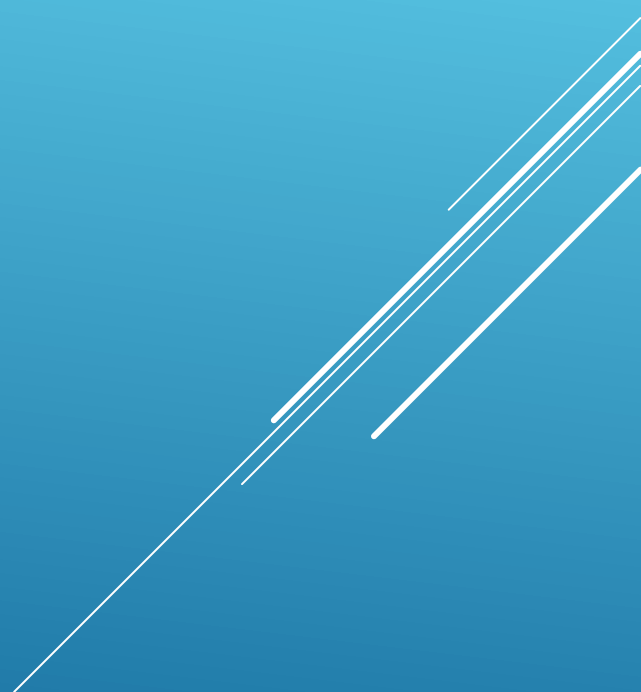
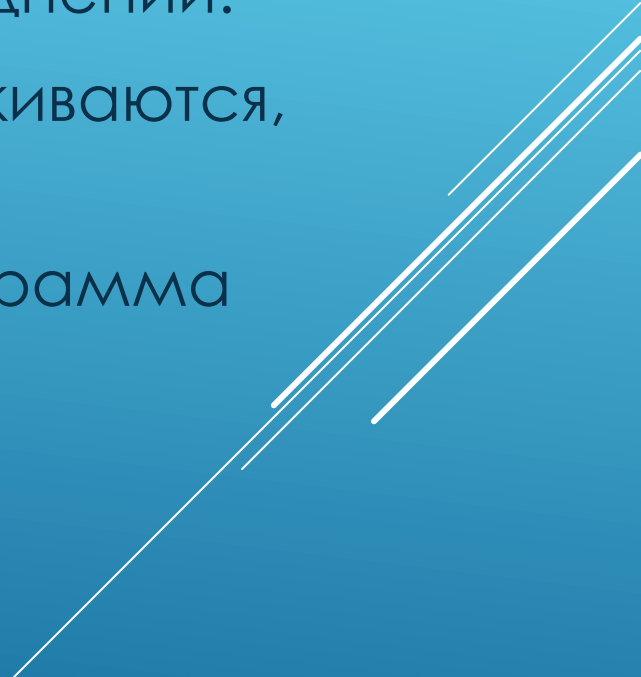


# ИСПОЛЬЗОВАНИЕ ОТЛАДОЧНЫХ СРЕДСТВ В СРЕДАХ ПРОГРАММИРОВАНИЯ


# ВВЕДЕНИЕ




# ОТЛАДКА

- ▶ Как известно ошибки могут быть синтаксическими и семантическими.
  - ▶ Синтаксические ошибки обнаруживаются компилятором, и потому их исправление обычно не вызывает затруднений.
  - ▶ Семантические ошибки компилятором не обнаруживаются, и потому найти и исправить гораздо сложнее.
  - ▶ Но, естественно, это должно быть сделано, т.к. программа должна работать корректно и надёжно.
- 


# ОТЛАДКА

- ▶ Надо сказать, что отладка – один из самых длительных и трудоёмких этапов жизненного цикла программного обеспечения.
  - ▶ Использование отладочных средств облегчает и ускоряет этот процесс.
- 


# ОТЛАДКА

- ▶ Для того, чтобы в программе найти ошибку, её надо прежде всего локализовать, т.к. выделить небольшой участок кода программы, где содержится ошибка.
  - ▶ Для этого необходимо проследить, что происходит при выполнении программы, как меняются значения переменных по ходу выполнения программы.
- 

# ОТЛАДКА

- ▶ До разработки функциональных сред программирования, которые предоставляют набор отладочных средств, приходилось вставлять в программу много операторов вывода.
  - ▶ Теперь мы можем выполнить программу по шагам и посмотреть, каковы значения переменных на каждом шаге программы.
- 

# ОТЛАДКА


- ▶ Мы рассмотрим использование отладочных средств на примере редактора PascalABC.
  - ▶ Но надо сказать, что набор этих средств универсальный, и возможности, предоставляемые разными средами программирования, не отличаются друг от друга.
- 

# СРЕДСТВА ОТЛАДКИ





# СРЕДСТВА ОТЛАДКИ

- ▶ Отладочные средства включают в себя:
    - возможности для пошагового выполнения программы,
    - средства наблюдения значений переменных.
- 

# ПОШАГОВОЕ ВЫПОЛНЕНИЕ ПРОГРАММЫ

- ▶ Если в PascalABC мы откроем пункт меню «Программа», мы увидим не только команды для запуска программы, но также пункты
  - «Шаг без входа в подпрограмму»,
  - «Шаг с входом в подпрограмму»,
  - «Шаг с выходом из подпрограммы»,
  - «Идти к курсору».

# ПОШАГОВОЕ ВЫПОЛНЕНИЕ ПРОГРАММЫ

- ▶ С помощью этих пунктов меню можно управлять процессом выполнения программы.
- ▶ Каждому пункту соответствует своя функциональная клавиша, так что при частом использовании этих средств можно будет запомнить и использовать назначенные функциональные клавиши.









AB PascalABC.NET

Файл Правка Вид Программа Сервис Модули Помощь

•Program.pas

{ } Глобальные имена

```
{ Даны три одно  
Какой из мас  
program Lab14;  
  
const  
  nmax = 20;  
  
type  
  mas = array [
```

	Выполнить	F9
	Выполнить без связи с оболочкой	Shift+F9
	Параметры командной строки...	
	Завершить	Ctrl+F2
	Компилировать	Ctrl+F9
	Перекомпилировать все	Ctrl+Shift+F9
	Шаг без входа в подпрограмму	F8
	Шаг с входом в подпрограмму	F7
	Шаг с выходом из подпрограммы	F10
	Идти к курсору	F4

# ТОЧКИ ОСТАНОВА

- ▶ Кроме того, для управления процессом выполнения программы можно использовать так называемые *точки останова*.
- ▶ Для того, чтобы установить точку останова на какой-либо оператор, обычно надо нажать левой клавишей мыши на полосу слева от текста программы.
- ▶ После установки точки останова выполнение программы всегда будет останавливаться на выделенном операторе, независимо от использованной команды.

