

Муниципальное бюджетное образовательное учреждение  
высшего образования «Самарская академия  
государственного и муниципального управления»

Юридический факультет  
Кафедра Общонаучных дисциплин и права

Задание №3  
по дисциплине «Программная инженерия»  
на тему «Динамические массивы. Файлы»

Выполнил Иванов И.И.,  
студент группы 921-Д

Проверила Коробецкая А.А.

## Задача

Написать программу для работы с динамическими числовыми массивами с поддержкой возможностей:

1. Одновременная работа с несколькими одномерными массивами.
2. Случайная генерация массива.
3. Ввод массива пользователем вручную.
4. Просмотр массивов (вывод на экран).
5. Обработка массива по индивидуальному заданию.
6. Чтение и запись массивов в произвольный текстовый файл (CSV).

Организовать интерфейс программы в виде меню.

**Индивидуальное задание:** Вычислить среднее значение элементов массива.

## Словесное описание

Программа организует работу с множеством одномерных массивов. Массивы могут быть введены тремя способами: считаны из текстового файла, введены вручную или сгенерированы случайным образом. Длина создаваемых массивов произвольна.

Для любого массива может быть рассчитано среднее значение элементов.

Вывод результатов расчетов и исходных массивов осуществляется на экран. Сами массивы могут быть сохранены в текстовый файл.

## Математическая модель

Среднее значение массива  $A$  длиной  $n$  вычисляется по формуле:

$$m(A) = \sum_{i=1}^n A_i \quad (1)$$

## Описание работы программы

При запуске программы выводится меню, в котором пользователь может выбрать следующие действия:

1. Справка.
2. Просмотр массивов.
3. Загрузка массивов из файла.
4. Сохранение массивов в файл.
5. Создание нового массива.
6. Удаление всех массивов.
7. Выход.

При выборе пункта меню «Справка» на экран выводится описание задачи и возможностей программы. Нажатие любой клавиши приводит к возврату в меню.

При выборе пункта меню «Просмотр массивов» на экран выводится количество массивов и предлагается ввести номер массива для просмотра. Ввод –1 выводит на экран все массивы, ввод отрицательных значений кроме –1, или значений, превосходящих количество массивов, выводится сообщение об ошибке. Для каждого массива на экран выводится количество значений, среднее и сами значения элементов массива. При просмотре всех массивов они выводятся на экран по 3, для продолжения просмотра пользователю необходимо нажать любую клавишу. После просмотра нажатие любой клавиши возвращает пользователя в главное меню.

При выборе пункта меню «Загрузка массивов из файла» пользователю предлагается ввести имя файла для загрузки. Если файл существует, он считывается построчно, в противном случае выводится сообщение об ошибке. Если при обработке очередной строки не удалось считать хотя бы одно значение (присутствуют посторонние символы), соответствующий массив не будет создан, будет выведено сообщение об ошибке с указанием номера строки, но считывание файла не прервется. После завершения считывания будет выведено количество успешно загруженных массивов. Загруженные массивы добавляются к ранее созданным. Для возврата в меню пользователь должен нажать любую клавишу.

При выборе пункта меню «Сохранение массивов в файл» пользователю предлагается ввести имя файла. Если файл существует, будет выдан запрос на перезапись файла. Значения массивов сохраняются с точностью до 2 знаков после запятой. После успешной записи выдается соответствующее сообщение. Возврат в главное меню происходит при нажатии любой клавиши.

При выборе пункта меню «Создание нового массива» выводится подменю:

1. Случайная генерация.
2. Ручной ввод.
3. Возврат в главное меню.

При выборе случайной генерации, пользователю требуется ввести количество элементов массива (не менее одного), минимальное и максимальное значения (произвольные вещественные числа). После генерации значений выводится соответствующее сообщение.

При выборе ручного ввода пользователь сначала указывает количество элементов массива, а затем вводит их по одному. В случае неверного ввода значение требуется ввести повторно.

В обоих случаях созданный массив добавляется к уже существующим. После нажатия любой клавиши происходит возврат к подменю.

При выборе пункта главного меню «Удаление всех массивов» выводится запрос на подтверждение удаления. При подтверждении все сохраненные данные удаляются, и выводится сообщение. Возврат в главное меню происходит после нажатия любой клавиши.

Пункт меню «Выход» завершает работу приложения без сохранения данных.

**Входные данные:** массивы произвольной длины, содержащие вещественные значения.

**Выходные данные:** среднее значение каждого массива.

**Интерфейс ввода-вывода:** текстовый (командная строка) с использованием псевдографики.

**Возможные ошибки:**

<b>Код</b>	<b>Сообщение</b>	<b>Пояснение</b>	<b>Реакция программы</b>
01	Требуется ввести целое число в диапазоне -32768..32767	Неверный ввод пользователем целых чисел (текст, дробное число или выход за границы диапазона Integer)	Сообщение об ошибке и запрос повторного ввода числа
02	Требуется ввести число	Неверный ввод пользователем вещественных чисел (длины массива)	Сообщение об ошибке и запрос повторного ввода числа
03	Размер массива должен быть от 1 до 1000	При создании массива установлено ограничение в 1000 значений максимум, чтобы избежать слишком долгой работы программы	Сообщение об ошибке и отмена создания массива
04	Неверный номер массива	При просмотре массивов пользователь ввел неверный номер (отрицательный или больше числа массивов)	Сообщение об ошибке и возврат в главное меню
05	Не удалось считать массив из строки	При считывании массивов из текстового файла возникла ошибка преобразования текста в число	Сообщение об ошибке с указанием номера строки в файле, продолжение считывания со следующей строки
06	Файл не найден	Пользователь ввел имя несуществующего файла	Сообщение об ошибке и возврат в главное меню

**Инструментальные средства**

**Язык программирования:** Pascal

**Среда программирования:** Free Pascal Compiler 2.6

## Системные требования

**Операционная система:** Windows, DOS.

**Аппаратная платформа:** персональный компьютер архитектуры x86/64, монитор, клавиатура.

**Дополнительное программное обеспечение:** Компилятор Free Pascal Compiler 2.6 или новее.

## Декомпозиция задачи

Примечание. В описании процедуры указываются действия, выполняемые непосредственно в этой процедуре.

