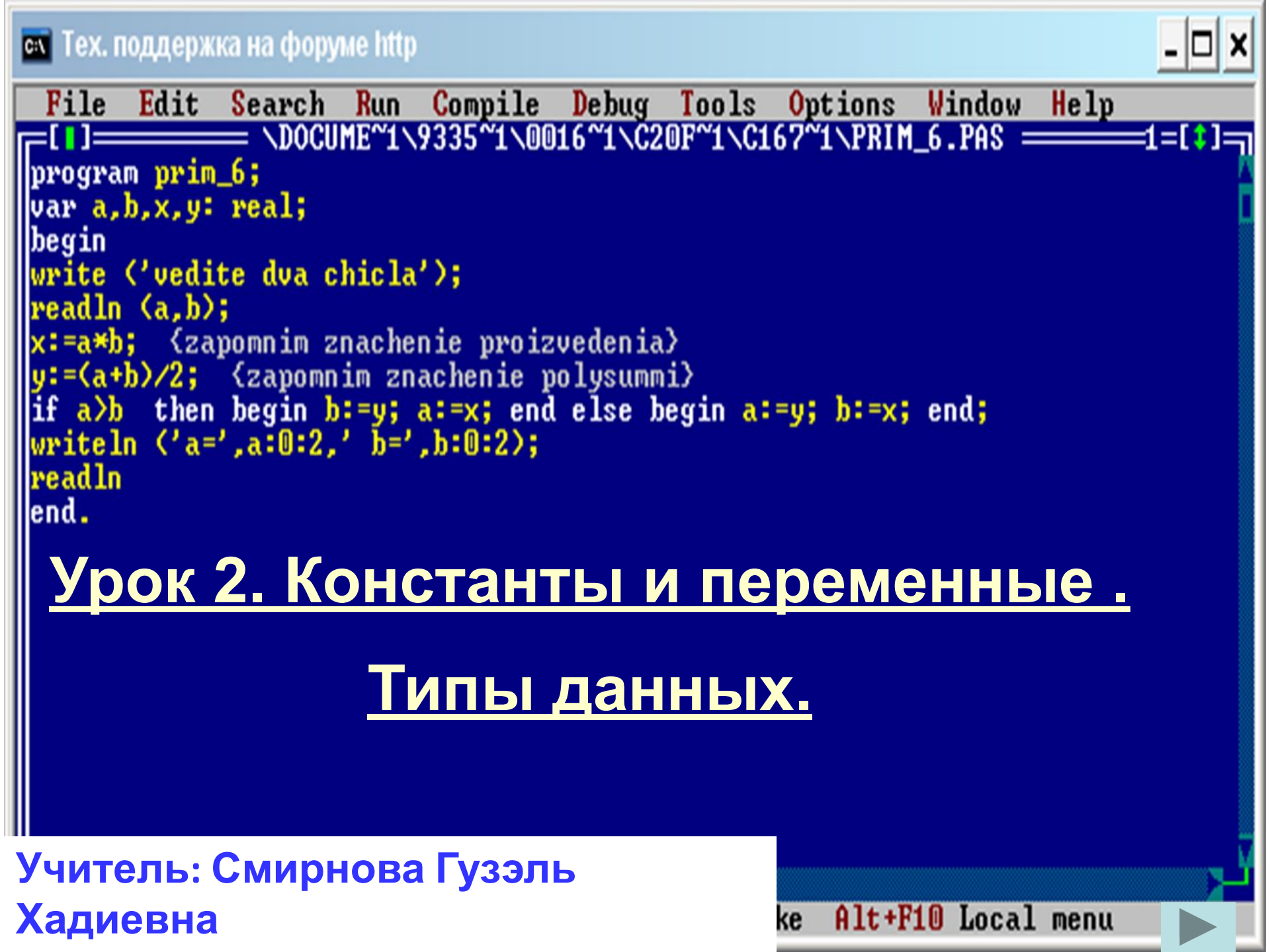


Урок 2. Константы и переменные .

Типы данных.

Учитель: Смирнова Гузэль Хадиевна

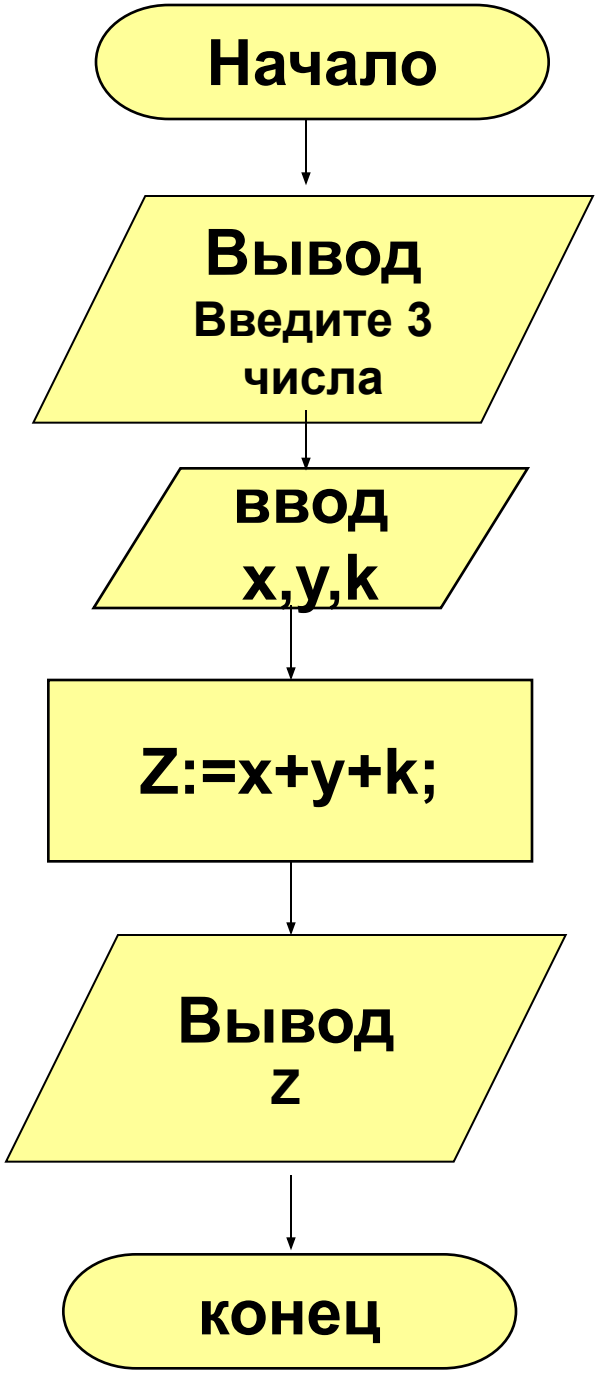
Урок 2.