PascalABC.NET

Файл Правка Вид Программа Сервис Модули Помощь

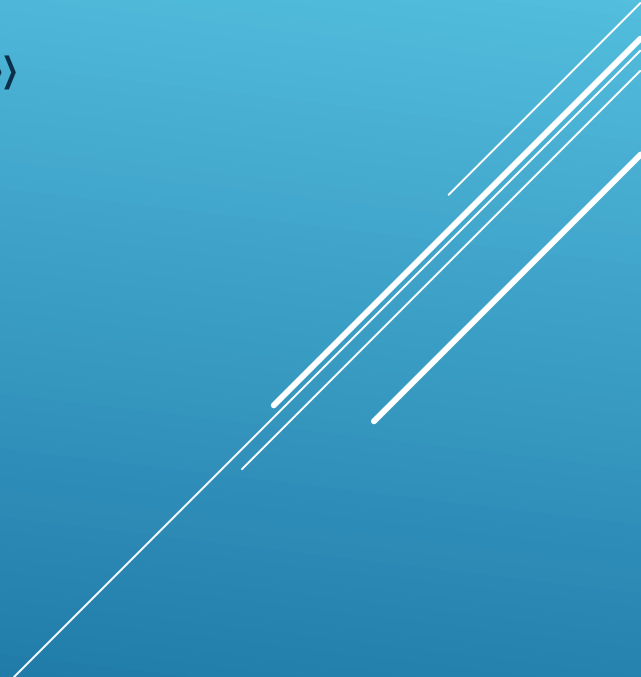
Program.pas

{ } Глобальные имена

```
AssignFile(f, ParamStr(1));
Reset(f);
Get(a, na, f);
Get(b, nb, f);
Get(c, nc, f);
readln(f, pa, pb, pc);
CloseFile(f);

{ Определение в каждом из массивов среднего арифметического элементов,
  больших заданного числа }
sra := Average(a, na, pa);
srb := Average(b, nb, pb);
src := Average(c, nc, pc);
```

# ПРОСМОТР ЗНАЧЕНИЙ ПЕРЕМЕННЫХ

- ▶ При пошаговом выполнении программы надо наблюдать изменение значений переменных.
  - ▶ Для этого в нижнем окне редактора PascalABC при пошаговом выполнении появляются две дополнительные вкладки – «Локальные переменные» и «Просмотр выражений».
- 

```
{ Определение в каждом из массивов среднего арифметического элементов,  
  больших заданного числа }  
sra := Average(a, na, pa);  
srb := Average(b, nb, pb);  
src := Average(c, nc, pc);  
  
{ Вывод полученных результатов }  
AssignFile(f, ParamStr(2));  
Rewrite(f);  
if (sra = srb) and (srb = src) then  
  writeln(f, 'Среднее арифметическое совпадает во всех массивах и равно ', sra:2:2)  
else if (sra = srb) and (sra < src) then  
  writeln(f, 'Массивы А и В имеют одинаковое минимальное среднее арифметическое, равное ', sra:2:2)  
else if (sra = src) and (sra < srb) then  
  writeln(f, 'Массивы А и С имеют одинаковое минимальное среднее арифметическое, равное ', sra:2:2)  
else if (srb = src) and (srb < sra) then  
  writeln(f, 'Массивы В и С имеют одинаковое минимальное среднее арифметическое, равное ', srb:2:2)  
else if (sra < srb) and (sra < src) then  
  writeln(f, 'Массив А имеет минимальное среднее арифметическое, равное ', sra:2:2)  
else if (srb < sra) and (srb < src) then
```

Окно вывода

Окно вывода

Список ошибок

Сообщения компилятора


Локальные переменные

Просмотр выражений

Компиляция прошла успешно (97 строк), 3 предупреждений



# ВКЛАДКА «ЛОКАЛЬНЫЕ ПЕРЕМЕННЫЕ»

- ▶ На вкладке «Локальные переменные» показываются значения глобальных и локальных переменных.
  - ▶ Глобальные переменные изначально скрыты, чтобы увидеть их надо нажать +.
  - ▶ Ниже отображаются значения локальных переменных, которые относятся к функции, которая выполняется.
- 

```
{ Функция поиска среднего арифметического элементов массива, больших v
function Average(const x: mas; n: integer; v: real): real;
var
  k, i: integer;
begin
  Result := 0;
  for i := 1 to n do
    if x[i] > v then
      begin
        Result := Result + x[i];
        k := k + 1;
      end;
  if k <> 0 then
    Result := Result / k;
end;
```

#### Локальные переменные

Переменная	Значение
Глобальные переменные	
Result	17
k	2
i	4
x	(8,0,-1,9,19,0,0,0,0,0,...)
n	5
v	2

Окно вывода Список ошибок Сообщения компилятора Локальные переменные Просмотр выражений

Компиляция прошла успешно (97 строк), 3 предупреждений

# ВКЛАДКА «ПРОСМОТР ВЫРАЖЕНИЙ»

- ▶ На вкладке «Просмотр выражений» можно ввести любые выражения, значения которых вы хотите увидеть.
- ▶ На рисунке мы видим несколько глобальных переменных, значение элемента массива  $x$  и значение выражения  $x[i] > v$ .
- ▶ Значения элементов массива  $x$  также отображаются на вкладке «Локальные переменные», но там они перечислены списком, и может быть сложно выделить нужный элемент массива, особенно с большим номером.

```
{ функция поиска среднего арифметического элементов массива, больших v
function Average(const x: mas; n: integer; v: real): real;
var
  k, i: integer;
begin
  Result := 0;
  for i := 1 to n do
    if x[i] > v then
      begin
        Result := Result + x[i];
        k := k + 1;
      end;
  if k <> 0 then
    Result := Result / k;
end;
```

