

# Команды ВВОД-ВЫВОД В Турбо Паскале

- Создайте новый документ.
- Запишите код программы:

```
PROGRAM pr2;  
  
BEGIN  
  
  Write('a= '); ReadLn(a);  
  
  Write('b= '); ReadLn(b);  
  
END.
```

Прежде чем запустить программу, как всегда разберёмся, что мы написали:

- команда `Write('a= ');` выводит на экран сообщение `a=` — надо же как-то сообщить пользователю, что программа от него требует;
- команда `ReadLn(a);` принимает данные, введённые с клавиатуры, и помещает эти данные в переменную `a`;
- во второй строчке основного блока выводится запрос `b=` и введённое пользователем число помещается в переменную `b`.

```
IUANOU\PR2.PAS
Error 3: Unknown identifier.
BEGIN
  Write('a= '); ReadLn(a);
  Write('b= '); ReadLn(b);
END.
3:24
F1 Help  F2 Save  F3 Open  Alt+F9 Compile  F9 Make  Alt+F10 Local menu
```

Компилятор не стал выполнять программу — он обнаружил ошибку, о чём ярко и весело сообщил в верхней строке: **Error 3: Unknow identifier.** — неопределённый идентификатор.

Итак, компилятор не захотел выполнять нашу программу из-за того, что переменные, которые мы используем в программе, не объявлены. Что такое переменная и как её объявить?

Переменная — это некоторая ячейка в памяти, которая имеет имя ([идентификатор](#)). Для того, чтобы Pascal мог правильно работать с переменной (в частности, знал, сколько памяти надо зарезервировать для неё), ему надо сообщить её тип. Типов данных в Pascal достаточно много и постепенно мы познакомимся со всеми (или почти со всеми). Сегодня же мы поработаем с числовыми данными.

- Исправьте программу, добавив в неё объявление переменных **a** и **b**:

```
PROGRAM pr2;
VAR
  a,b:integer;
BEGIN
  Write('a= '); ReadLn(a);
  Write('b= '); ReadLn(b);
END.
```

Как видите, секция объявления переменных расположена до основного блока и начинается ключевым словом [VAR](#). Сначала записывается имя переменной, затем, после символа `:`, пишется тип переменной:

**ИМЯ : ТИП ;**

Если переменных несколько (как в нашем случае), они перечисляются через запятую, хотя можно создать отдельное объявление для каждой переменной:

**VAR**

```
a:integer;
```

```
b:integer;
```

Раз уж мы ввели числа, давайте попросим их «посчитать».

- Добавьте в программу строку, выводящую значение суммы **a+b**:

```
PROGRAM pr2;  
  
VAR  
    a,b:integer;  
  
BEGIN  
    Write('a= '); ReadLn(a);  
    Write('b= '); ReadLn(b);  
    WriteLn('a+b= ',a+b);  
  
END.
```

- Запустите программу — теперь мы видим сумму введённых чисел:

**Замечание.** Обратите внимание, что последняя команда `WriteLn` имеет два параметра:

- строковая константа `'a+b= '`;
- арифметическое выражение `a+b`,

разделённых символом `,`. И, хотя внешне эти параметры очень похожи, первый из них приводит к выводу на экран текста `a+b=` (параметр взят в апострофы), второй же считает значение выражения и выводит его результат `53`.



Но в программах часто бывает необходимо не вывести значение какого-то выражения, а сохранить его для дальнейшего использования. В таких случаях результат присваивают какой-либо переменной.

- Дополните и измените программу так, чтобы сумма чисел сохранялась в переменной **c** и лишь затем выводилась на экран.

```
PROGRAM pr2;
VAR
  a,b:integer;
  c:integer;
BEGIN
  Write('a= '); ReadLn(a);
  Write('b= '); ReadLn(b);
  c:=a+b;
  WriteLn('a+b= ',c);
END.
```

Результат работы программы от этого не изменился, но мы в случае необходимости могли бы использовать значение суммы в других вычислениях.

