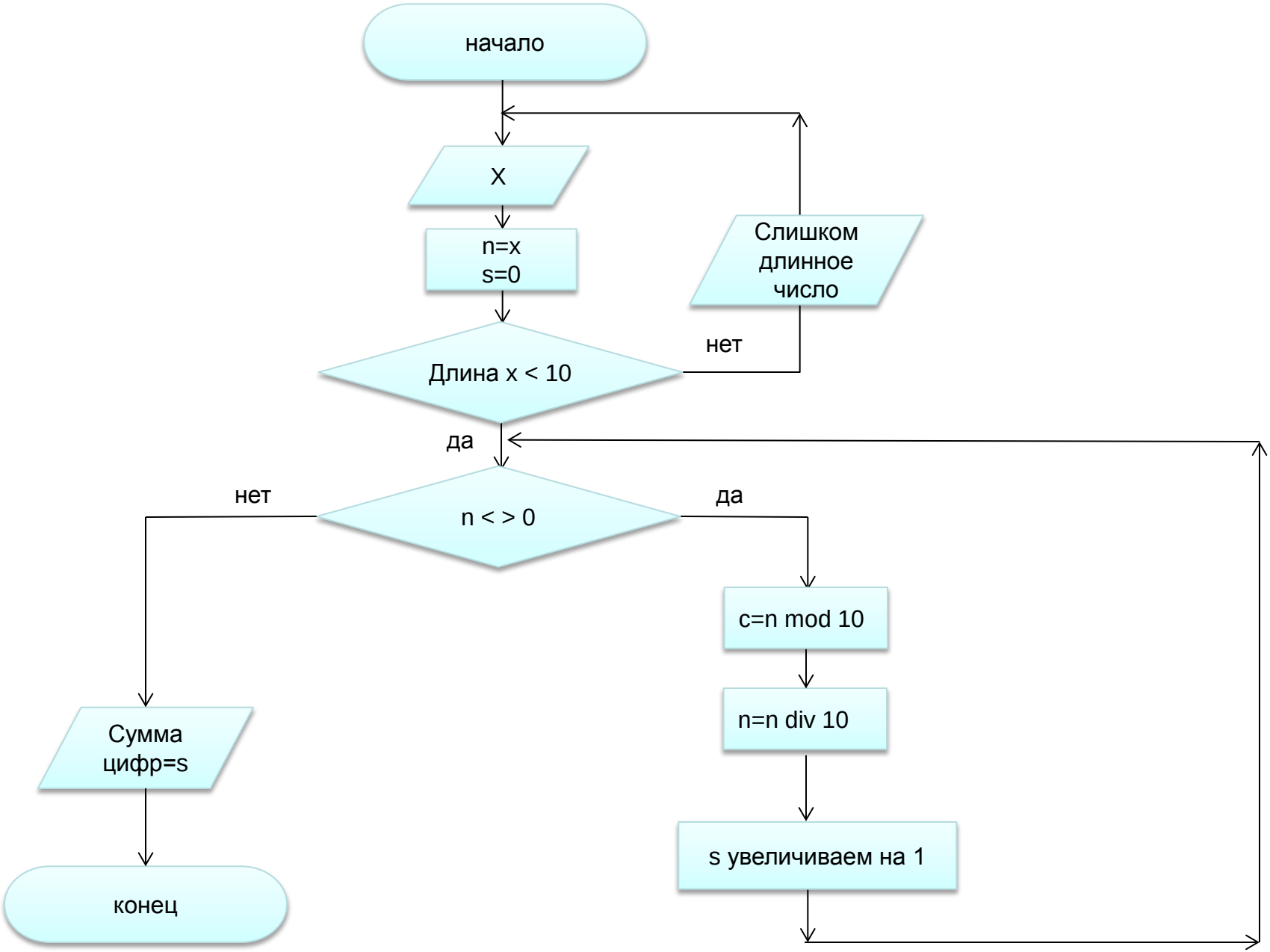
Экстремальные ситуации:

для варианта Int64

X=99999999999999999

Ответ: Слишком длинное число

Схема алгоритма



Задача №5. Написать программу, определяющую максимальную цифру натурального числа, заданного пользователем.