Сегодня, пожалуй, самый теоретически насыщенный урок. Но без основных понятий и правил обойтись невозможно. Вашу программу будет выполнять машина, а она все делает буквально. Поэтому правила написания программы очень жестки, но их немного. Одно утешает, что правила русского языка намного сложнее.

Для начала рассмотрим задачу:

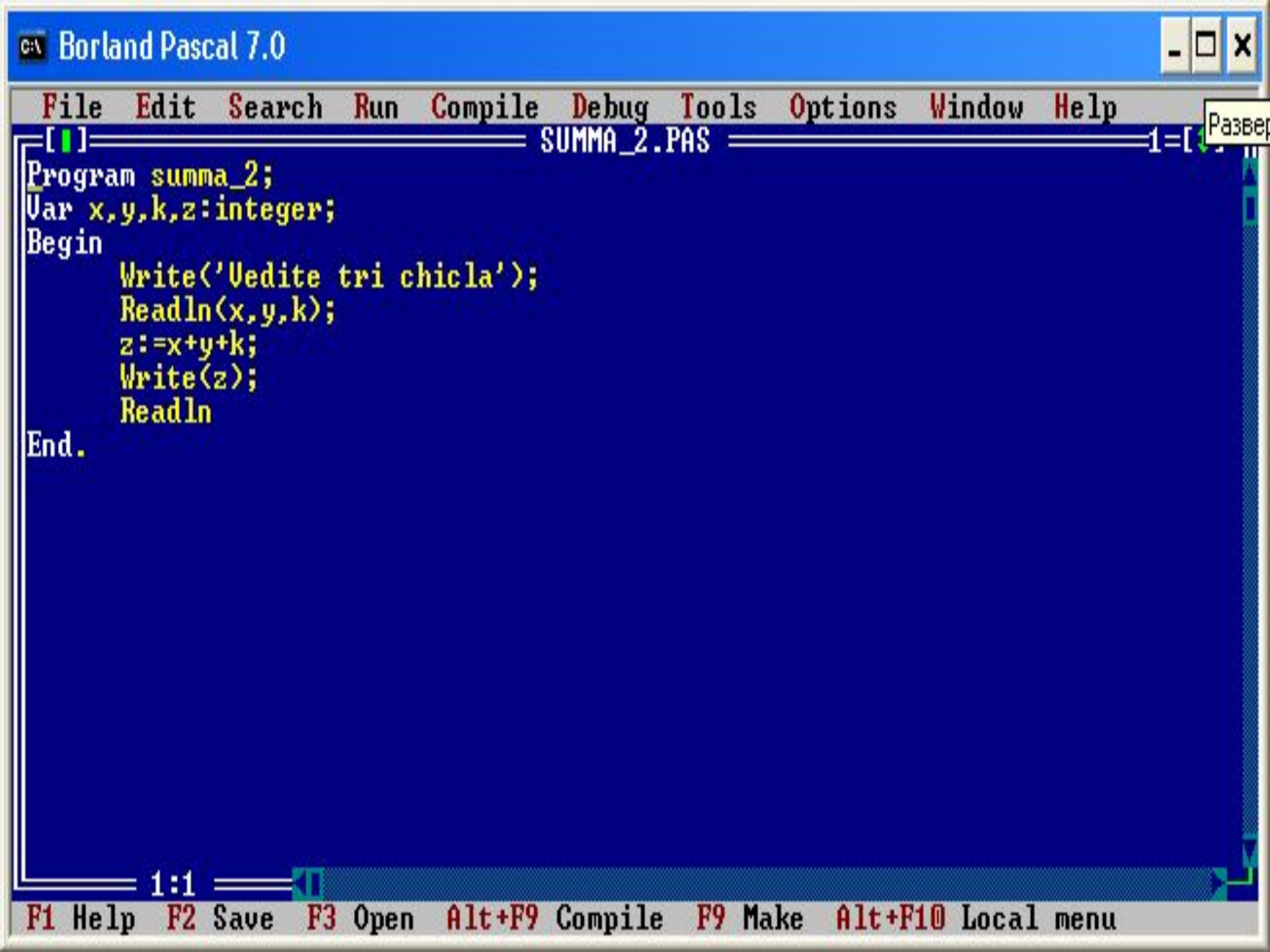
Напишите программу вычисления суммы 3 чисел.