**Замечание.** Выражение  $c := a + b$ ; можно прочесть так: переменной  $c$  присваивается значение суммы  $a + b$ .

**Обратите внимание!** Операция присваивания записывается двумя символами  $:=$  (двоеточие и знак равенства).

- А теперь самостоятельно дополните программу так, чтобы кроме суммы выводились ещё разность (операция  $-$ ), произведение (операция  $*$ ) и частное (операция  $/$ ).



```
VAR
    a,b:integer;
    c:integer;
BEGIN
    Write('a= '); ReadLn(a);
    Write('b= '); ReadLn(b);
    c:=a+b;
    WriteLn('a+b= ',c);
    c:=a-b;
    WriteLn('a-b= ',c);
    c:=a*b;
    WriteLn('a*b= ',c);
    c:=a/b;
    WriteLn('a:b= ',c);
END.
```

Ошибка **Type mismatch** (несоответствие типов) появляется в том случае, когда переменной одного типа присваивается значение другого типа.

В нашем примере ошибка возникла тогда, когда переменной **c** мы попытались присвоить частное чисел **a** и **b** (вы ведь помните, что при делении двух чисел может получиться дробное число? Как видите, Pascal это тоже помнит).

- Для исправления этой ошибки нам придётся изменить тип переменной **c** на вещественный (**real**):

```
VAR
  a,b:integer;
  c:real;
BEGIN
  Write('a= '); ReadLn(a);
  Write('b= '); ReadLn(b);

  c:=a+b;
  WriteLn('a+b= ',c);

  c:=a-b;
  WriteLn('a-b= ',c);

  c:=a*b;
  WriteLn('a*b= ',c);

  c:=a/b;
  WriteLn('a:b= ',c);
END.
```

Конечно, результат `1.9000000000000000E+001` вас не очень порадует (хотя это всего лишь вещественное представление числа `19` — обратите внимание, это ведь известная вам из курса алгебры стандартная запись числа  $1,9 \cdot 10^1$ ).

К счастью, в Pascal есть возможность вывести вещественное число в привычном нам виде.

```
VAR
    a,b:integer;
    c:real;
BEGIN
    Write('a= '); ReadLn(a);
    Write('b= '); ReadLn(b);

    c:=a+b;
    WriteLn('a+b= ',c:6:2);

    c:=a-b;
    WriteLn('a-b= ',c:6:2);

    c:=a*b;
    WriteLn('a*b= ',c:6:2);

    c:=a/b;
    WriteLn('a:b= ',c:6:2);
END.
```

Запись `c:6:2` означает, что при выводе числа на экран для него будет отведено шесть позиций, две из которых отведены под дробную часть (соответственно, ещё одна позиция под точку, а на целую часть вместе со знаком останется три позиции).

**Обратите внимание!** В Pascal в вещественных числах целая часть от дробной отделяется точкой, а не запятой.

Вы, конечно, уже заметили, что в то время, когда по команде `ReadLn` среда ожидает от нас ввода данных, она не выполняет никаких действий. Работа программы будет продолжена только после того, как мы введём требуемые данные и нажмём клавишу `Enter`. Это свойство используется для того, чтобы задержать закрытие пользовательского экрана до тех пор, пока мы не рассмотрим результат выполнения программы.

На скриншоте изображена ситуация, когда компилятор среды Borland Pascal обнаружил ошибку.

1.