Выполним декомпозицию задачи на подпрограммы «сверху вниз». Декомпозицию верхнего уровня выполним в соответствии с пунктами меню.

### Запуск главного меню (RunMainMenu)

*Параметры:* -

*Возвращает:* -

*Описание:* Выводит меню на экран и проверяет нажатие управляющих клавиш пользователем.

### Отображение меню на экране (ShowMainMenu)

*Параметры:* -

*Возвращает:* -

*Описание:* Отображает главное меню в виде псевдографики.

### Вывод справки (ShowHelp)

*Параметры:* -

*Возвращает:* -

*Описание:* Выводит на экран справку по программе.

### Вывод массивов (ShowArrays)

*Параметры:* -

*Возвращает:* -

*Описание:* Отображает общее количество массивов, для одного выбранного массива, либо для всех массивов.

Вывод всех массивов и отдельного массива выносится в отдельные процедуры.

### Чтение массивов из текстового файла (ReadArraysFromTextFile)

*Параметры:* -

*Возвращает:* -

*Описание:* Считывает массивы из текстового файла, имя которого вводит пользователь. Если файл не найден, происходит возврат в главное меню. Если какой-то из массивов был считан с ошибками, он будет пропущен. После

завершения считывания файла отображается общее количество добавленных массивов.

Считывание отдельного массива из текстовой строки выносится в отдельную функцию.

### **Сохранение массивов в текстовый файл (WriteArraysToTextFile)**

*Параметры:* -

*Возвращает:* -

*Описание:* Все массивы сохраняются в файл, имя которого вводит пользователь. Если файл существует, пользователю предлагается подтвердить замену файла. После завершения записи файла отображается информационное сообщение и пользователю предлагается вернуться в главное меню.

Преобразование отдельного массива в строку выполняется в отдельной функции.

### **Создание нового массива (CreateNewArray)**

*Параметры:* -

*Возвращает:* -

*Описание:* Добавляет один новый массив к уже существующим. Пользователю необходимо через подменю выбрать один из двух вариантов: случайная генерация значений массива или ввод значений вручную. Созданный массив добавляется к уже существующим.

Каждый способ создания массива выносится в отдельную функцию.

### **Удаление данных из всех массивов (ClearAll)**

*Параметры:* -

*Возвращает:* -

*Описание:* Полностью удаляет данные во всех массивах. У пользователя запрашивается подтверждение на удаление данных.

### **Вывод на экран всех массивов (PrintAllArrays)**

*Параметры:* -

*Возвращает:* -

*Описание:* Содержит цикл для перебора всех массивов. После каждого третьего включается пауза.

### **Вывод на экран одного массива (PrintArray)**

*Параметры:* массив, который необходимо вывести

*Возвращает:* -

*Описание:* Выводит количество значений в массиве, среднее значение и все элементы массива в строку.

### **Вывод подменю создания массива на экран (ShowNewArrayMenu)**

*Параметры:* -

*Возвращает:* -

*Описание:* Отображает подменю в виде псевдографики.

### **Создание нового массива через ручной ввод (InputArray)**

*Параметры:* -

*Возвращает:* созданный массив

*Описание:* Запрашивает у пользователя размер массива. Затем пользователь вводит каждое значение.

Ввод размера массива и ввод значений вынесены в отдельные функции.

### **Создание нового массива путем случайное генерации (GenerateArray)**

*Параметры:* -

*Возвращает:* созданный массив

*Описание:* Запрашивает у пользователя размер массива, минимальное и максимальное значения. Затем массив генерируется в заданных пределах.

Ввод размера массива и ввод минимального и максимального значений выполняется через отдельные функции.

### **Ввод размера массива (ReadSize)**

*Параметры:* -

*Возвращает:* размер массива (целое число от 1 до 1000)

*Описание:* Запрашивает у пользователя ввод целого числа, проверяет диапазон значений.

Ввод целого числа вынесен в отдельную функцию.

### **Запись массива в строку (WriteArrayToString)**

*Параметры:* записываемый массив

*Возвращает:* полученная строка

*Описание:* В цикле записывает значения массива в строку с точностью 2 знака после запятой. Разделители чисел – точка с запятой (;).

### **Чтения массива из строки (ReadArrayFromString)**

*Параметры:* исходная строка

*Возвращает:* считанный массив

*Описание:* Считывает значения, разделенные точкой с запятой, из строки в массив. В случае возникновения ошибки считывание прерывается и функция возвращает nil.

### **Вычисление среднего (GetAverage)**

*Параметры:* массив

*Возвращает:* среднее значение (вещественное число)

*Описание:* Вычисляет среднее (сумма значений, деленная на их количество). Если массив пуст, будет возвращен 0.

### **Ввод вещественного значения (ReadValue)**

*Параметры:* -

*Возвращает:* введенное значение

*Описание:* Считывает введенное пользователем значение и преобразует его в число. В случае ошибки ввода значение запрашивается повторно. Отменить ввод нельзя.

### **Ввод целого значения (ReadInt)**

*Параметры:* -

*Возвращает:* введенное значение

*Описание:* Считывает введенное пользователем значение и преобразует его в число. В случае ошибки ввода, либо выхода за пределы диапазона Integer, значение запрашивается повторно. Отменить ввод нельзя.

Во всех действиях возврат в главное меню происходит по нажатию любой клавиши. Вынесем это действие в процедуру.

### **Пауза (PressAnyKey)**

*Параметры:* -

*Возвращает:* -

*Описание:* Ожидает нажатия любой клавиши от пользователя.

Вынесем в процедуру вывод сообщения об ошибке.