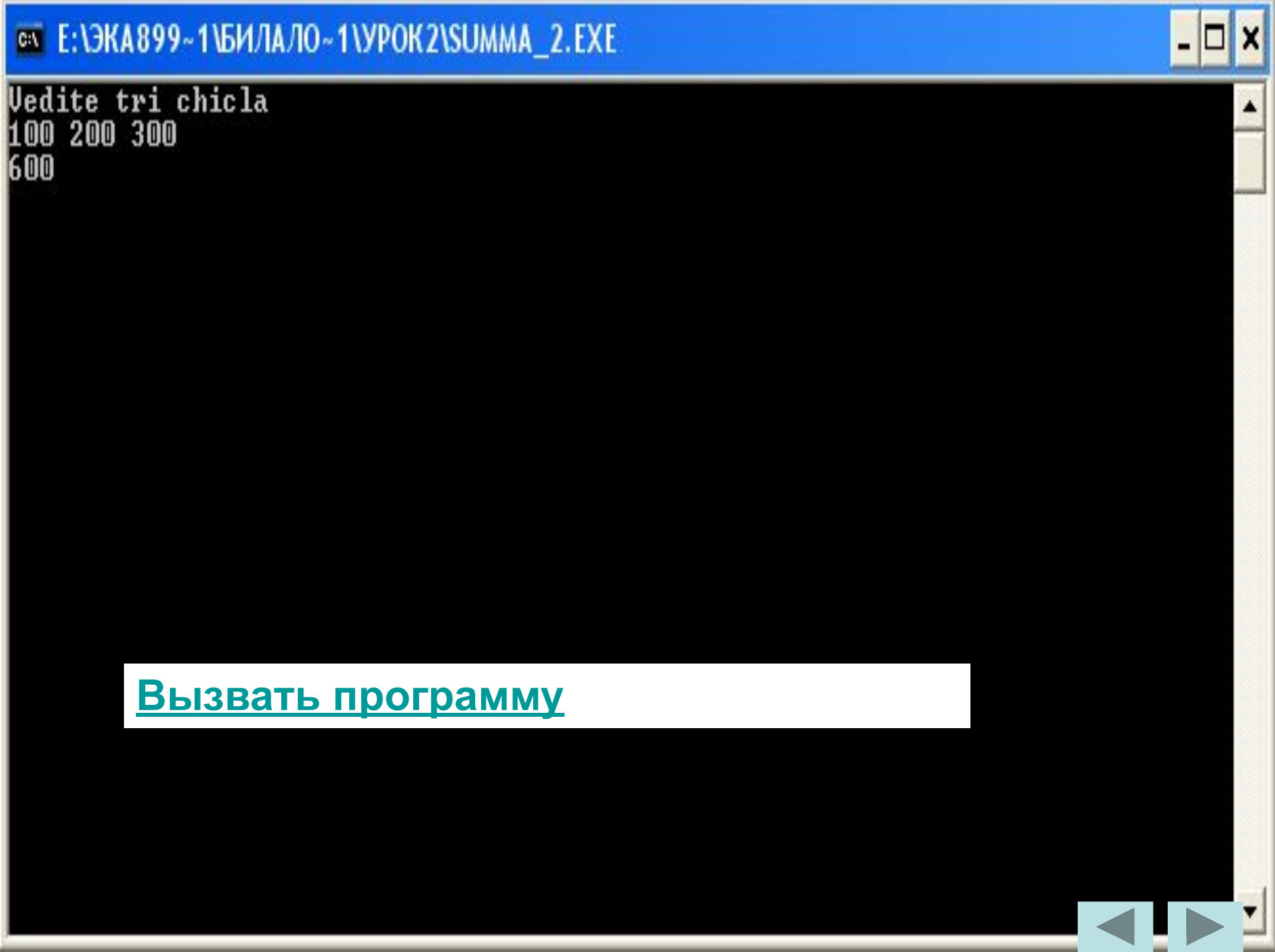
Блок – схема программы



Урок 2.

```
Program SUMMA;
Var   X,Y,K,Z: integer;
Begin
        Writeln('Введите три числа ');
Readln(X,Y,K);
Z:=X+Y+K;
Writeln(Z);
readln
End.
```


Borland Pascal 7.0
File Edit Search Run Compile Debug Tools Options Window Help
SUMMA_2.PAS
Program summa_2;
Var x,y,k,z:integer;
Begin
Write('Vedite tri chicla');
Readln(x,y,k);
z:=x+y+k;
Write(z);
Readln
End.
1:1
F1 Help F2 Save F3 Open Alt+F9 Compile F9 Make Alt+F10 Local menu


E:\ЭКА899~1\БИЛАЛО~1\УРОК2\SUMMA_2.EXE
Vedite tri chicla
100 200 300
600
Вызвать программу

На этом примере видны основные особенности записи программы:

1)всякая программа начинается с заголовка – слова PROGRAM и следующего за ним названия программы; (в последних версиях Паскаля это предложение писать не обязательно)

2)после заголовка располагается описательная часть программы;

3)между служебными словами BEGIN и END записывается алгоритм решения задачи;

4)команды (операторы) разделяются точкой с запятой ;

5)Русский текст заключается в апострофы ‘ … ‘ ;

6)перечисляемые объекты разделяются запятой.

Имена и зарезервированные слова.

Текст программы записывается при помощи латинских букв, цифр и знаков. Буквы допускаются прописные и строчные, причем для компилятора записи VAR, vAR, VaR идентичны.

Особую роль в тексте программы играют имена и зарезервированные слова. Имена применяются для обозначения программы и ее объектов. Имя может состоять из любого количества букв или цифр, но должно начинаться с буквы. В имя можно включать знак подчеркивания (например Prim_1). В программе SUMMA пять имен: SUMMA, X, Y, K, Z. Программисты часто используют осмысленные имена в своих программах. Это всегда полезно, а в сложных программах совершенно необходимо.

Имена и зарезервированные слова.

Зарезервированные слова применяют для обозначения операторов (команд) и других элементов языка Паскаль. Их нельзя использовать в качестве имен и во всех программах они имеют одинаковый смысл. Зарезервированными словами в нашем примере являются слова: **PROGRAM(**программа), **VAR(**переменные), **BEGIN**(начало), **READ**(читать), **WRITELN**(писать), **END**(конец).