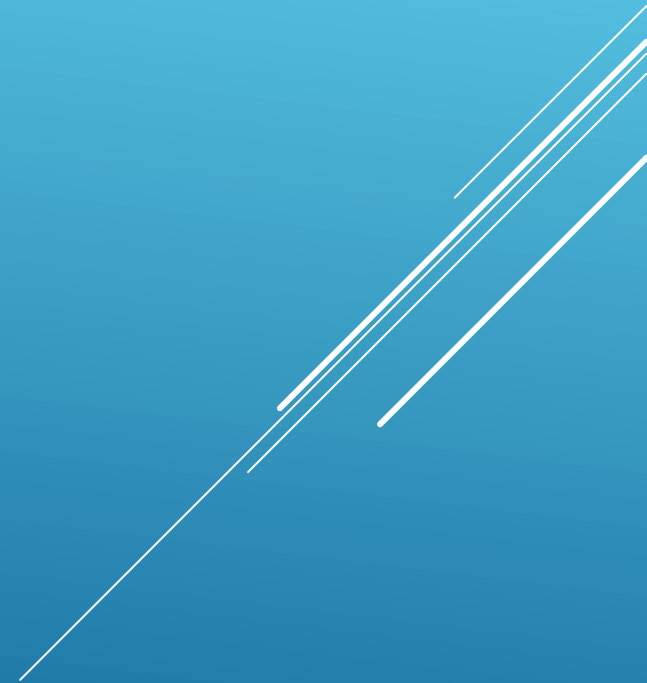
#### Просмотр выражений

Выражение	Значение
♦ sra	0
♦ srb	0
♦ src	0
♦ x[i]	9
♦ x[i]>v	True

Окно вывода | Список ошибок | Сообщения компилятора | Локальные переменные | Просмотр выражений

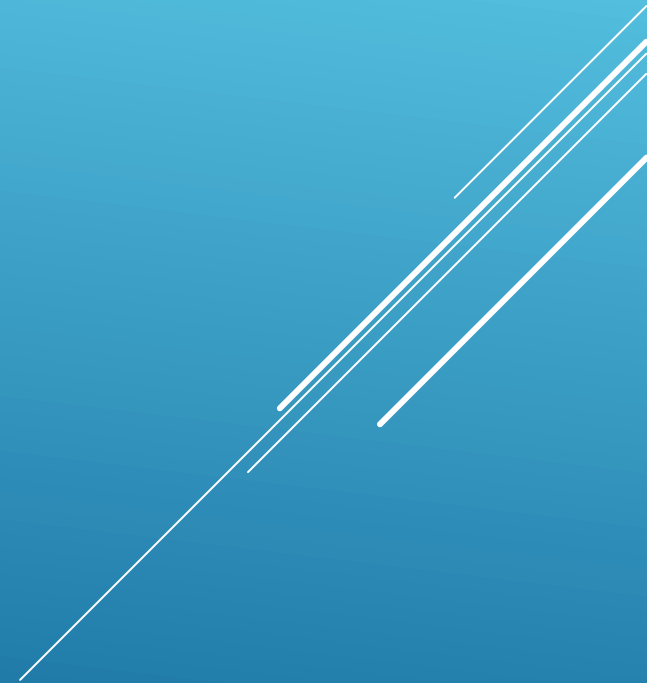
Компиляция прошла успешно (97 строк), 3 предупреждений

# ПРИМЕР ОТЛАДКИ



# ПОСТАНОВКА ЗАДАЧИ

- ▶ Рассмотрим пример отладки программы, которая определяет какой из трёх одномерных массивов имеет наименьшее среднее арифметическое элементов, больших заданного числа.



# ПОСТАНОВКА ЗАДАЧИ

- ▶ В программе мы имеем:
  - объявления переменных;
  - процедуры ввода и вывода;
  - функцию поиска среднего арифметического элементов, больших заданного числа, в одном массиве;
  - основную программу, которая вводит исходные данные, вызывает функцию поиска среднего арифметического для каждого из трёх массива, и сравнивает полученные значения.

```
PascalABC.NET
Файл  Правка  Вид  Программа  Сервис  Модули  Помощь
[Icons]
Program.pas
{ } Глобальные имена
{ Даны три одномерных массива a, b и c разного размера.
  Какой из массивов имеет наименьшее среднее арифметическое элементов,
  больших заданного числа? }
program Lab14;

const
  nmax = 20;

type
  mas = array [1..nmax] of real;

var
  a, b, c: mas;           { Массивы }
  na, nb, nc: integer;   { Количество элементов в массивах }
  pa, pb, pc: real;      { Заданное число для каждого массива }
  sra, srb, src: real;   { Среднее арифметическое для каждого массива }
  f: TextFile;          { файловая переменная }
```



PascalABC.NET

Файл Плавка Вид Программа Сервис Модули Помощь

Program.pas

{ } Глобальные имена

```
{ Процедура ввода одномерного массива из файла }
procedure Get(var x: mas; var n: integer; var f: TextFile);
var
  i: integer;
begin
  readln(f, n);
  for i := 1 to n do
    read(f, x[i]);
end;

{ Процедура вывода одномерного массива в файл }
procedure Put(const x: mas; n: integer; name: string; var f: TextFile);
var
  i: integer;
begin
  writeln(f, 'The array ', name, ' of ', n:2, ' elements');
  for i := 1 to n do
    write(f, x[i]:8:2);
  writeln(f); writeln(f);
end;
```

PascalABC.NET

Файл Плавка Вид Программа Сервис Модули Помощь

Program.pas

{ } Глобальные имена


```
{ Функция поиска среднего арифметического элементов массива, больших заданного }
function Average(const x: mas; n: integer; v: real): real;
var
    k, i: integer;
begin
    Result := 0;
    for i := 1 to n do
        if x[i] > v then
            begin
                Result := Result + x[i];
                k := k + 1;
            end;
    Result := Result / k;
end;
```

```
PascalABC.NET
Файл  Правка  Вид  Программа  Сервис  Модули  Помощь
[Icons]
Program.pas
{ } Глобальные имена
begin
  { Ввод исходных данных }
  if ParamCount < 2 then
  begin
    writeln('Недостаточно параметров');
    exit;
  end;
  if not FileExists(ParamStr(1)) then
  begin
    writeln('Невозможно открыть файл '', ParamStr(1), '' для чтения');
    exit;
  end;
  AssignFile(f, ParamStr(1));
  Reset(f);
  Get(a, na, f);
  Get(b, nb, f);
  Get(c, nc, f);
  readln(f, pa, pb, pc);
  CloseFile(f);

  { Определение в каждом из массивов среднего арифметического элементов,
    больших заданного числа }
  sra := Average(a, na, pa);
  srb := Average(b, nb, pb);
  src := Average(c, nc, pc);
end;
```


```
PascalABC.NET
Файл  Правка  Вид  Программа  Сервис  Модули  Помощь
[Icons]
Program.pas
{ } Глобальные имена
{ Вывод полученных результатов }
AssignFile(f, ParamStr(2));
Rewrite(f);
if (sra = srb) and (srb = src) then
  writeln(f, 'Среднее арифметическое совпадает во всех массивах и равно ', s
else if (sra = srb) and (sra < src) then
  writeln(f, 'Массивы А и В имеют одинаковое минимальное среднее арифметиче
else if (sra = src) and (sra < srb) then
  writeln(f, 'Массивы А и С имеют одинаковое минимальное среднее арифметиче
else if (srb = src) and (srb < sra) then
  writeln(f, 'Массивы В и С имеют одинаковое минимальное среднее арифметиче
else if (sra < srb) and (sra < src) then
  writeln(f, 'Массив А имеет минимальное среднее арифметическое, равно ',
else if (srb < sra) and (srb < src) then
  writeln(f, 'Массив В имеет минимальное среднее арифметическое, равно ',
else
  writeln(f, 'Массив С имеет минимальное среднее арифметическое, равно ',
CloseFile(f);
end.
```