### **Показать сообщение об ошибке (PrintError)**

*Параметры:* возникшая ошибка (структура, содержит код ошибки и ее текстовое описание)

*Возвращает:* -

*Описание:* Выводит на экран слово «Ошибка», ее числовой код и текстовое пояснение.

## **Блок-схемы**

Тело программы включает только вызов процедуры для запуска главного меню RunMainMenu, поэтому в виде блок-схемы не показано.

На рисунке 1 представлена блок-схема процедуры RunMainMenu, на рисунке 2 – подпрограммы вычисления среднего значения массива GetAverage. Прочие подпрограммы в виде блок-схем не показаны.



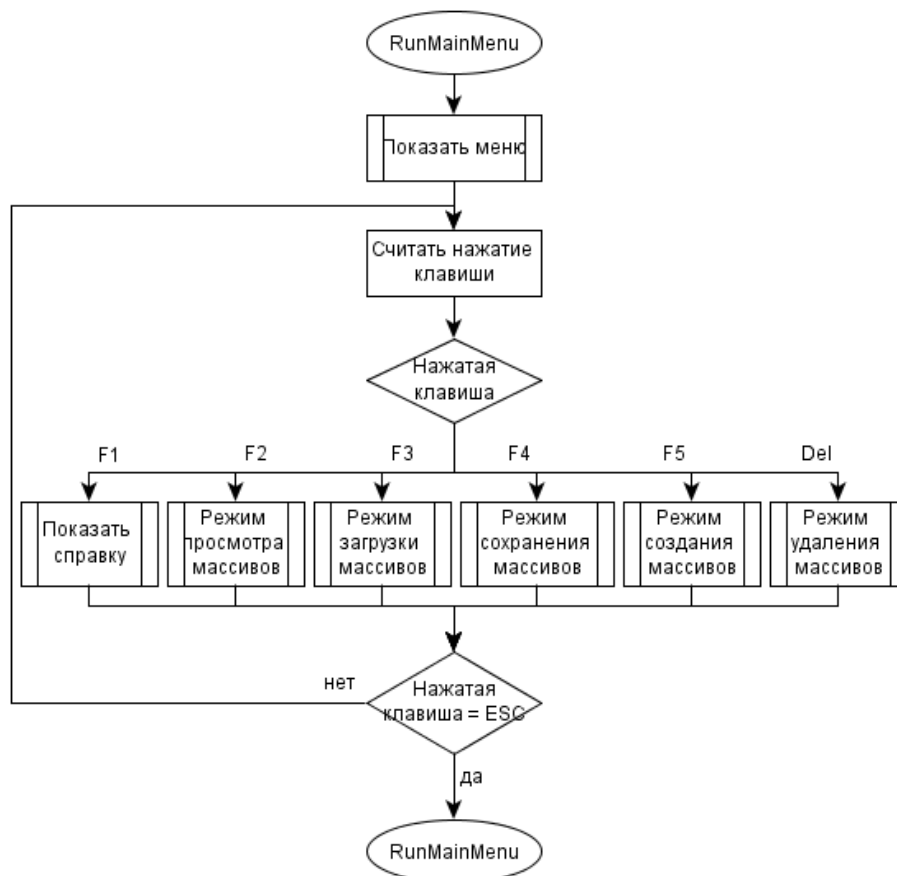


Рис. 1. Блок-схема процедуры запуска главного меню

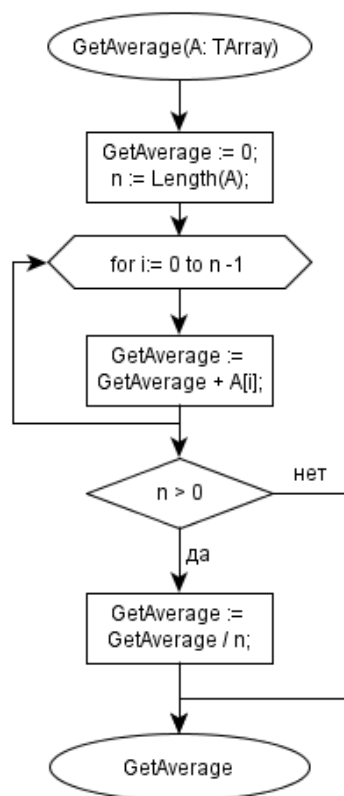


Рис. 2. Блок-схема подпрограммы вычисления среднего значения

## Исходный код

### Программный модуль Average.pas

```
program Average;
{ Написать программу для работы с динамическими числовыми массивами
с поддержкой возможностей:
1. Одновременная работа с несколькими одномерными массивами.
2. Случайная генерация массива.
3. Ввод массива пользователем вручную.
4. Просмотр массивов (вывод на экран).
5. Обработка массива по индивидуальному заданию.
6. Чтение и запись массивов в произвольный текстовый файл (CSV).

Организовать интерфейс программы в виде меню.

Задание на обработку массива:
Вычислить среднее значение элементов массива.}

uses crt, sysutils;

// =====

type
  TValue = Real; //значение массива
  TArray = array of TValue; //массив
  TArrays = array of TArray; //много массивов

var
  AllArrays: TArrays;

// ОШИБКИ =====

type
  //сообщение об ошибке
  TErrorMsg = record
    Code: Byte; //код ошибки
    Text: String; //текст пояснения
  end;

//имеющиеся коды и сообщения об ошибках
const
  Msg_Error = 'Ошибка';

  ERRORMSG_WrongInt: TErrorMsg
  = (Code: 1; Text: 'Требуется ввести целое число в диапазоне -32768..32767');
  ERRORMSG_WrongValue: TErrorMsg
  = (Code: 2; Text: 'Требуется ввести число');
  ERRORMSG_WrongSize: TErrorMsg
  = (Code: 3; Text: 'Размер массива должен быть от 1 до 1000');
  ERRORMSG_WrongArrayIndex: TErrorMsg
  = (Code: 4; Text: 'Неверный номер массива');
  ERRORMSG_CantReadArray: TErrorMsg
  = (Code: 5; Text: 'Не удалось считать массив из строки');
  ERRORMSG_FileNotFound: TErrorMsg
  = (Code: 6; Text: 'Файл не найден');

//вывод сообщения об ошибке
procedure PrintError(const E: TErrorMsg);
begin
  Writeln(Msg_Error, ' ', E.Code, ': ', E.Text);
end;
```

```

// ВВОД ДАННЫХ С КЛАВИАТУРЫ =====

//ввод целого числа
function ReadInt: Integer;
var
  Code: Integer;
  LongIntVal: LongInt; //промежуточное с диапазоном больше Integer
  //запрос ввода с попыткой перевести строку в число
  procedure DoRead(var Value: LongInt; var Code: Integer);
  var
    Str: String;
  begin
    Readln(Str);
    Val(Str, Value, Code);
  end;
begin
  {R-}
  //считывать значения, пока не будет введено без ошибок
  repeat
    DoRead(LongIntVal, Code);
    //выход за пределы диапазона Integer
    if (LongIntVal < Low(Integer)) or (LongIntVal > High(Integer)) then
      Code := -1;
    //ошибка преобразования строки в число
    if (Code <> 0) then
      PrintError(ERRORMSG_WrongInt);
  until Code = 0;
  ReadInt := LongIntVal;
  {R+}
end;

//ввод дробного числа (элемента массива)
function ReadValue: Real;
var Code: Integer;
  //запрос ввода с попыткой перевести строку в число
  procedure DoRead(var Value: Real; var Code: Integer);
  var Str: String;
  begin
    Readln(Str);
    Val(Str, Value, Code);
  end;
begin
  // первое чтение