Константы и переменные.

Данные, которыми оперирует программа, могут быть определены в ней как неизменные, либо как способные изменять свое значение в ходе выполнения программы. Первые называются константами, а вторые переменными. И переменные и константы размещаются в памяти компьютера (в так называемой «ячейке памяти»). В программе переменные должны быть описаны в предложении VAR (от слова VARIABLE-переменная), а константы – в предложении CONST. У всякой величины имеется три основных свойства: имя, значение, тип.

Тип INTEGER (целый).

И в жизни и в программировании очень полезно использовать понятие типа. Предположим, вам сказали, что вы должны приобрести Джой. Больше ничего не известно – сколько у него ног, предмет это или животное и т.д. в то же время очень многое можно сообщить, добавив лишь одно слово, определяющее тип объекта по имени Джой, например, Джой – собака, или Джой – человек, или Джой – компьютер.

Любая константа или переменная, использованная в программе, принадлежит к определенному типу. Тип задает множество допустимых значений переменных, внешний вид констант, возможные операции над значениями.

Тип INTEGER (целый).

Значения величины типа INTEGER в Паскале не может быть меньше –32768 или больше 32767.

Константы целого типа записываются в виде последовательности цифр со знаком или без него.

Переменные должны быть перечислены в описательной части программы в предложении

VAR имя _переменной:INTEGER;

Над величинами целого типа допустимы арифметический операции: +(сложение), –(вычитание), *(умножение), DIV(деление нацело), MOD(нахождение остатка от целочисленного деления).

Все операции вырабатывают результат целого типа. Например, 15 DIV 4 = 3, 25 MOD 4 = 1.

Над целыми разрешено и обычное деление, оно обозначается косой чертой «/» и дает результат вещественного типа.

Тип REAL(вещественный).

Константы вещественного типа (числа с дробной частью) изображаются с десятичной точкой:
12.3, –1.5, –0.75 или в показательной форме: -0.45E5, 6.7 E-10, 0.355E6 (для получения числа в обычном виде надо перенести запятую на число разрядов указанных после E вправо, если число положительное, влево, если отрицательное). Например, 6.7E-10=0.00000000067.
Вещественные переменные требуют описания предложением
VAR имя:REAL;
Над величинами вещественного типа допустимы арифметические операции: +(сложение), -(вычитание), *(умножение), /(деление).

Оператор присваивания

Оператор присваивания придает переменной конкретное значение, например:

X:=2; Y:=5,

одновременно уничтожая старое. Редкая программа обходится без оператора присваивания.

Присваивать можно значение другой переменной или результат вычисления арифметического выражения: A:=B; A:+B+C; X:=Y+2-Z.

Формат команды:

<имя переменной>:=<выражение>;

Исполнение команды присваивания происходит в таком порядке: сначала вычисляется <выражение>, затем полученное значение присваивается переменной.

Пример 1. Пусть переменная A имела значение 6. какое значение получит переменная A после выполнения команды: A:=2 * A - 1.

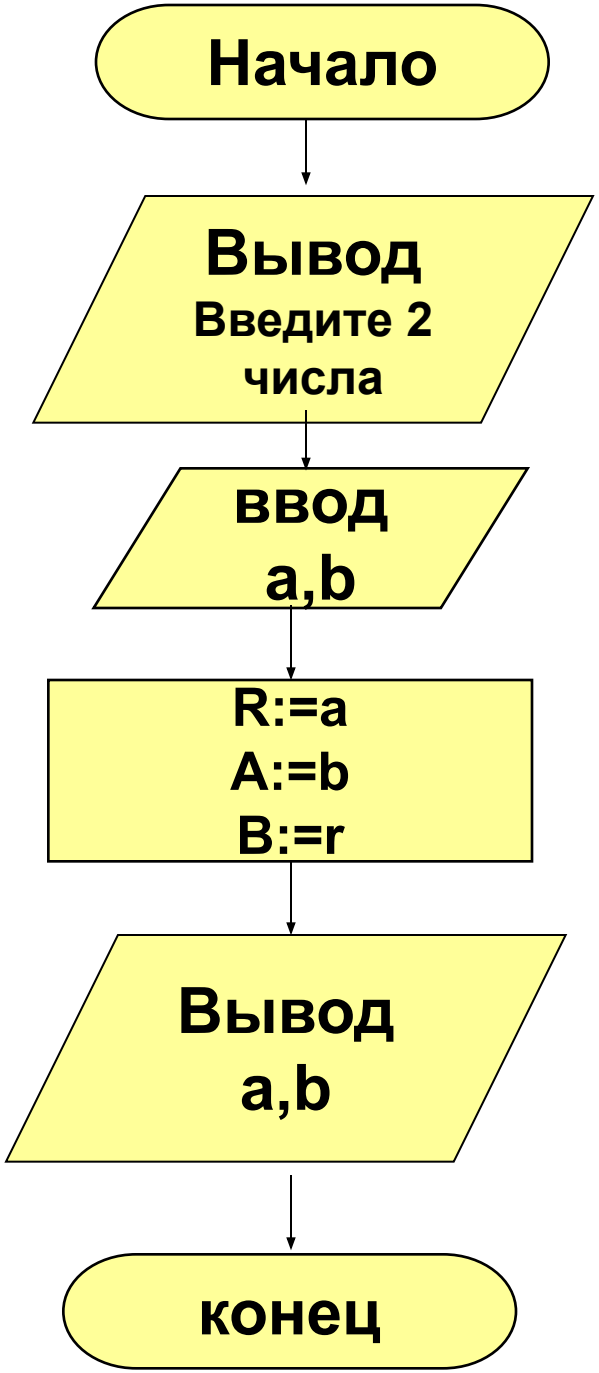
Оператор присваивания

Решение. ***Вычисление выражения 2 * A – 1 при A=6 даст число 11. Значит новое значение переменной A будет равно 11.***

Пример 2.