# ПЕРВЫЙ ТЕСТ

- ▶ В первом тесте результат должен быть следующий:  
в массивах **A** и **B** среднее арифметическое одинаково и равно 12.
  - ▶ Однако мы видим, что программа выдаёт другой результат.
- 

in1.txt - Notepad2

Файл Изменить Вид Настройки ?




```
1 5
2 8 0 -1 9 19 5
3 4
4 11 12 13 -5
5 6
6 -1 -2 -5 10 20 40
7 2 3 9
```

Строка 2 : 7    Столбец 14    Выделено 61 байт    ANSI    CR+LF    INS    Обычный текст

out.txt - Notepad2

Файл Изменить Вид Настройки ?



```
1 Массив В имеет минимальное среднее арифметическое, равное 10.00
2
```

Строка 1 : 2    Столбец 1    Выделено 66 байт    ANSI    CR+LF    INS    Обычный текст

# ПЕРВЫЙ ТЕСТ

- ▶ Попробуем пройти программу по шагам и для начала убедимся, что ввод осуществляется правильно.
- ▶ Для этого выполним «Шаг без входа в подпрограмму» и будем наблюдать значения переменных на вкладке «Локальные переменные».
- ▶ Пока не будем заходить в подпрограмму ввода, т.е. будем повторять «Шаг без входа в подпрограмму».

PascalABC.NET

Файл Правка Вид Программа Сервис Модули Помощь

•Program.pas\*

Глобальные имена

```
begin
  { Ввод исходных данных }
  if ParamCount < 3 then
  begin
    writeln('Неверное количество параметров');
    exit;
  end;
  if not FileExists(ParamStr(1)) then
  begin
    writeln('Невозможно открыть файл "', ParamStr(1), '" для чтения');
    exit;
  end;
  AssignFile(f, ParamStr(1));
  Reset(f);
  Get(a, na, f);
  Get(b, nb, f);
  Get(c, nc, f);
  readln(f, pa, pb, pc);
  CloseFile(f);

  { Определение в каждом из массивов среднего арифметического
    больших заданного числа }
  sra := Average(a, na, pa);
  srb := Average(b, nb, pb);
  src := Average(c, nc, pc);
```

Выполнить F9  
Выполнить без связи с оболочкой Shift+F9  
Параметры командной строки...  
Завершить Ctrl+F2  
Компилировать Ctrl+F9  
Перекомпилировать все Ctrl+Shift+F9  
Шаг без входа в подпрограмму F8  
Шаг с входом в подпрограмму F7  
Шаг с выходом из подпрограммы F10  
Идти к курсору F4



# ПЕРВЫЙ ТЕСТ

- ▶ Когда мы пройдём ввод исходных данных, мы увидим, что количество элементов массива **V** равно 5, хотя в файле с исходными данными было записано 4.

```
AssignFile(f, ParamStr(1));
Reset(f);
Get(a, na, f);
Get(b, nb, f);
Get(c, nc, f);
readln(f, pa, pb, pc);
CloseFile(f);
```

```
{ Определение в каждом из массивов среднего арифметического
  больших заданного числа }
```

```
sra := Average(a, na, pa);
srb := Average(b, nb, pb);
src := Average(c, nc, pc);
```

```
{ Вывод полученных результатов }
```

```
AssignFile(f, ParamStr(2));
Rewrite(f);
if (sra = srb) and (srb = src) then
```

#### Локальные переменные


Переменная	Значение
Глобальные переменные	
a	(8,0,-1,9,19,0,0,0,0,0,...)
b	(4,11,12,13,-5,0,0,0,0,0,...)
c	(-1,-2,-5,10,20,40,0,0,0,0,...)
na	5
nb	5
nc	6
pa	2

Окно вывода | Список ошибок | Сообщения компилятора | Локальные переменные | Просмотр

Компиляция прошла успешно (96 строк), 3 предупреждений

in1.txt - Notepad2

Файл Изменить Вид Настройки ?




```
1 5
2 8 0 -1 9 19 5
3 4
4 11 12 13 -5
5 6
6 -1 -2 -5 10 20 40
7 2 3 9
```

Строка 2 : 7 Столбец 14 Выделено 61 байт ANSI CR+LF INS Обычный текст

out.txt - Notepad2

Файл Изменить Вид Настройки ?



```
1 Массив В имеет минимальное среднее арифметическое, равное 10.00
2
```


Строка 1 : 2 Столбец 1 Выделено 66 байт ANSI CR+LF INS Обычный текст

# ПЕРВЫЙ ТЕСТ

- ▶ Если мы внимательно посмотрим на файл с исходными данными, мы увидим, что во второй строке записано 6 чисел, хотя в массиве **A** должно быть 5 элементов массива.
- ▶ В данном случае шестое число 5 при вводе было прочитано как количество элементов массива **B**, что и привело к искажению исходных данных.

in1.txt - Notepad2

Файл Изменить Вид Настройки ?




```
1 5
2 8 0 -1 9 19 5
3 4
4 11 12 13 -5
5 6
6 -1 -2 -5 10 20 40
7 2 3 9
```

Строка 2 : 7 Столбец 14 Выделено 61 байт ANSI CR+LF INS Обычный текст

out.txt - Notepad2

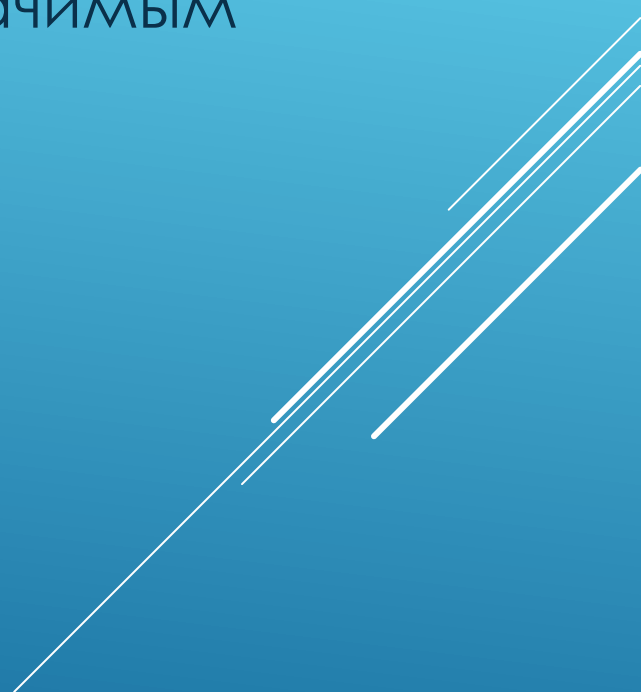
Файл Изменить Вид Настройки ?



```
1 Массив В имеет минимальное среднее арифметическое, равное 10.00
2
```