  DoRead(ReadValue, Code);
  // пока есть ошибка - повторить ввод
  while (Code <> 0) do
    begin
      PrintError(ERRORMSG_WrongValue);
      DoRead(ReadValue, Code);
    end;
end;

procedure PressAnyKey;
begin
  Writeln;
  Writeln('Для возврата в меню нажмите любую клавишу...');
  ReadKey;
end;

// ОБРАБОТКА =====

// вычисление среднего значения элементов массива
// = сумма / количество
// результат дробный независимо от типа данных в массиве

```

```

function GetAverage(A: TArray): Real;
var i, n: Integer;
begin
    //сумма
    GetAverage := 0;
    n := Length(A);
    for i := 0 to n - 1 do
        GetAverage := GetAverage + A[i];
    // деление на количество
    if n > 0 then
        GetAverage := GetAverage / n;
end;

// МАССИВ <=> СТРОКА =====

//чтение отдельного массива из текстовой строки
//разделитель чисел ;
//в случае ошибки возвращает nil
function ReadArrayFromString(S: String): TArray;
var
    Code: Integer; //ошибка преобразования текста в число
    A: TArray; //временная переменная массива
    i: Integer;
    S1: String;
begin
    // создаем пустой массив
    SetLength(A, 0);
    //считываем данные в массив
    Code := 0; i := 1;
    //пока есть ; и нет ошибок, повторить
    while (pos(';', S) > 0) and (Code = 0) do
    begin
        SetLength(A, i);
        S1 := copy(S, 1, pos(';', S) - 1);
        if S1 <> '' then //пустые значения пропускаются
        {R-}
            Val(S1, A[i-1], Code);
        {R+}
        Delete(S, 1, pos(';', S));
        {// TEST 3
        Writeln('Значение ', i, ' (', S1, ') = ', A[i-1]);
        //}
        Inc(i);
    end;
    //последнее значение - до конца строки
    if Code = 0 then
    begin
        SetLength(A, i);
        if S <> '' then
        {R-}
            Val(S, A[i-1], Code);
        {R+}
        {// TEST 3
        Writeln('Значение ', i, ' (', S, ') = ', A[i-1]);
        //}
    end;

    //удалось ли считать массив ?
    if Code = 0 then
        ReadArrayFromString := A
    else
    begin
        Finalize(A);
        ReadArrayFromString := nil;
    end;
end;

```

```

    PrintError(ERRORMSG_CantReadArray);
end;
end;

//записать массив A в строку текста
//разделители чисел ;
function WriteArrayToString(A: TArray): String;
var
    S, S1: String;
    i: Integer;
begin
    S := '';
    //запись значений в цикле
    for i := 0 to Length(A) - 2 do
    begin
        Str(A[i]:0:2, S1);
        S := S + S1 + ',';
    end;
    //последнее значение без ;
    Str(A[Length(A) - 1]:0:2, S1);
    S := S + S1;
    //результат
    WriteArrayToString := S;
end;

// СОЗДАНИЕ МАССИВОВ =====

//ввод размера массива
//допустимый размер от 0 до 1000
//в противном случае возвращает -1
function ReadSize: Integer;
begin
    Write('Введите размер массива: ');
    ReadSize := ReadInt;
    //контроль ввода
    if (ReadSize < 0) or (ReadSize > 1000) then
    begin
        PrintError(ERRORMSG_WrongSize);
        ReadSize := -1;
    end;
end;

//генерация массива
function GenerateArray: TArray;
var
    i, n: Integer;
    Min, Max: TValue;
begin
    ClrScr;
    n := ReadSize;
    if n > 0 then
    begin
        Write('Введите минимальное значение для генерации: ');
        Min := ReadValue;
        Write('Введите максимальное значение для генерации: ');
        Max := ReadValue;
        SetLength(GenerateArray, n);
        for i := 0 to n-1 do
            GenerateArray[i] := Random * (Max - Min) + Min;
        Write('Массив успешно сгенерирован');
    end;
    PressAnyKey;
end;