Поменяйте между собой значения двух переменных A и B, воспользовавшись третьей переменной R для временного хранения значения.

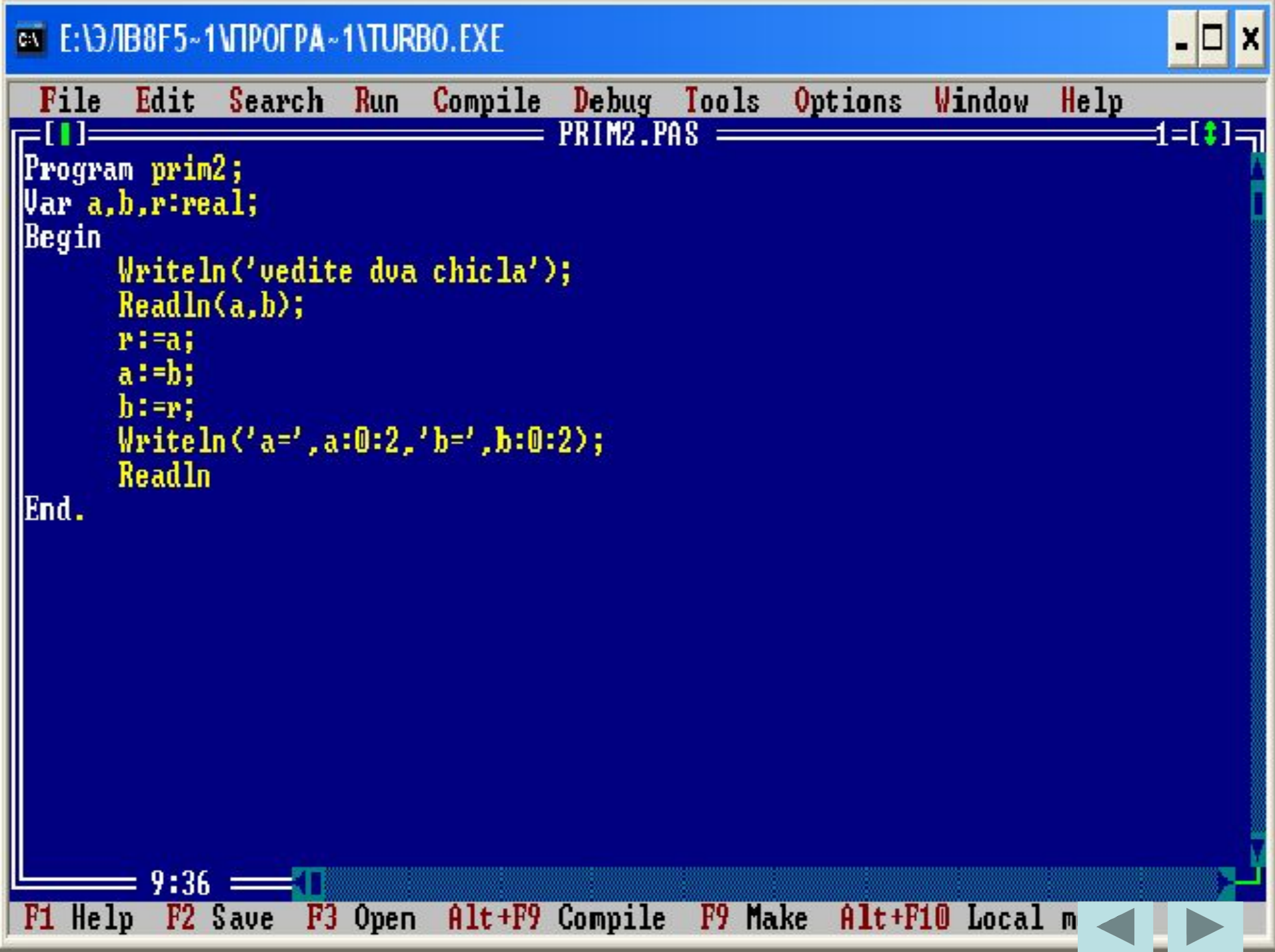
Блок – схема программы

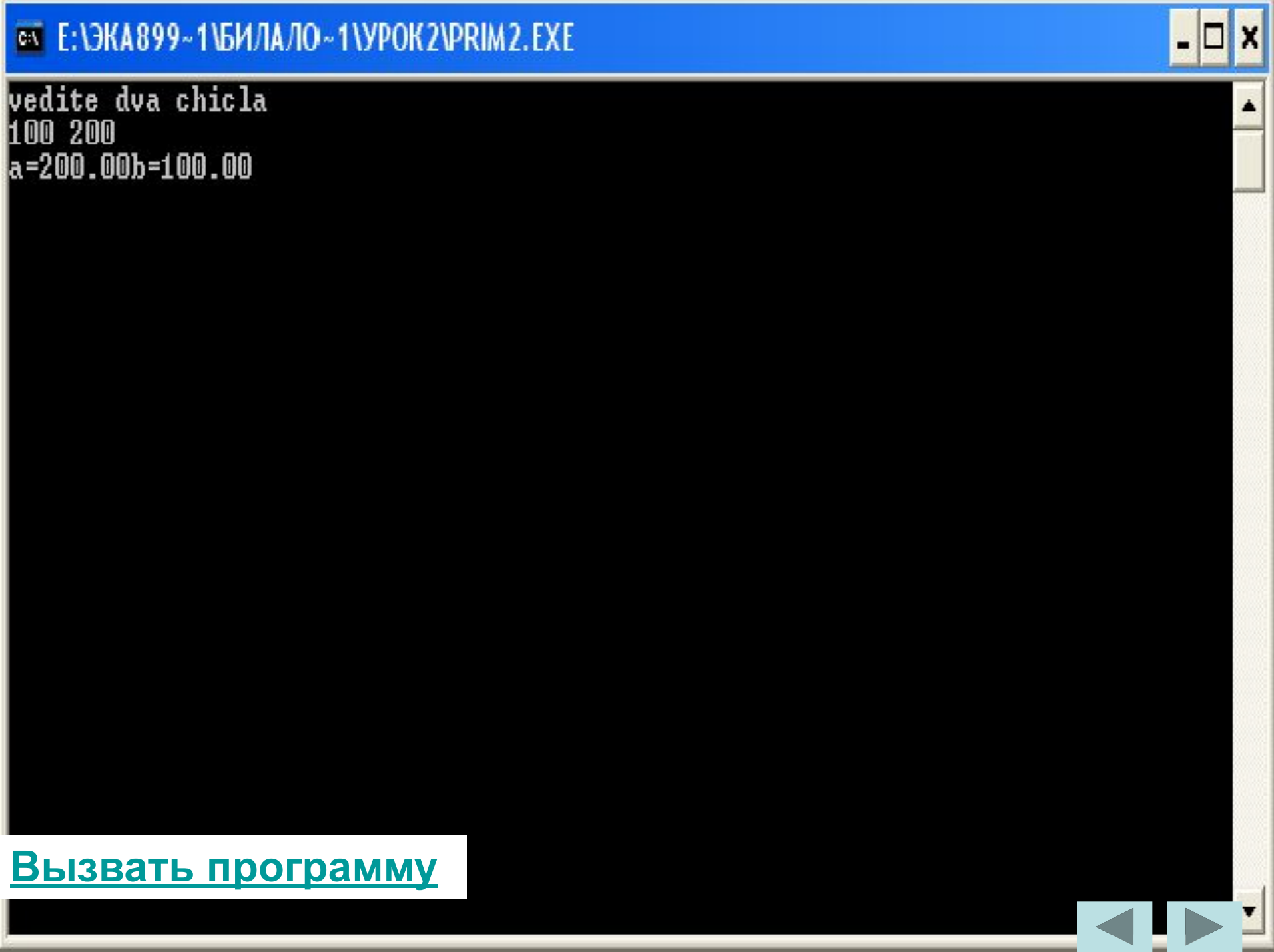


Оператор присваивания

Решение.

```
Program  prim2_2;
Var    a, b, r: real;
begin
    Writeln('Введите два числа ');
    Readln(a,b);
    R:=a;
    A:=b;
    B:=r;
    Writeln(' a=' , a:0:2, '  b=' , b:0:2);
Readln
End.
```


E:\ЭЛВ8F5~1\ПРОГРА~1\TURBO.EXE
File Edit Search Run Compile Debug Tools Options Window Help
PRIM2.PAS
Program prim2;
Var a,b,r:real;
Begin
Writeln('vedite dva chicla');
Readln(a,b);
r:=a;
a:=b;
b:=r;
Writeln('a=',a:0:2,'b=',b:0:2);
Readln
End.
9:36
F1 Help F2 Save F3 Open Alt+F9 Compile F9 Make Alt+F10 Local m


E:\ЭКА899~1\БИЛАЛО~1\УРОК2\PRIM2.EXE
vedite dva chicla
100 200
a=200.00b=100.00
Вызвать программу

Оператор присваивания

Задачи.

1)Определить конечное значение переменных X и Y в результате выполнения следующих алгоритмов:

а)
```
X:=2
X:=X * X
X:=X * X * X
X:=X * X * X * X
```

б)
```
X:=1.5
X:=2 * X + 1
Y:=X/2
Y:=X + Y
X:=X – Y
```

Оператор присваивания

2)Поменяйте между собой значения трех переменных X, Y и Z по схеме тройного квартирного обмена: X→Y→Z→X(см. разбор задачи ниже)

3)Присвойте переменной N ее собственное значение, увеличенное в N раз.