Строка 1 : 2 Столбец 1 Выделено 66 байт ANSI CR+LF INS Обычный текст

# ПЕРВЫЙ ТЕСТ

- ▶ Если мы удалим это число и вновь запустим программу, мы увидим корректный результат.
  - ▶ В данном случае мы нашли ошибку не в программе, а в исходных данных, что, впрочем, тоже является значимым результатом.
- 

in1.txt - Notepad2

Файл Изменить Вид Настройки ?




```
1 5
2 8 0 -1 9 19
3 4
4 11 12 13 -5
5 6
6 -1 -2 -5 10 20 40
7 2 3 9
```

Строка 2 : 7    Столбец 12    Выделено 59 байт    ANSI    CR+LF    INS    Обычный текст

out.txt - Notepad2

Файл Изменить Вид Настройки ?



```
1 Массивы A и B имеют одинаковое минимальное среднее
арифметическое, равное 12.00
2
```

Строка 1 : 2    Столбец 1    Выделено 82 байт    ANSI    CR+LF    INS    Обычный текст

# ВТОРОЙ ТЕСТ

- ▶ Рассмотрим другой пример.
- ▶ Здесь в массиве **A** нет элементов, больших заданного числа, поэтому среднее арифметическое должно быть равно 0 и должно быть минимальным.
- ▶ В принципе, следовало бы проверять наличие в массивах элементов, больших заданного числа, и в случае отсутствия таких элементов не рассматривать этот массив или вообще считать такой набор данных неадекватным, но в данной программе реализован упрощённый подход.
- ▶ Но мы видим, что в качестве ответа выдаётся массив **C**.



in2.txt - Notepad2

Файл Изменить Вид Настройки ?

File Edit View Settings ?

File Edit View Settings ?

```
1 5
2 8 0 -1 9 19
3 4
4 11 12 13 -5
5 6
6 -1 -2 -5 10 20 40
7 55 8 9
```

Строка 4 : 7 Столбец 12 Выделено 60 байт ANSI CR+LF INS Обычный текст

out.txt - Notepad2

Файл Изменить Вид Настройки ?

File Edit View Settings ?

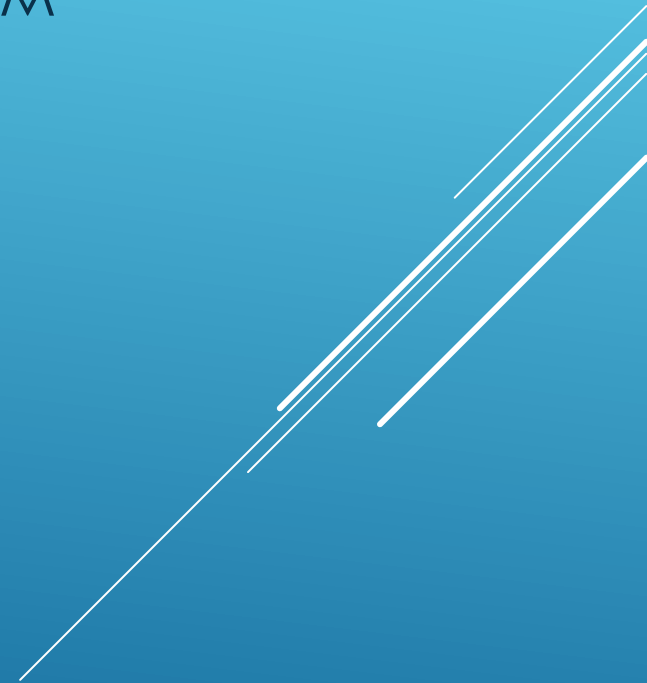
File Edit View Settings ?

```
1 Массив С имеет минимальное среднее арифметическое, равное 23.33
2
```

Строка 1 : 2 Столбец 1 Выделено 66 байт ANSI CR+LF INS Обычный текст

# ВТОРОЙ ТЕСТ

- ▶ Давайте поставим точку прерывания на первый вызов функции подсчёта среднего арифметического и запустим программу.
- ▶ Выполнение остановится на операторе, помеченном точкой прерывания.



PascalABC.NET

Файл Плавка Вид Программа Сервис Модули Помощь

•Program.pas

{ } Глобальные имена

```
{ Определе  
БОЛЬШИХ :  
sra := Ave  
srb := Ave  
src := Ave  
  
{ Вывод по  
AssignFile(f, ParamStr(2));  
Rewrite(f);  
if (sra = srb) and (srb = src) then  
    writeln(f, 'Среднее арифметическое совпадает во всех массивах');  
else if (sra = srb) and (sra < src) then  
    writeln(f, 'Массивы А и В имеют одинаковое минимальное среднее арифметическое');  
else if (sra = src) and (sra < srb) then  
    writeln(f, 'Массивы А и С имеют одинаковое минимальное среднее арифметическое');  
else if (srb = src) and (srb < sra) then  
    writeln(f, 'Массивы В и С имеют одинаковое минимальное среднее арифметическое');  
else if (sra < srb) and (sra < src) then  
    writeln(f, 'Массив А имеет минимальное среднее арифметическое');  
else if (srb < sra) and (srb < src) then  
    writeln(f, 'Массив В имеет минимальное среднее арифметическое');  
else  
    writeln(f, 'Массив С имеет минимальное среднее арифметическое');  
CloseFile(f);
```

Выполнить F9

Выполнить без связи с оболочкой Shift+F9

Параметры командной строки...

Завершить Ctrl+F2

Компилировать Ctrl+F9

Перекомпилировать все Ctrl+Shift+F9

Шаг без входа в подпрограмму F8

Шаг с входом в подпрограмму F7

Шаг с выходом из подпрограммы F10

Идти к курсору F4

его арифметического

# ВТОРОЙ ТЕСТ

- ▶ Можно посмотреть значения локальных переменных и убедиться, что всё введено правильно, после чего выполним «Шаг с входом в подпрограмму».