```



```

//вывод одного массива на экран
procedure PrintArray(A: TArray);
var i, n: Integer;
begin
  n := Length(A);
  Writeln('Размер массива = ', n);
  Writeln('Среднее значение = ', GetAverage(A):0:2);
  Writeln('Значения: ');
  Writeln(WriteArrayToString(A));
  Writeln;
end;

//показать все массивы
procedure PrintAllArrays;
var j, m: Integer;
begin
  Writeln;
  j := 0; m := Length(AllArrays);
  repeat
    Writeln('Массив №', j+1);
    PrintArray(AllArrays[j]);
    Inc(j);
    if (j mod 3 = 0) and (j < m) then
      begin
        Writeln('Для продолжения вывода нажмите любую клавишу');
        Readkey;
      end;
  until j >= m;
end;

//показать массив (-ы)
procedure ShowArrays;
var j, m: Integer;
begin
  ClrScr;
  Writeln('                                ВЫВОД МАССИВОВ НА ЭКРАН');
  m := Length(AllArrays);
  if (m > 0) then
    begin
      Writeln('Всего массивов: ', m);
      Write('Введите номер массива для вывода (-1 - все): ');
      j := ReadInt;
      if (j < -1) or (j >= Length(AllArrays)) then //неверный индекс
        PrintError(ERRORMSG_WrongArrayIndex)
      else if j = -1 then //показать все
        PrintAllArrays
      else //показать один
        PrintArray(AllArrays[j-1]);
    end
  else
    Writeln('Массив исходных данных пуст');
  PressAnyKey;
end;

// ОЧИСТКА МАССИВОВ =====

//полная очистка массивов
procedure ClearAll;
var j: Integer;
    Cmd: String;
begin
  ClrScr;
  Writeln('Все данные будут удалены. Продолжить? (y - да, n - нет)');
  repeat

```

```

    Readln(Cmd);
    Cmd := LowerCase(Cmd);
until (Cmd = 'y') or (Cmd = 'n');
if (Cmd = 'y') then
begin
    for j := 0 to Length(AllArrays) - 1 do
        Finalize(AllArrays[j]);
    Finalize(AllArrays);
    Writeln('Данные успешно удалены');
end;
PressAnyKey;
end;

// ТЕКСТОВЫЕ ФАЙЛЫ =====

//считывание массивов данных из текстового файла
procedure ReadArraysFromTextFile;
var
    FileName: String; //имя файла
    F: Text; //текстовый файл
    S: String; //одна строка из файла
    j: Integer; //номер текущей строки в файле
    m: Integer; //сколько удалось считать массивов
    A: TArray; //текущий массив
begin
    ClrScr;
    Writeln('                          СЧИТАТЬ МАССИВЫ ИЗ ТЕКСТОВОГО ФАЙЛА');
    Writeln('Массивы будут добавлены к ранее созданным. ');
    Write('Введите имя файла: ');
    Readln(FileName);

    if FileExists(FileName) then
    begin
        Writeln('Выполняется чтение файла. Пожалуйста, подождите...');

        //открытие файла
        Assign(F, FileName);
        Reset(F);

        j := 0; m := 0;
        //считывание построчно
        while not Eof(F) do
        begin
            Inc(j);
            Readln(F, S);
            A := ReadArrayFromString(S);
            //если считывание удачное - добавить массив
            if A <> nil then
            begin
                SetLength(AllArrays, Length(AllArrays) + 1);
                AllArrays[Length(AllArrays) - 1] := A;
                Inc(m);
            end
            else //иначе - ошибка, но файл продолжает считываться
                Writeln(' (строка ', j, ') ');
        end;
        //закрытие файла
        Close(F);
        Writeln('Чтение файла завершено. Успешно считано ', m, ' массивов');
    end
    else
        PrintError(ERRORMSG_FileNotFound);
end;

```



```

    PressAnyKey;
end;

//запись массивов в текстовый файл
procedure WriteArraysToTextFile;
var
    FileName: String;//имя файла
    F: Text;
    Cmd, S: String;
    CanWrite: Boolean;
    j: Integer;
begin
    ClrScr;
    Writeln('                                СОХРАНИТЬ МАССИВЫ В ТЕКСТОВЫЙ ФАЙЛ');
    Writeln('Существующие файлы будут перезаписаны. ');
    Write('Введите имя файла: ');
    Readln(FileName);

    //если файл существует - запросить перезапись
    CanWrite := True;
    if FileExists(FileName) then
    begin
        Writeln('Файл существует. Перезаписать? (y - да, n - нет)');
        repeat
            Read(Cmd);
            Cmd := LowerCase(Cmd);
            until not ((Cmd = 'y') or (Cmd = 'n'));
            //нельзя перезаписать
            if Cmd = 'n' then
                CanWrite := False;
        end;

        //запись в файл разрешена
        if CanWrite then
        begin
            //открыть файл для чтения
            Assign(F, FileName);
            Rewrite(F);
            //запись в цикле
            for j := 0 to Length(AllArrays) - 1 do
            begin
                S := WriteArrayToString(AllArrays[j]);
                Writeln(F, S);
            end;
            //закрывать файл
            Close(F);
            Writeln('Запись файла завершена');
        end;

        PressAnyKey;
    end;

    // СПРАВКА =====
    procedure ShowHelp;
    begin
        ClrScr;
        Writeln('                                РАБОТА С МАССИВАМИ ДАННЫХ');
        Writeln;
        Writeln('    Программа предназначена для обработки больших массивов числовых
данных. ');
        Writeln('    Поддерживается: ');
        Writeln('    - ввод-вывод в текстовые файлы');
    end;

```



```

    end;
    PressAnyKey;
end;
//}

begin
Randomize;

{ // TEST 1
Writeln('Тест 1: Ввод целых и дробных чисел');
Writeln(ReadInt);
Writeln(ReadValue:0:2);
PressAnyKey;
//}
{ // TEST 2
Writeln('Тест 2: Ввод размера массива');
Writeln(ReadSize);
PressAnyKey;
//}
{ // TEST 3
RunTest3;
//}

RunMainMenu;
end.

```

## Тестовые примеры

Примечание. Приведенное оформление порядка тестирования не претендует на абсолютную правильность. Это только одна из возможных форм записи. На практике программы зачастую слишком сложны и велики, чтобы записывать порядок тестирования так подробно.