4)Чему равно X в результате выполнения программы

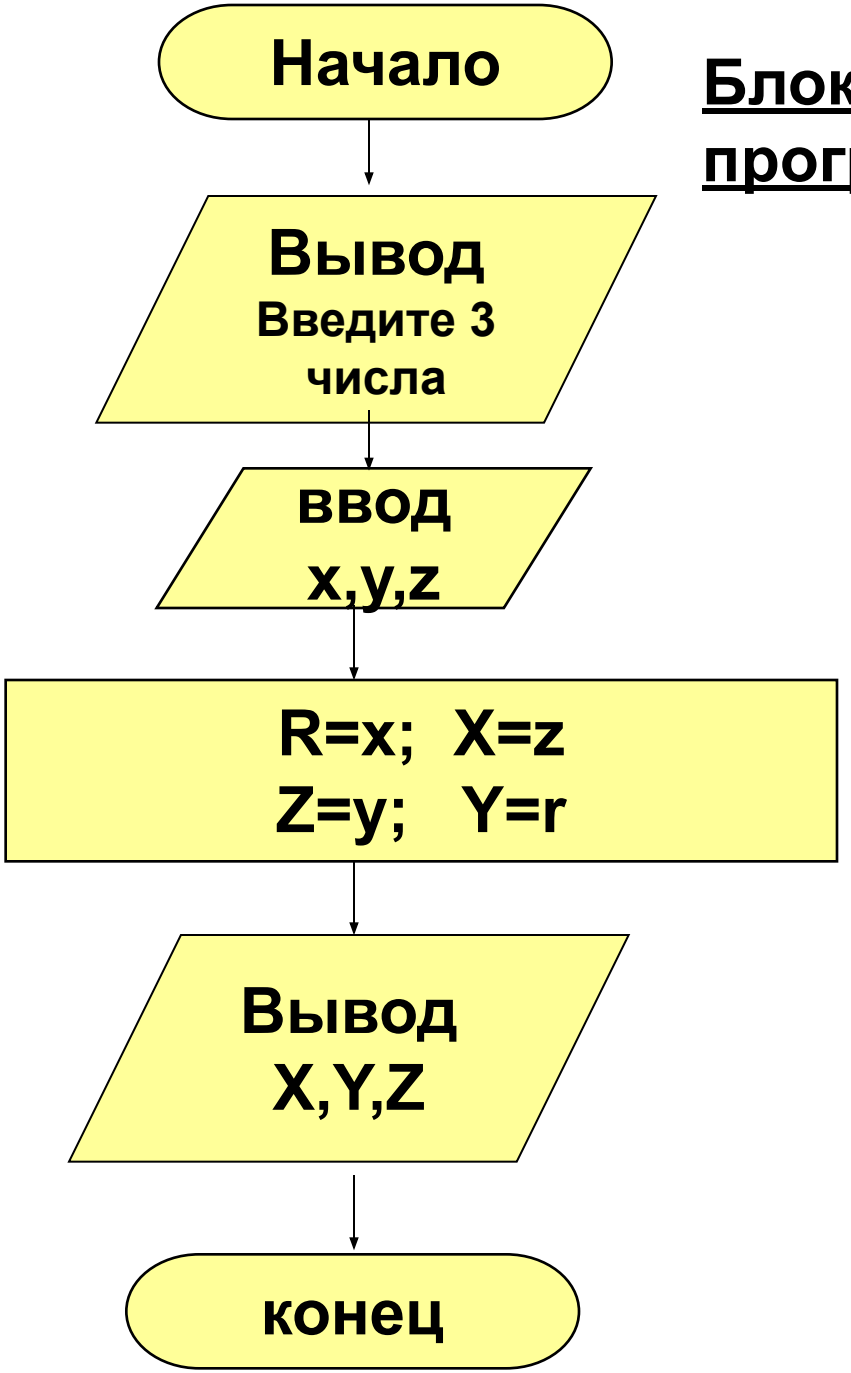
```
X:=2;
X:=X+X;
X:=X-X
```

5)Указать значения величин a и b после выполнения следующих операторов присваивания:

а)
```
a:=5.8
b:=-7.9
b:= a
a:=b
```

б)
```
a:=0
b:=-9.99
b:=a
a:=b
```

Блок – схема программы



Решение

Для начала проверьте правильность выполнения предыдущих заданий. И если не все получилось, не огорчайтесь, не ошибается тот, кто ничего не делает.

1) а) X=16777216 б) X=-2 Y=6

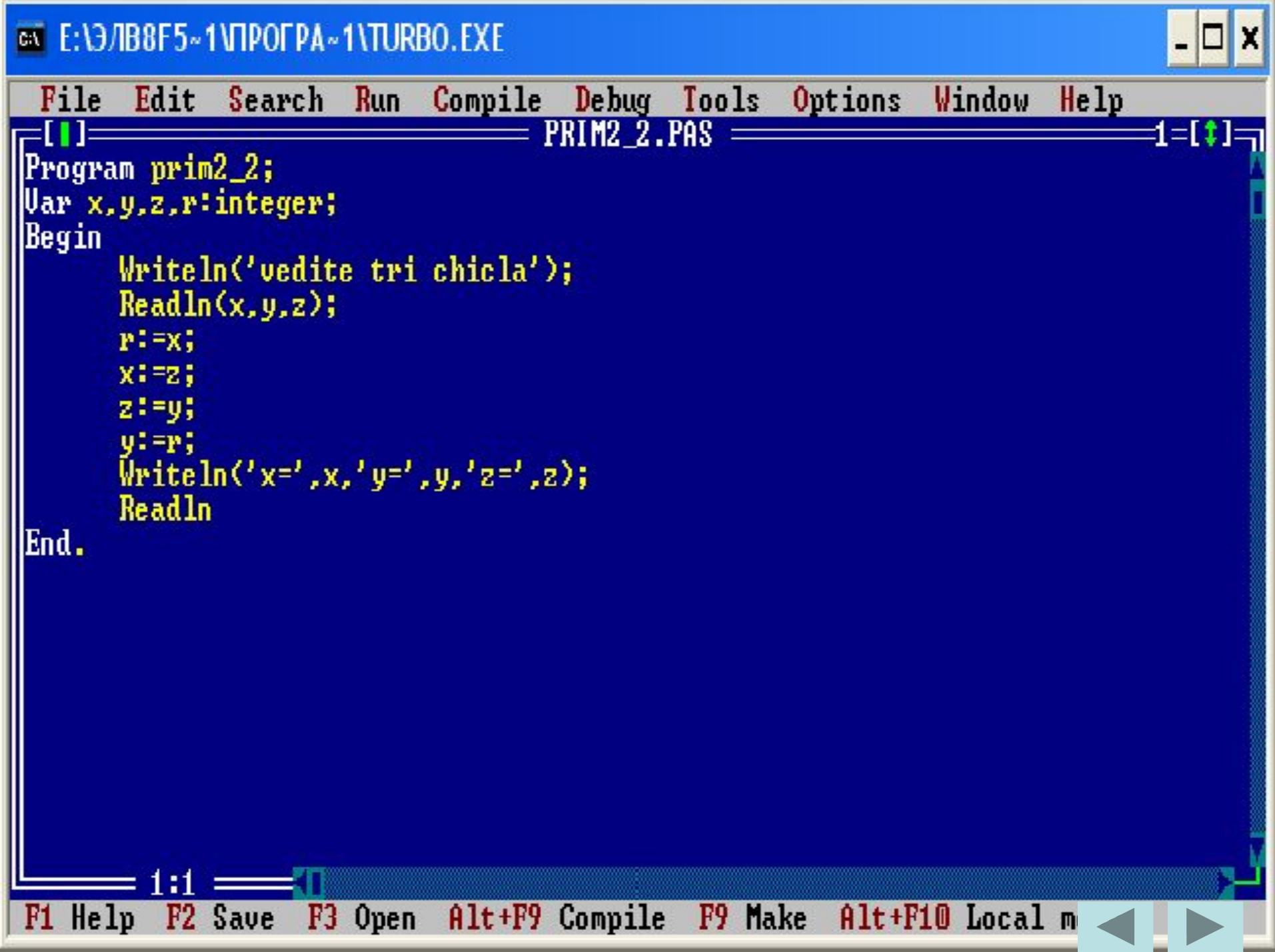
```
Program  prim2_2;
Var    X,Y,Z,R: real;
Begin
     Writeln('Введите три числа ');
     Readln(X,Y,Z);
     R:=X;
     X:=Z;
     Z:=Y;
     Y:=R;
     Writeln(' X=' , X, '  Y=' , Y, ' Z=', Z);
End.
```