Математическая модель.

В заданном натуральном числе надо выделить каждую цифру и определить максимальную из них:

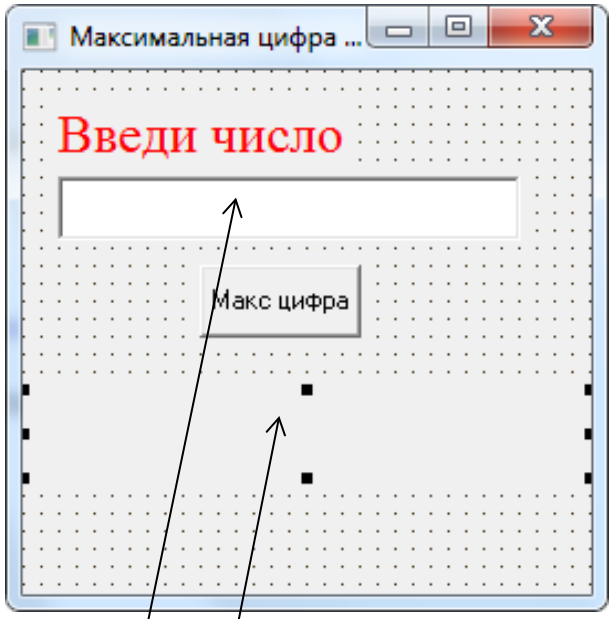
1. Предположим, что максимальная цифра в числе = 0 и присвоим текущему значению максимума это значение
2. Для выделения последней цифры числа используем операцию получение остатка от целочисленного деления на 10
3. Для отбрасывания последней цифры числа используем операцию целочисленного деления на 10
4. Сравниваем выделенную цифру с текущим значением максимума и, если она превышает это значение, переопределяем его.
5. Шаги 2 - 4 повторяем до тех пор, пока результат 3 шага ≠0

Описание данных

Название	Идентификатор в программе	Фактический смысл	Возможные значения Тип	Начальное значение
Исходные данные				
N	**n**	Вводимое число	Int64 <19 цифр	**N**
m	**m**	Максимальная цифра	Byte	**0**
Промежуточные данные (вспомогательные величины)				
c	c	Выделенная цифра	Byte	
Результат				
Вариант ответа на экране: при X=10845: Максимальная цифра=8				

Конструирование окна программы и код программы

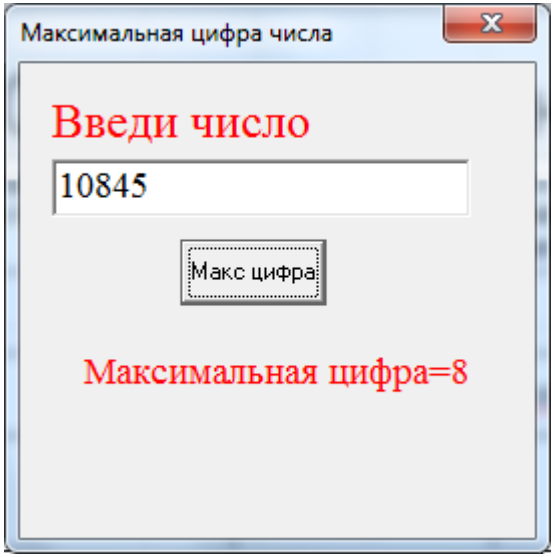
```
procedure TForm1.Button1Click(Sender: TObject);
Var  c, m:  Byte;
          n:  Int64;


Begin {1}
    if length(edit1.Text) < 19 then {результат
    компьютерного эксперимента}
    Begin {2}
          n:=StrToInt64(edit1.Text);
          m:=0;
          while n<>0 do
          begin {3}
                c:=n mod 10;
                if c>m then m:=c;
                n:=n div 10;
          end; {3}
          label2.Caption:= ‘Максимальная цифра=’ + IntToStr(m);
      End {2}
      else label2.Caption:=‘ Слишком длинное число';
end; {1}
```