PascalABC.NET

Файл Плавка Вид Программа Сервис Модули Помощь

•Program.pas [Запу...]


{ } Глобальные имена

```
{ Определе  
больших  
sra := Ave  
srb := Ave  
src := Ave  
  
{ Вывод по  
AssignFile(f, ParamStr(2));  
Rewrite(f);  
if (sra = srб) and (srб = src) then  
    writeln(f, 'Среднее арифметическое совпадает во всех массивах');  
else if (sra = srб) and (sra < src) then  
    writeln(f, 'Массивы А и В имеют одинаковое минимальное среднее арифметическое');  
else if (sra = src) and (sra < srб) then  
    writeln(f, 'Массивы А и С имеют одинаковое минимальное среднее арифметическое');  
else if (srб = src) and (srб < sra) then  
    writeln(f, 'Массивы В и С имеют одинаковое минимальное среднее арифметическое');  
else if (sra < srб) and (sra < src) then  
    writeln(f, 'Массив А имеет минимальное среднее арифметическое');  
else if (srб < sra) and (srб < src) then  
    writeln(f, 'Массив В имеет минимальное среднее арифметическое');  
else  
    writeln(f, 'Массив С имеет минимальное среднее арифметическое');  
CloseFile(f);
```

Выполнить F9  
Выполнить без связи с оболочкой Shift+F9  
Параметры командной строки...  
Завершить Ctrl+F2  
Компилировать Ctrl+F9  
Перекомпилировать все Ctrl+Shift+F9  
Шаг без входа в подпрограмму F8  
Шаг с входом в подпрограмму F7  
Шаг с выходом из подпрограммы F10  
Идти к курсору F4

его арифметического

# ВТОРОЙ ТЕСТ

- ▶ Давайте попробуем не выполнять все шаги функции, а сразу перейдём к последнему оператору функции и посмотрим значения переменных.
  - ▶ Для перехода к последнему оператору можно поставить на него курсор и выполнить команду «Идти к курсору».
- 

PascalABC.NET

Файл Плавка Вид Программа Сервис Модули Помощь

•Program.pas [Запу...]

{ } Глобальные имена

```
{ ФУНКЦИЯ ПОИСКА ЭЛЕМЕНТОВ МАССИВА, БОЛЬШЕ ЧЕМ
function Ave(v: real): real;
var
  k, i: integer;
begin
  Result := 0;
  for i := 1 to k do
    if x[i] > v then
      begin
        Result := Result + x[i];
        k := k + 1;
      end;
  Result := Result / k;
end;

begin
  { Ввод исходных данных }
  if ParamCount < 2 then
    begin
      writeln('Недостаточно параметров');
      exit;
    end;
  if not FileExists(ParamStr(1)) then
    begin
```

Выполнить F9  
Выполнить без связи с оболочкой Shift+F9  
Параметры командной строки...  
Завершить Ctrl+F2  
Компилировать Ctrl+F9  
Перекомпилировать все Ctrl+Shift+F9  
Шаг без входа в подпрограмму F8  
Шаг с входом в подпрограмму F7  
Шаг с выходом из подпрограммы F10  
Идти к курсору F4

ЭЛЕМЕНТОВ МАССИВА, БОЛЬШЕ ЧЕМ  
v: real): real;

Result := 0;

Result := Result / k;

writeln('Недостаточно параметров');

exit;

FileExists(ParamStr(1)) then

# ВТОРОЙ ТЕСТ

- ▶ Для того, чтобы увидеть значение переменной, можно во время пошагового выполнения навести курсор на имя нужной переменной, и возле него появится всплывающее окошко.
- ▶ В данном случае мы видим, что значение переменной **k** равно 0.



PascalABC.NET

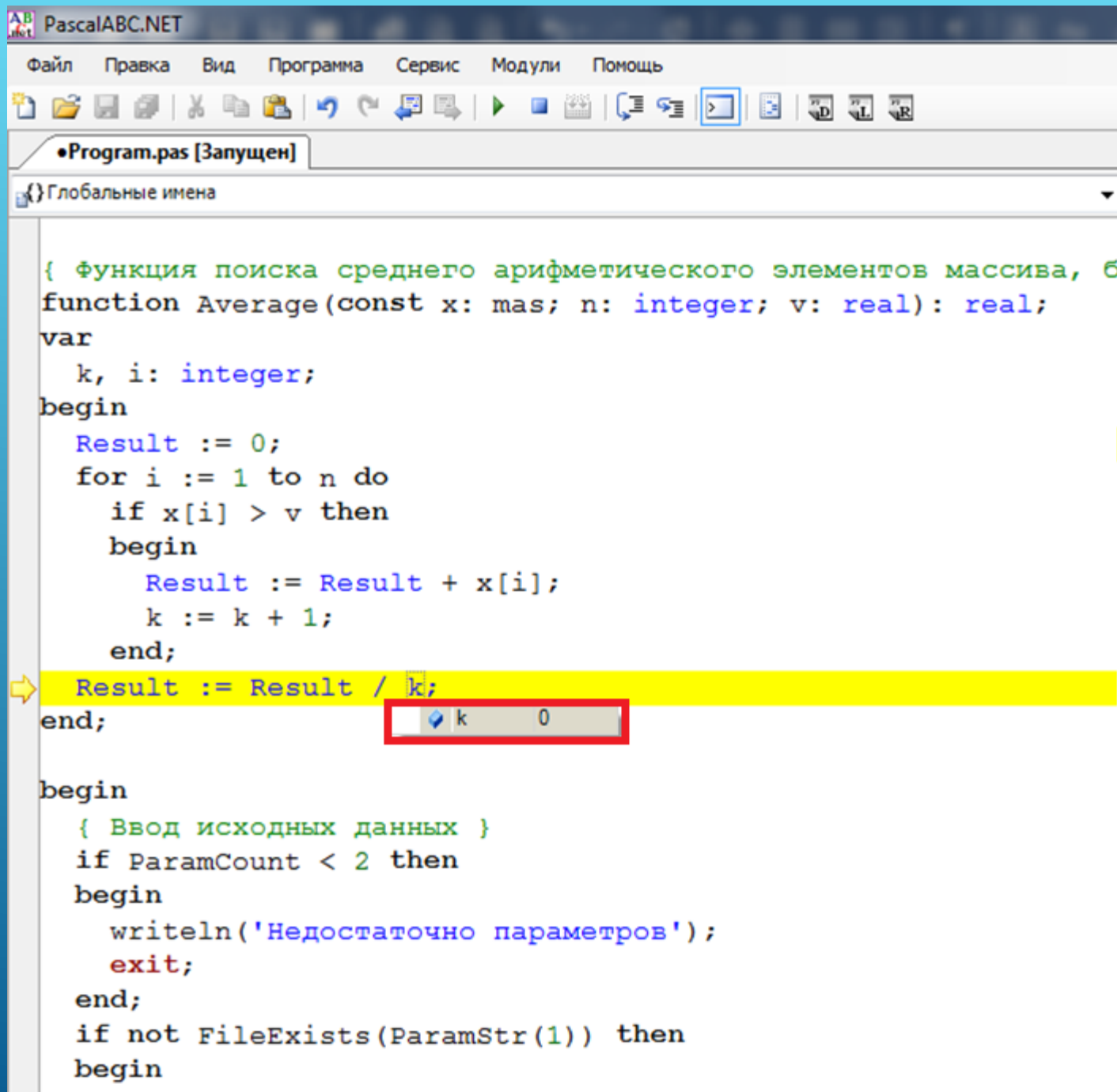
Файл Плавка Вид Программа Сервис Модули Помощь