Данная программа является достаточно объемной, поэтому тестирование состоит из нескольких частей. Сначала выполняется тестирование отдельных процедур и функций, а затем работы программы в целом.

При тестировании подпрограммы вызываются из тела программы (закомментированная часть с меткой TEST).

Примечание. Процедуры и функции, взятые из ранее созданных программ, где они уже были протестированы, повторного тестирования не требуют.

### Тестирование функций ReadInt, ReadValue (метка TEST 1)

*Подготовка к тестированию:* раскомментировать TEST 1 в теле программы (убрать открывающую скобку {). После тестирования закомментировать обратно.

*Порядок тестирования:* ручной ввод входных величин.

№	Входные данные		Результат	Примечание
	ReadInt	ReadValue		
1	123q		Ошибка 1: Требуется ввести целое число в диапазоне - 32768..32767	Введен текст вместо числа

	57.1		Ошибка 1: ...	Введено дробное число вместо целого
			Ошибка 1: ...	Введена пустая строка
	12		12	
		my	Ошибка 2: Требуется ввести число	Введен текст вместо числа
		21.2	21.20	
2	-13		-13	Отрицательные числа не вызывают ошибок
		-52	-52.00	
3	33000		Ошибка 1: ...	Число выходит из диапазона Integer
	9876543210		Ошибка 1: ...	Число выходит из диапазона LongInt
	1		1	
		9876543210123456	9876543210123460.00	Число было округлено

### Тестирование функции ReadSize (метка TEST 2)

*Подготовка к тестированию:* раскомментировать TEST 2 в теле программы (убрать открывающую скобку {). После тестирования закомментировать обратно.

*Порядок тестирования:* ручной ввод входных величин.

№	Входные данные	Результат	Примечание
1	-1	Ошибка 3: Размер массива должен быть от 1 до 1000	Отрицательное число
	2000	Ошибка 3: ...	Слишком большое значение
	124	124	

### Тестирование функций ReadArrayFromString, WriteArrayToString, GetAverage (метка TEST 3)

*Подготовка к тестированию:* раскомментировать TEST 3 во всем тексте программы (выполнить автоматическую замену “{ TEST 3” на “{ TEST 3”). После тестирования закомментировать обратно.

*Порядок тестирования:* ручной ввод входных величин.

№	Входные данные	Результат	Примечание
1	1;2;3;4;5	1;2;3;4;5 Значение 1 (1) = 1.000000000000000E+000 Значение 2 (2) = 2.000000000000000E+000 Значение 3 (3) = 3.000000000000000E+000 Значение 4 (4) = 4.000000000000000E+000 Значение 5 (5) = 5.000000000000000E+000 Среднее значение = 3.000000000000000E+000 1.00;2.00;3.00;4.00;5.00	

2	1	Значение 1 (1) = 1.000000000000000E+000 Среднее значение = 1.000000000000000E+000 1.00	Массив с одним значением
3	1;2.3;B;4	Значение 1 (1) = 1.000000000000000E+000 Значение 2 (2.3) = 2.300000000000000E+000 Значение 3 (B) = 0.000000000000000E+000 Ошибка 5: Не удалось считать массив из строки	Если в строке встречается текст вместо числа, считывание прерывается
	1;2;;	Значение 1 (1) = 1.000000000000000E+000 Значение 2 (2) = 2.000000000000000E+000 Значение 3 ( ) = 0.000000000000000E+000 Значение 4 ( ) = 0.000000000000000E+000	Пустые значения считаются равными 0

## Тестирование работы программы

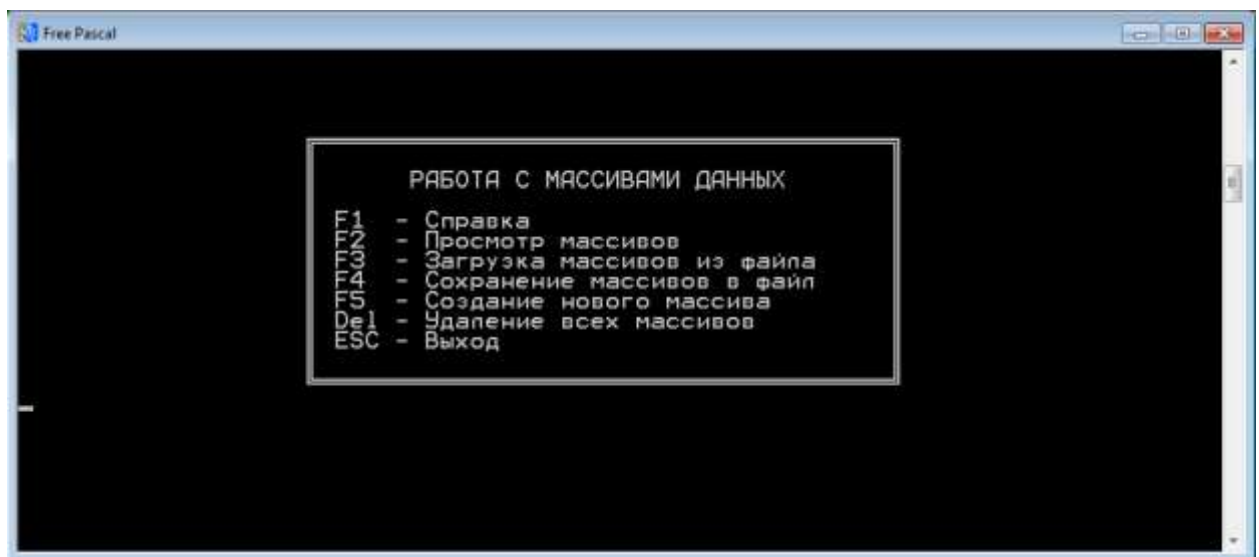
**Подготовка к тестированию:** Необходимо подготовить и сохранить в директорию с программой файл “1.csv”, содержащий достаточно большое количество массивов (не менее 100). Массивы содержатся в строках, разделитель чисел точка с запятой (;), дробная часть числа отделяется точкой (.). По крайней мере один массив должен содержать текст вместо числа.