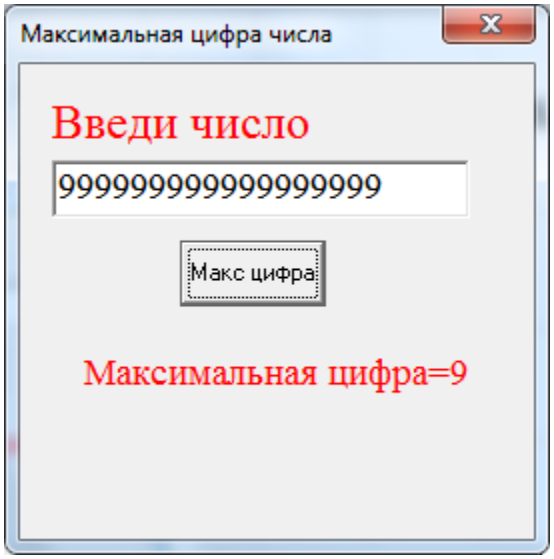


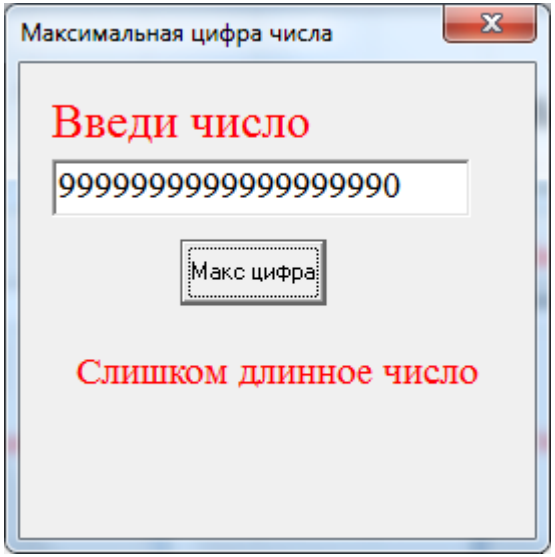
Edit1 — Текстовое поля для ***ввода*** числа

Label2 — метка для вывода результата

Тестовые примеры для варианта Int64







Нормальные ситуации:

X=10845 Ответ: Максимальная цифра=8

X=9999999999999999

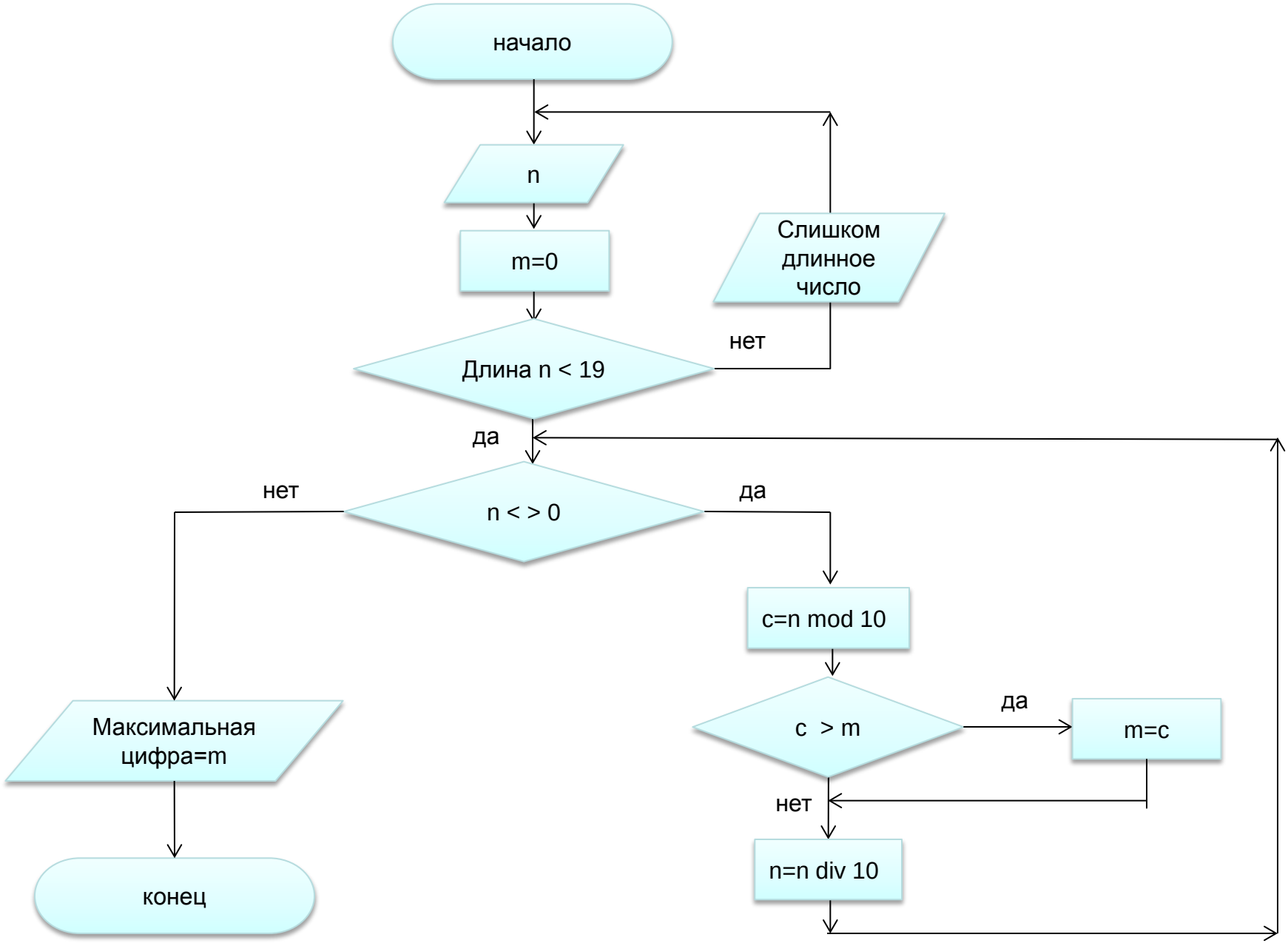
Ответ: Максимальная цифра=9

Экстремальные ситуации:

X=99999999999999990

Ответ: Слишком длинное число

Схема алгоритма



Задача №6. В слове МУХА закодировано 4-х-значное число (каждой букве соответствует цифра). Слову СЛОН тоже соответствует число. Написать программу, в которой подбираются варианты всевозможных чисел, таких, чтобы выполнялось равенство МУХА + МУХА = СЛОН.

Математическая модель.

Используем позиционную систему счисления. Для каждой буквы перебираем возможные комбинации неповторяющихся цифр (т.к в слове МУХА буквы не повторяются), таких, чтобы в удвоенном числе (соответствующем слову СЛОН) получились также неповторяющиеся цифры:

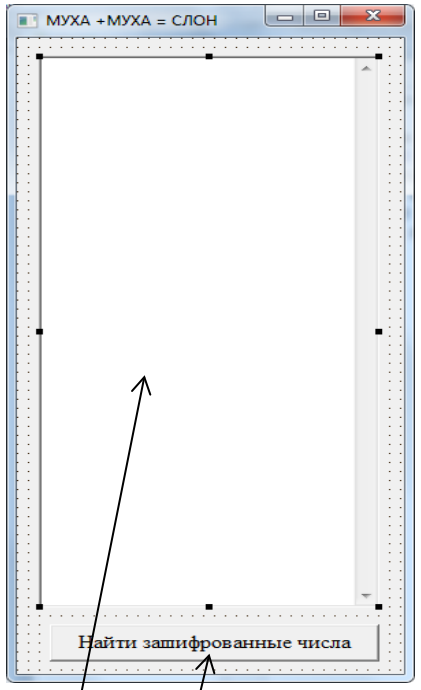
1. Левая цифра в числе muxa <=4, иначе левая цифра в числе clon получится 0 или перейдёт в следующий левый разряд, что не допустимо. Это отражается в вариантах перебора цифры m (счетный цикл для левой цифры от 1 до 4).
2. Для цифр y, x возможны варианты от 0 до 9
3. Для цифры a возможны варианты от 1 до 9
4. Для получения числа myxa используем позиционное представление:
   myxa = m*1000 + y*100 + x*10 +a
5. Тогда число clon = myxa + myxa
6. Для проверки условий на возможные варианты цифр в числе clon используем:
   1) выделение последней цифры числа (операция Mod на 10)
   2) отбрасывание цифр числа (операция Div на 10, на 100 или 1000)
7. Всего в задаче получается 8 цифр (m, y, x, a, s, l, o, n), которые сохраняем в массиве, чтобы проверить все комбинации и исключить повторы
8. Числа, удовлетворяющие всем условиям, выводим в результат

Описание данных

Идентификатор в программе	**Фактический смысл** (закодированное слово)	**Возможные значения Тип**	**Начальное значение**
myxa	МУХА	Integer	
clon	СЛОН	Integer	
m	Выделенная цифра	Byte	
y	Выделенная цифра	Byte	
x	Выделенная цифра	Byte	
a	Выделенная цифра	Byte	
c	Выделенная цифра	Byte	
l	Выделенная цифра	Byte	
o	Выделенная цифра	Byte	
n	Выделенная цифра	Byte	
num	Порядковый номер найденного варианта	Integer	0
i	Вспомогательная переменная =шаг цикла	Byte	1 to 8
j	Вспомогательная переменная =шаг цикла	Byte	1 to 8
p	Массив выделенных цифр для перебора (исключения повторов), размерность [1..8]	Byte	
pr	Признак отсутствия повторов при переборе всех 8 цифр	Boolean	true

Конструирование окна программы и код программы

```
procedure TForm1.Button1Click(Sender: TObject);
var pr :boolean;
    m, y, x, a, c, l, o, n, i, j :byte;
    num, myxa, clon :integer;
    p :array[1..8] of byte;
begin
  memo1.Clear ;
  num:=0 ;
  for m :=1 to 4 do
     for y :=0 to 9 do
       if m < > y then    for x :=0 to 9 do
          if (x < > y) and (x < > m) then    for a :=1 to 9 do
            if (a < > m) and (a < > y) and (a < > x) then
              begin
                myxa := m * 1000 + y * 100 + x * 10 + a ;
                clon := myxa + myxa ;
                c := clon div 1000 ;
                l := (clon div 100) mod 10 ;
                o := (clon div 10) mod 10 ;
                n :=clon mod 10 ;
                p[1] :=m; p[2] :=y; p[3] :=x; p[4] :=a; p[5] :=c; p[6] :=l; p[7] :=o; p[8] :=n;
                pr :=true;
                for i :=1 to 8 do
                   for j :=1 to 8 do
                     if i < > j then if p[i] = p[j] then pr :=false;
                if pr then
                begin
                  num := num +1;
                  memo1.Lines.Append ( IntToStr (num) +  ') МУХА = ' + IntToStr ( myxa )  + ' СЛОН= ' + IntToStr ( clon ) );
                end;
           end;
end;
```



Memo1 - поле для вывода результата с вертикальной прокруткой

Button1 – командная кнопка для запуска процесса

Результат работы программы

Окно вывода результатов показано не полностью.
Написать программу, отладить и найти общее количество закодированных чисел.
Показать результат преподавателю.