•Program.pas [Запущен]

{ } Глобальные имена

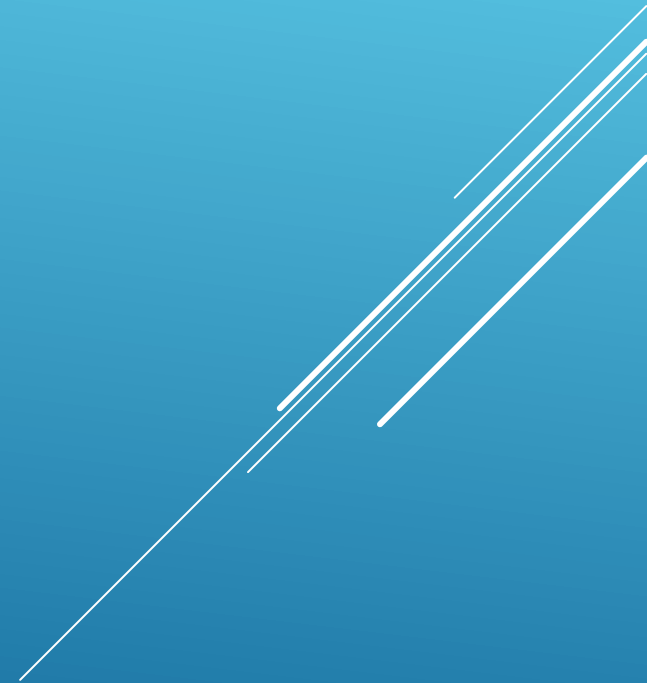
```
{ ФУНКЦИЯ ПОИСКА СРЕДНЕГО АРИФМЕТИЧЕСКОГО ЭЛЕМЕНТОВ МАССИВА, БОЛЬШЕ ЧЕМ v }
function Average(const x: mas; n: integer; v: real): real;
var
  k, i: integer;
begin
  Result := 0;
  for i := 1 to n do
    if x[i] > v then
      begin
        Result := Result + x[i];
        k := k + 1;
      end;
  Result := Result / k;
end;

begin
  { Ввод исходных данных }
  if ParamCount < 2 then
    begin
      writeln('Недостаточно параметров');
      exit;
    end;
  if not FileExists(ParamStr(1)) then
    begin
```



# ВТОРОЙ ТЕСТ

- ▶ Выполним операцию деления и увидим, что значение результата не определено.



PascalABC.NET

Файл Плавка Вид Программа Сервис Модули Помощь

•Program.pas [Запущен]

{ } Глобальные имена

```
{ функция поиска среднего арифметического элементов массива, б
function Average(const x: mas; n: integer; v: real): real;
var
  k, i: integer;
begin
  Result := 0;
  for i := 1 to n do
    if x[i] > v then
      begin
        Result := Result + x[i];
        k := k + 1;
      end;
  Result := Result / k;
end;
begin
  { Ввод исходных данных }
  if ParamCount < 2 then
    begin
      writeln('Недостаточно параметров');
      exit;
    end;
  if not FileExists(ParamStr(1)) then
    begin
```

Result NaN

# ВТОРОЙ ТЕСТ

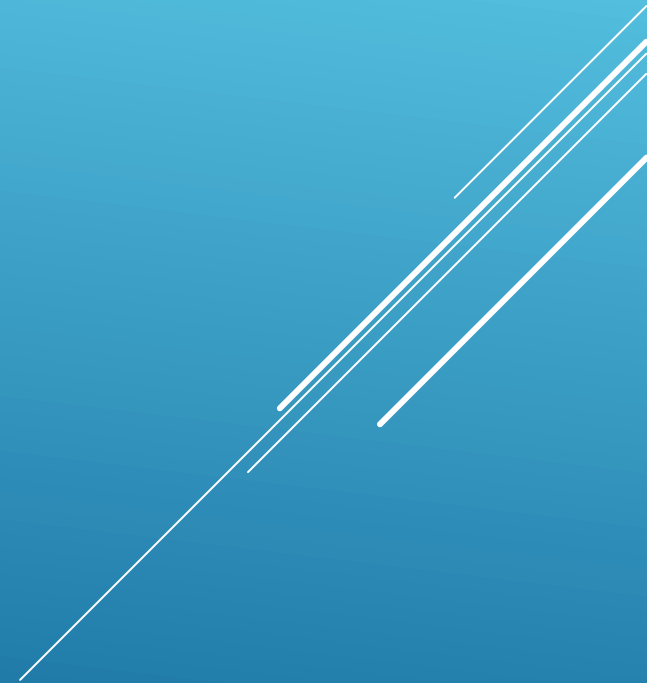
- ▶ Если выполнить по шагам основную программу, можно увидеть, что в случае такого результата в одном массиве, все операции сравнения оказываются ложными, и просто выполняется оператор, входящий в последний оператор **else**.

```
PascalABC.NET
Файл  Правка  Вид  Программа  Сервис  Модули  Помощь
[Иконки панели инструментов]
•Program.pas [Запущен]
{} Глобальные имена
{ Определение в каждом из массивов среднего арифметического
  больших заданного числа }
sra := Average(a, na, pa);
srb := Average(b, nb, pb);
src := Average(c, nc, pc);

{ Вывод полученных результатов }
AssignFile(f, ParamStr(2));
Rewrite(f);
if (sra = srb) and (srb = src) then
  writeln(f, 'Среднее арифметическое совпадает во всех массивах');
else if (sra = srb) and (sra < src) then
  writeln(f, 'Массивы А и В имеют одинаковое минимальное среднее арифметическое');
else if (sra = src) and (sra < srb) then
  writeln(f, 'Массивы А и С имеют одинаковое минимальное среднее арифметическое');
else if (srb = src) and (srb < sra) then
  writeln(f, 'Массивы В и С имеют одинаковое минимальное среднее арифметическое');
else if (sra < srb) and (sra < src) then
  writeln(f, 'Массив А имеет минимальное среднее арифметическое');
else if (srb < sra) and (srb < src) then
  writeln(f, 'Массив В имеет минимальное среднее арифметическое');
else
  writeln(f, 'Массив С имеет минимальное среднее арифметическое');
CloseFile(f);
end.
```

## ВТОРОЙ ТЕСТ

- ▶ Изменим функцию так, чтобы деление выполнялось только при  $k$ , не равном 0, и мы увидим, что теперь программа выдаёт ожидаемый результат.