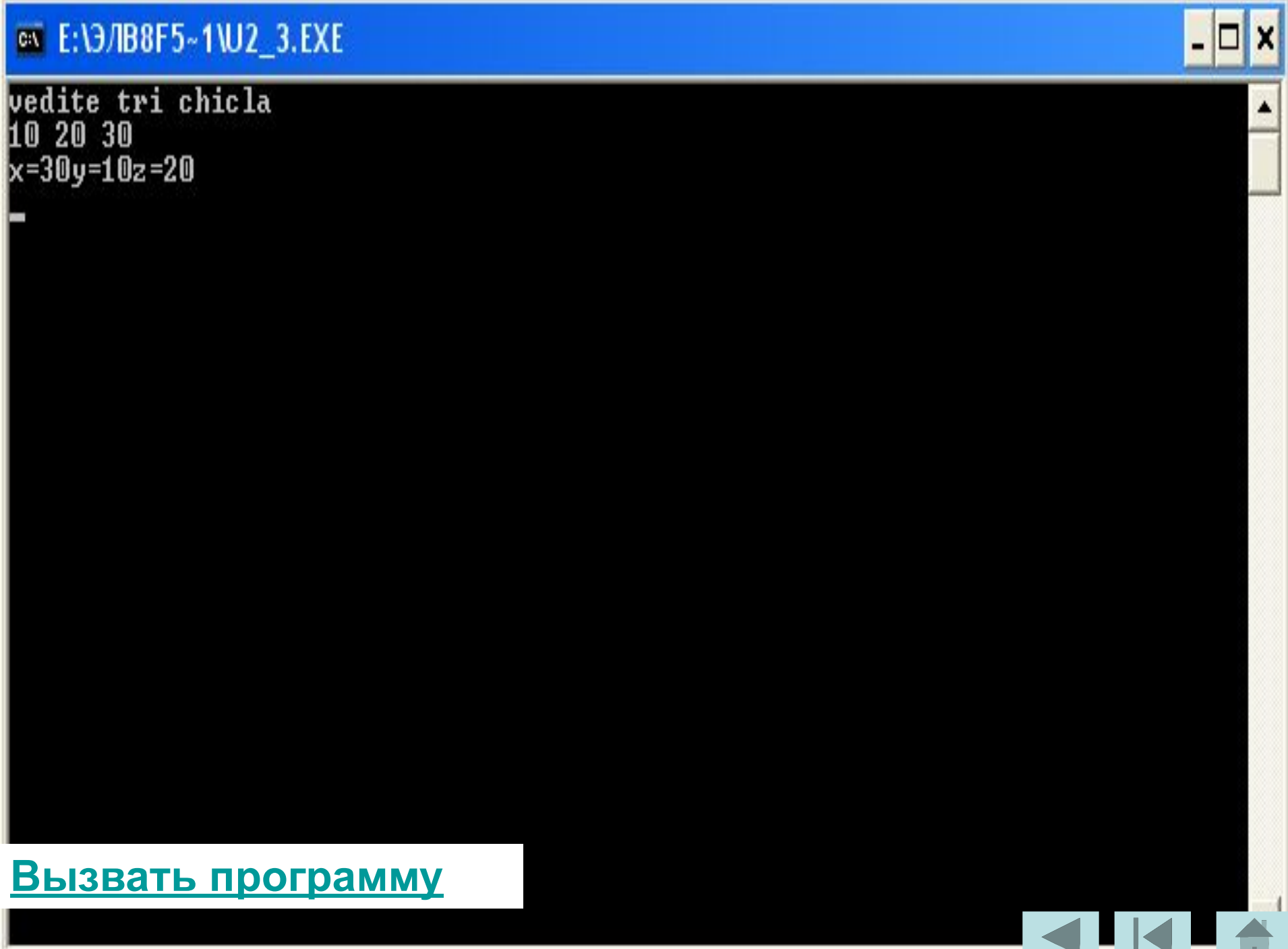
3) N:=N*N

4) X=0

5)а) b=5.8; a=5.8

б) b=0; a=0


E:\ЭЛВ8F5~1\ПРОГРА~1\TURBO.EXE
File Edit Search Run Compile Debug Tools Options Window Help
PRIM2_2.PAS
Program prim2_2;
Var x,y,z,r:integer;
Begin
Writeln('vedite tri chicla');
Readln(x,y,z);
r:=x;
x:=z;
z:=y;
y:=r;
Writeln('x=',x,'y=',y,'z=',z);
Readln
End.
1:1
F1 Help F2 Save F3 Open Alt+F9 Compile F9 Make Alt+F10 Local



Вызвать программу