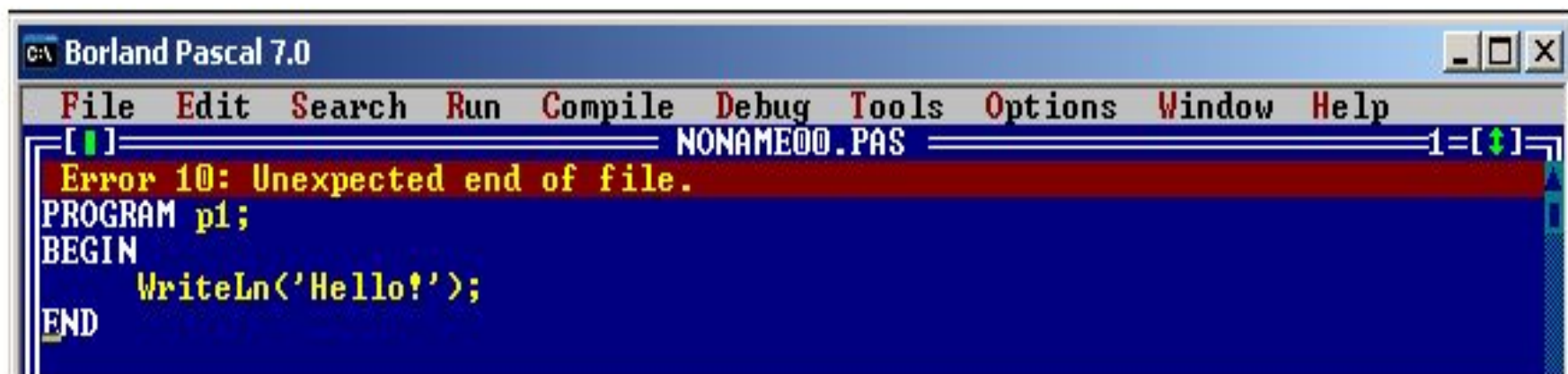


The screenshot shows the Borland Pascal 7.0 IDE window. The menu bar includes File, Edit, Search, Run, Compile, Debug, Tools, Options, Window, and Help. The title bar reads 'C:\ Borland Pascal 7.0'. The editor window is titled 'NONAME00.PAS' and contains the following Pascal code:

```
PROGRAM p=1;  
BEGIN  
    WriteLn('Hello!');  
END.
```

A red error message banner at the top of the editor reads: **Error 85: ";" expected.** The cursor is positioned at the end of the first line.

2.



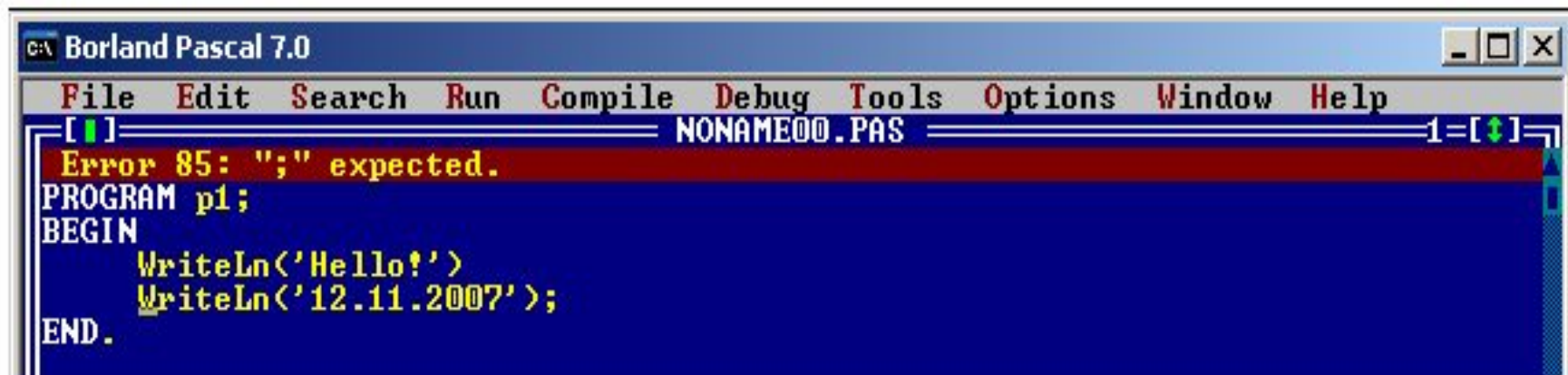
The screenshot shows the Borland Pascal 7.0 IDE window. The menu bar includes File, Edit, Search, Run, Compile, Debug, Tools, Options, Window, and Help. The title bar reads 'C:\ Borland Pascal 7.0'. The editor window is titled 'NONAME00.PAS' and contains the following Pascal code:

```
PROGRAM p1;  
BEGIN  
    WriteLn('Hello!');  
END
```

A red error message banner at the top of the editor reads: **Error 10: Unexpected end of file.** The cursor is positioned at the end of the last line.