PascalABC.NET

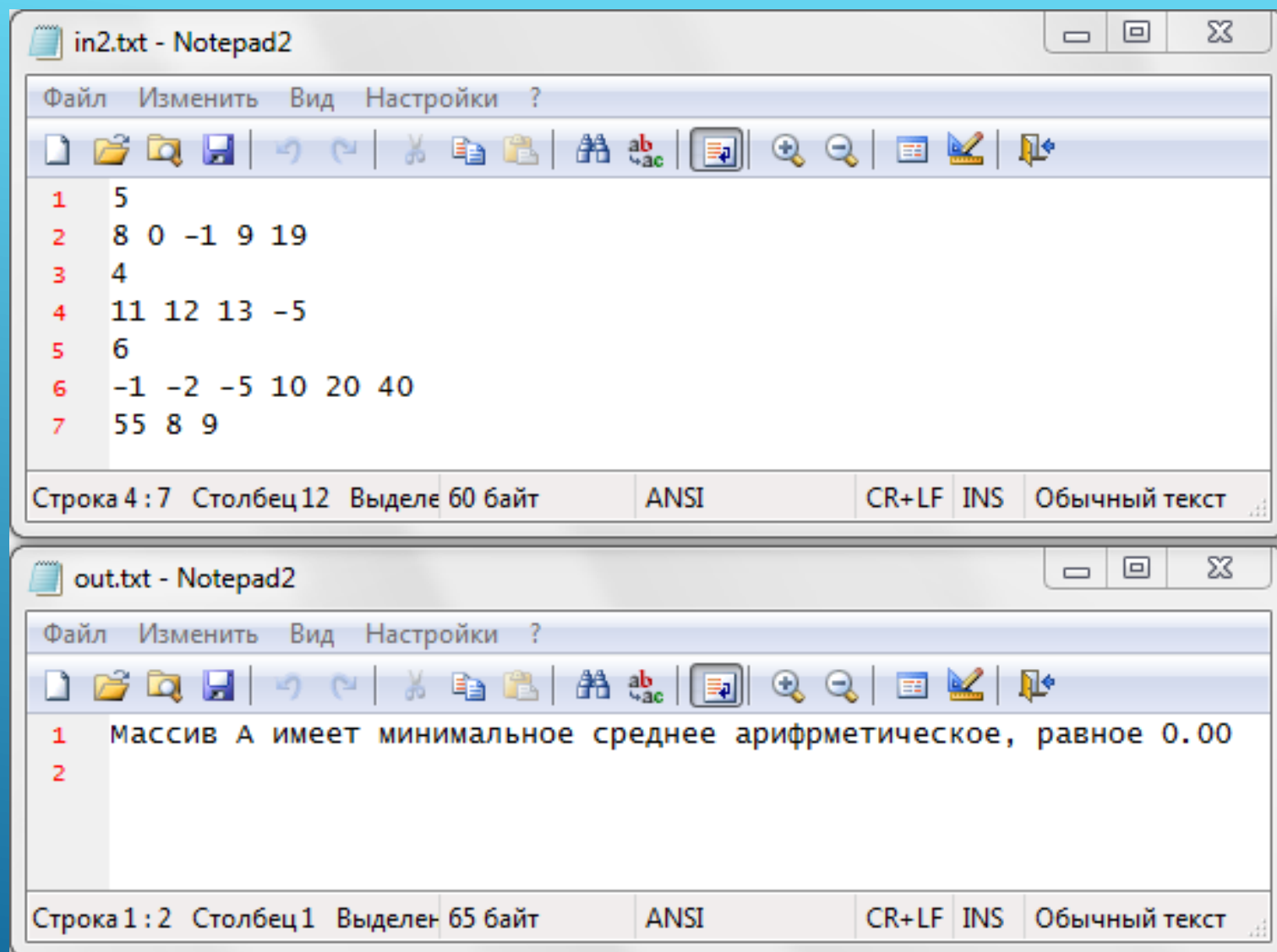
Файл Плавка Вид Программа Сервис Модули Помощь

•Program.pas

{ } Глобальные имена

```
{ ФУНКЦИЯ ПОИСКА СРЕДНЕГО АРИФМЕТИЧЕСКОГО ЭЛЕМЕНТОВ МАССИВА, БОЛЬШЕ ЧЕМ В
function Average(const x: mas; n: integer; v: real): real;
var
  k, i: integer;
begin
  Result := 0;
  for i := 1 to n do
    if x[i] > v then
      begin
        Result := Result + x[i];
        k := k + 1;
      end;
  if k <> 0 then
    Result := Result / k;
end;

begin
  { Ввод исходных данных }
  if ParamCount < 2 then
    begin
      writeln('Недостаточно параметров');
      exit;
    end;
  if not FileExists(ParamStr(1)) then
```







# ЗАКЛЮЧЕНИЕ




# ЗАКЛЮЧЕНИЕ

- ▶ Мы познакомились со средствами отладки программ, которые предоставляются современными средами программирования.
  - ▶ Эти средства позволяют организовать пошаговое выполнение программы и наблюдать значения переменных.
- 

# ЗАКЛЮЧЕНИЕ

- ▶ Пошаговое выполнение
    - шаг без входа в подпрограмму
    - шаг с входом в подпрограмму
    - шаг с выходом из подпрограммы
    - идти к курсору
    - точки прерывания
- 

# ЗАКЛЮЧЕНИЕ

- ▶ Наблюдение значений переменных
    - локальные переменные
    - просмотр выражений
    - наведение курсора на имя
- 

# ЗАКЛЮЧЕНИЕ

- ▶ Все современные среды программирования предоставляют указанные возможности.
  - ▶ Конечно, их реализация может несколько отличаться, но зная, что искать, вы всегда найдёте эти средства.
- 