**Порядок тестирования:**

### Шаг 1. Начало работы

*Действия пользователя:* Загрузить программу.

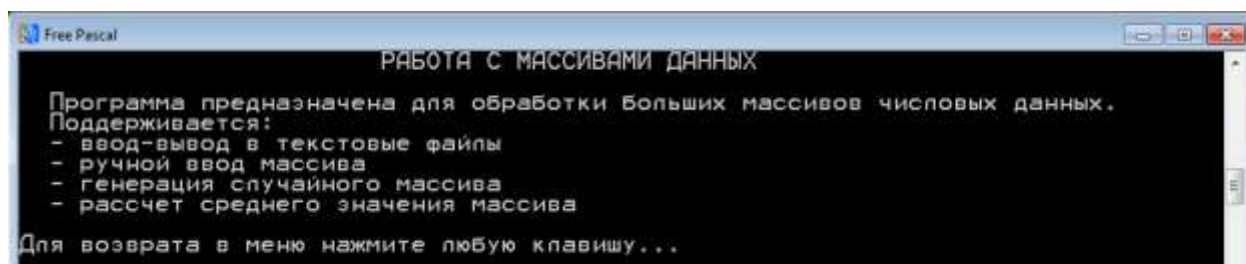
*Результат:* Отображается главное меню.



### Шаг 2. Просмотр справки

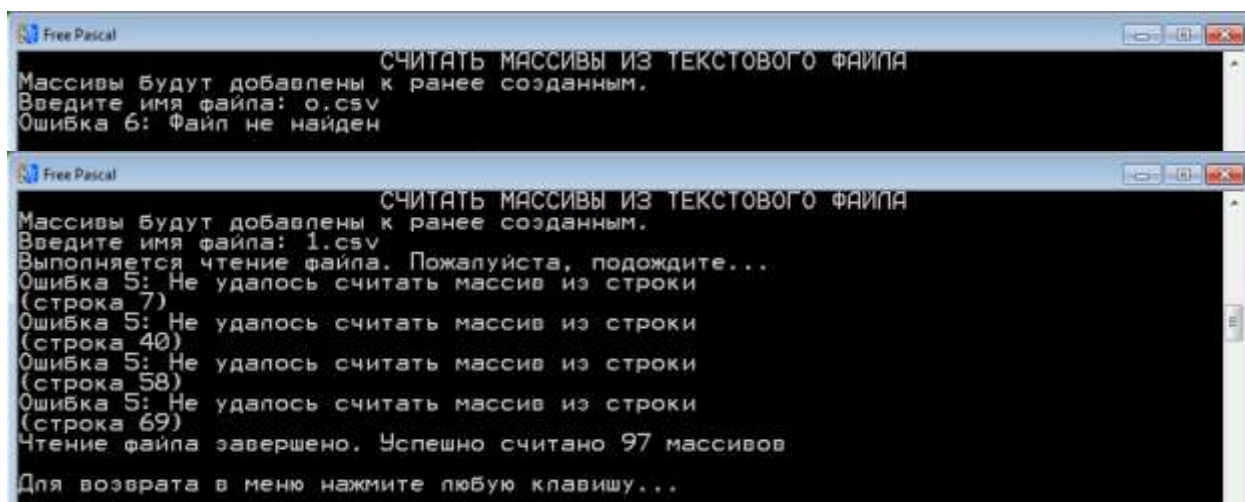
*Действия пользователя:* Нажать F1, просмотреть справку, нажать любую клавишу для возврата в главное меню.

*Результат:* Отображается справка.



### Шаг 3. Загрузка массивов из файла

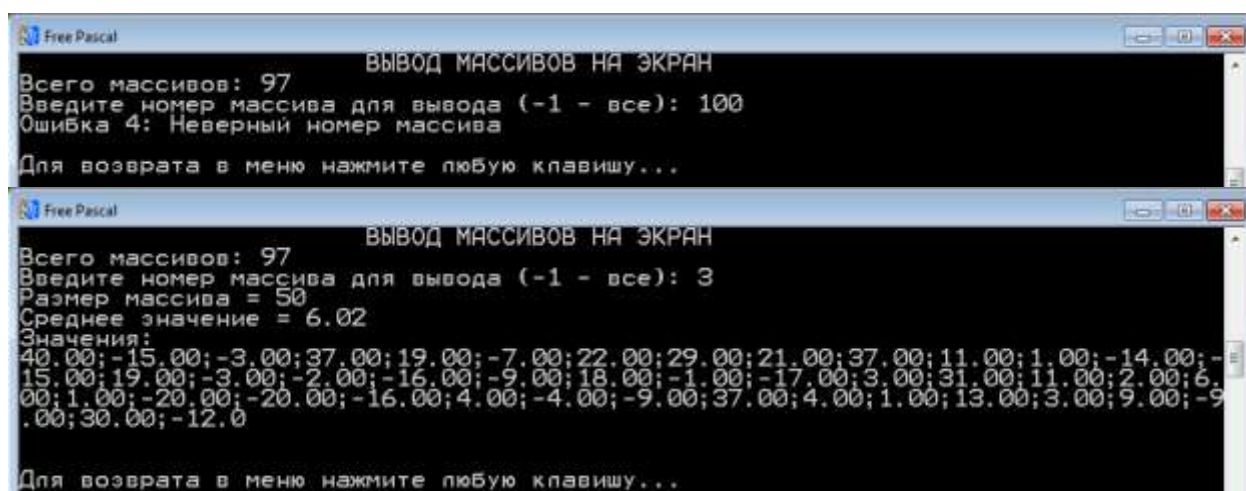
<i>Действия пользователя</i>	<i>Результат</i>
Нажать F3.	Программа перейдет в режим загрузки массивов из файла.
Ввести имя несуществующего файла	Программа выведет сообщение об ошибке. Программа предложит вернуться в главное меню.
Нажать любую клавишу.	Возврат в главное меню.
Нажать F3.	Программа перейдет в режим загрузки массивов из файла.
Ввести верное имя файла с тестовыми примерами	Начнется считывание файла. Когда считывание дойдет до массива с ошибками, программа выдаст сообщение с указанием номера строки, но считывание не прервется. После обработки файлов будет выведено число успешно считанных массивов. Программа предложит вернуться в главное меню.
Нажать любую клавишу для возврата в главное меню.	Возврат в главное меню.