3.




The screenshot shows the Borland Pascal 7.0 IDE window titled "c:\ Borland Pascal 7.0". The menu bar includes "File", "Edit", "Search", "Run", "Compile", "Debug", "Tools", "Options", "Window", and "Help". The editor window is titled "NONAME00.PAS" and contains the following Pascal code:

```
PROGRAM p1;  
BEGIN  
    WriteLn('Hello!')  
    WriteLn('12.11.2007');  
END.
```

A red error message bar at the top of the editor displays "Error 85: ";" expected." in yellow text. The error points to the second line of code, "WriteLn('Hello!')", where a semicolon is missing at the end of the statement.

4.



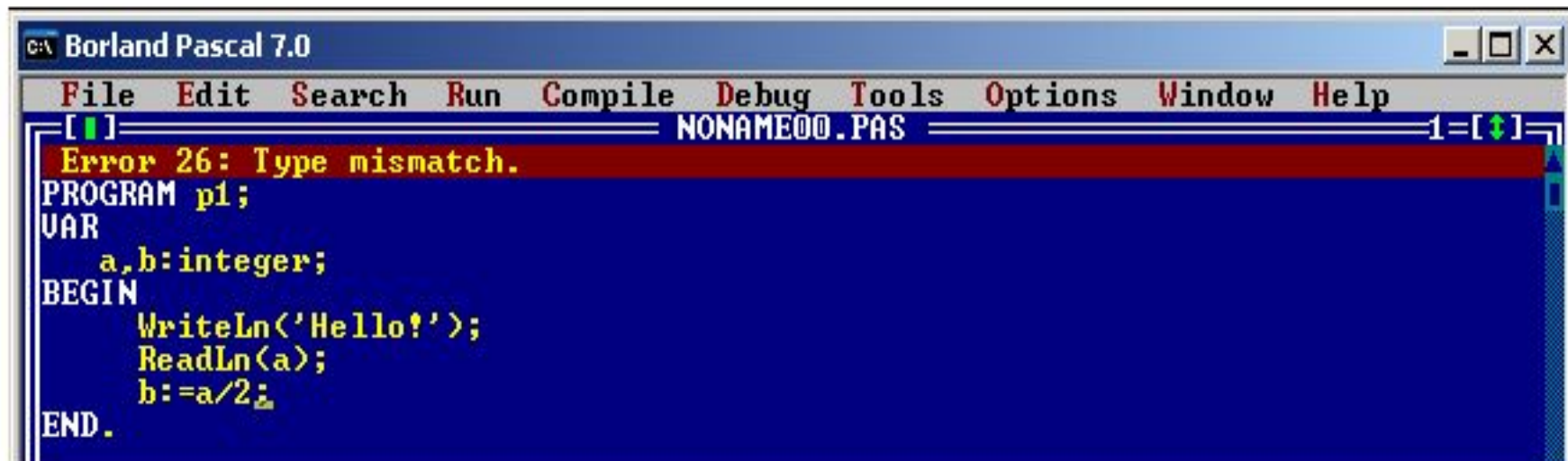
The screenshot shows the Borland Pascal 7.0 IDE window titled "c:\ Borland Pascal 7.0". The menu bar includes "File", "Edit", "Search", "Run", "Compile", "Debug", "Tools", "Options", "Window", and "Help". The editor window is titled "NONAME00.PAS" and contains the following Pascal code:

```
PROGRAM p1;  
BEGIN  
    WriteLn('Hello!');  
    ReadLn(a);  
END.
```

A red error message bar at the top of the editor displays "Error 3: Unknown identifier." in yellow text. The error points to the second line of code, "ReadLn(a);", where the variable "a" has not been declared.



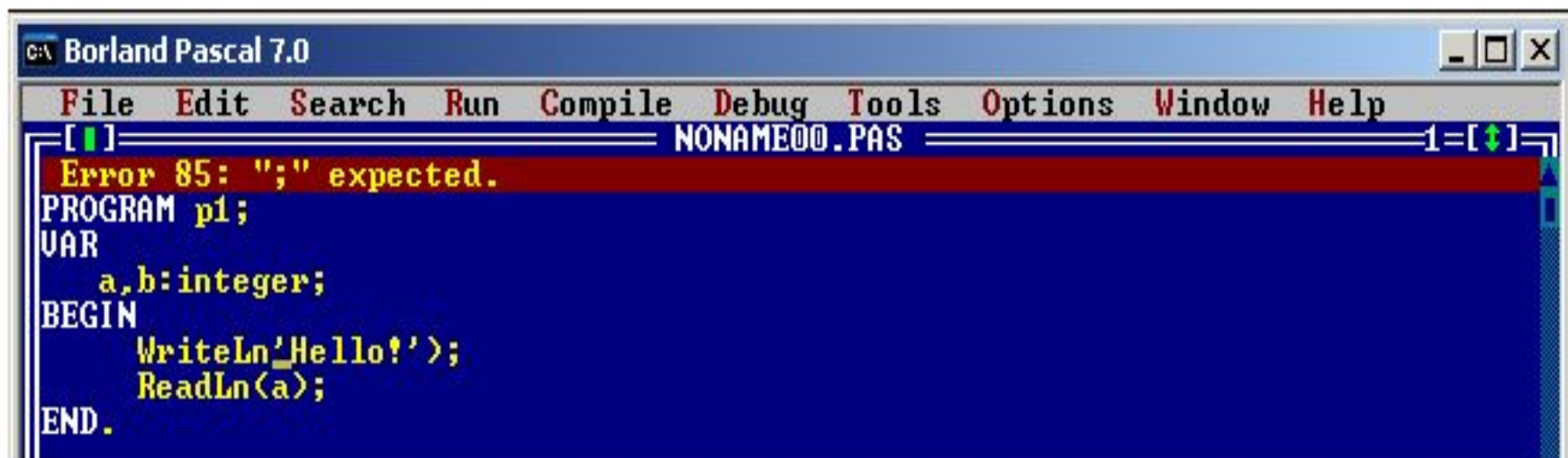
5.



The screenshot shows the Borland Pascal 7.0 IDE with a menu bar (File, Edit, Search, Run, Compile, Debug, Tools, Options, Window, Help) and a toolbar. The main window displays the source code for a program named NONAME00.PAS. A red error banner at the top reads "Error 26: Type mismatch." The code is as follows:

```
PROGRAM p1;  
VAR  
    a,b:integer;  
BEGIN  
    WriteLn('Hello!');  
    ReadLn(a);  
    b:=a/2;  
END.
```

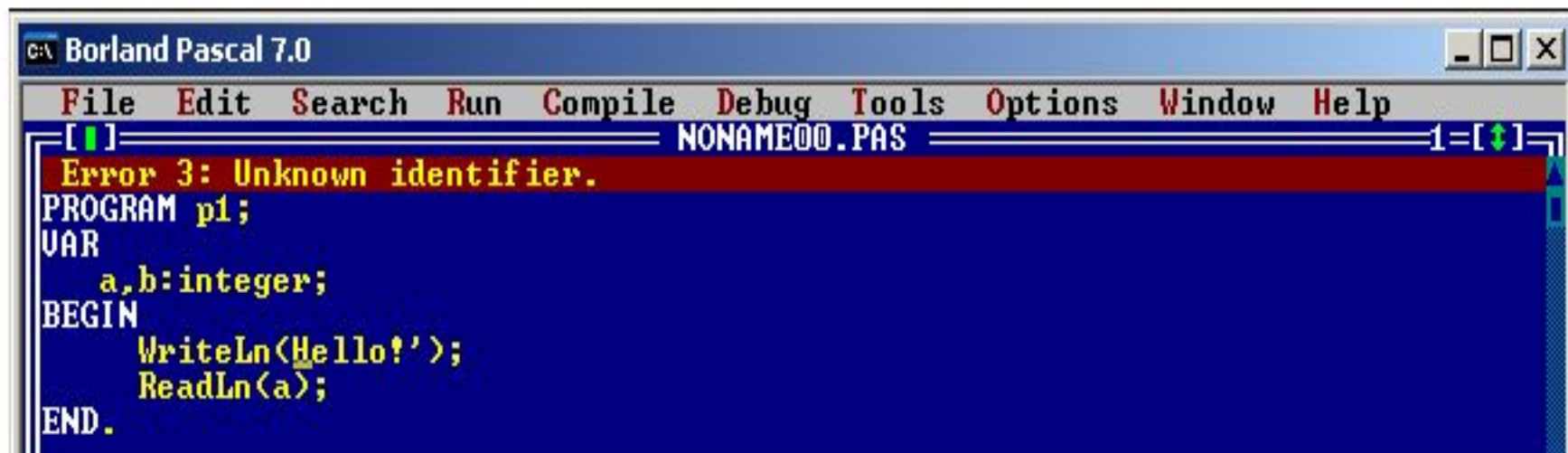
6.



The screenshot shows the Borland Pascal 7.0 IDE with a menu bar (File, Edit, Search, Run, Compile, Debug, Tools, Options, Window, Help) and a toolbar. The main window displays the source code for a program named NONAME00.PAS. A red error banner at the top reads "Error 85: ';' expected." The code is as follows:

```
PROGRAM p1;  
VAR  
    a,b:integer;  
BEGIN  
    WriteLn'Hello!');  
    ReadLn(a);  
END.
```

7.



The image shows a screenshot of the Borland Pascal 7.0 IDE. The window title is "c:\ Borland Pascal 7.0". The menu bar includes "File", "Edit", "Search", "Run", "Compile", "Debug", "Tools", "Options", "Window", and "Help". The current file is "NONAME00.PAS". A red error message is displayed: "Error 3: Unknown identifier." The code in the editor is as follows:

```
PROGRAM p1;  
VAR  
    a,b:integer;  
BEGIN  
    WriteLn('Hello!');  
    ReadLn(a);  
END.
```

## Задание 1

Создайте программу, которая запрашивает три целых числа и выводит на экран сумму первых двух умноженную на третье число.

**Протокол работы программы:**

```
a= 12
```

```
b= 17
```

```
c= 9
```

```
(a+b)*c= 261
```

Жирным выделен текст, который необходимо ввести пользователю.

## Задание 2

Создайте программу, которая запрашивает два целых числа и выводит на экран удвоенное первое число и половину второго числа.

**Протокол работы программы:**

```
a, b= 12 17  
a*2= 24 b:2= 8.5
```

Жирным выделен текст, который необходимо ввести пользователю.

**Указание.** Для запроса двух (и более) чисел в одной команде `ReadLn` необходимо перечислить их через запятую (`ReadLn (a, b)`). При вводе этих чисел во время выполнения программы они разделяются пробелом.

### Задание 3

Создайте программу, которая запрашивает два вещественных числа и выводит на экран их сумму, разность, произведение и частное.

**Протокол работы программы:**

```
a= 1.2  
b= -1.7  
a+b= -0.50   a-b=  2.90  
a*b= -2.04   a:b= -0.71
```

Жирным выделен текст, который необходимо ввести пользователю.

## Задание 4

Создайте программу, которая запрашивает значение вещественного числа  $x$  и выводит на экран сумму и разность  $x$  и  $100$ .

**Протокол работы программы:**

```
x= 15.2  
15.2+100= 115.2  
15.2-100= -84.8
```

Жирным выделен текст, который необходимо ввести пользователю.