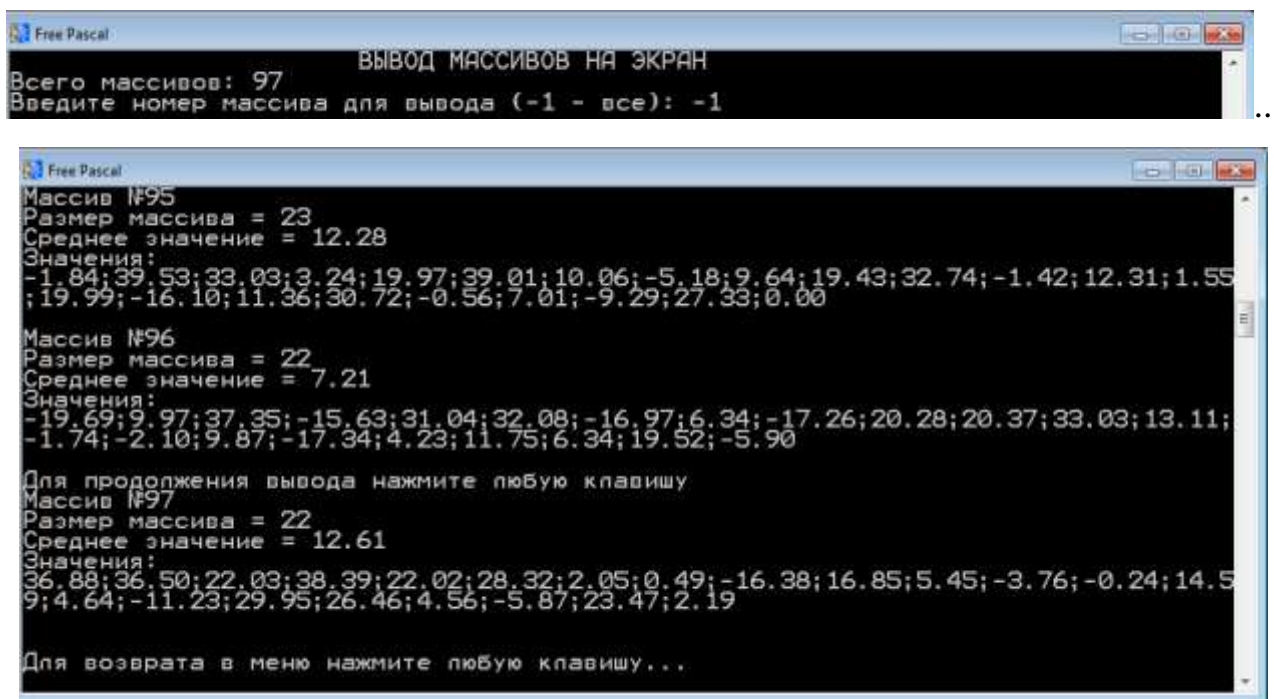
#### Шаг 4. Просмотр массивов

<i>Действия пользователя</i>	<i>Результат</i>
Нажать F2.	Программа перейдет в режим просмотра массивов. Будет выведено общее количество массивов. Программа предложит ввести номер массива для отображения.
Ввести номер массива, превосходящий их количество	Будет выведено сообщение об ошибке. Программа предложит вернуться в главное меню.
Нажать любую клавишу для возврата в главное меню.	Возврат в главное меню.
Нажать F2.	Программа перейдет в режим просмотра массивов. Будет выведено общее количество массивов. Программа предложит ввести номер массива для отображения.
Ввести верный номер массива, например 5.	Будет выведено: – размер массива – среднее значение массива – значения массива в строку Программа предложит вернуться в главное меню.
Нажать любую клавишу для возврата	Возврат в главное меню.

В главное меню.	
Нажать F2.	Программа перейдет в режим просмотра массивов. Будет выведено общее количество массивов. Программа предложит ввести номер массива для отображения.
Ввести -1.	Программа начнет выводить все массивы по 3. После каждого третьего массива программа предложит нажать любую клавишу для продолжения.
Нажимать любую клавишу, пока вывод массива не будет завершен.	Программа предложит вернуться в главное меню.
Нажать любую клавишу для возврата в главное меню.	Возврат в главное меню.



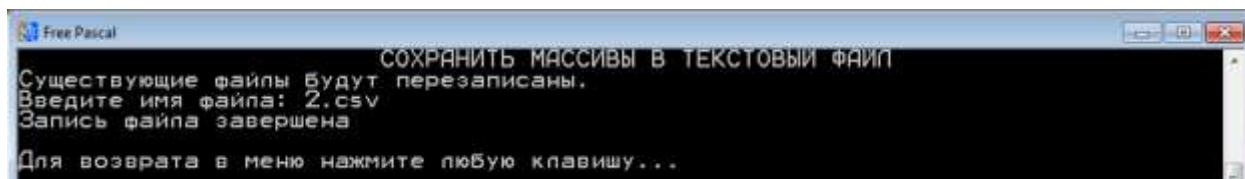




### Шаг 5. Сохранение массивов в файл

<i>Действия пользователя</i>	<i>Результат</i>
Нажать F4.	Программа перейдет в режим сохранения массивов в файл. Программа предложит имя файла для сохранения.
Ввести имя несуществующего файла, например 2.csv	Будет выполнено сохранение в указанный файл. Программа предложит вернуться в главное меню.
Нажать любую клавишу для возврата в главное меню. Проверить содержимое сохраненного файла через внешнюю программу.	Возврат в главное меню.
Нажать F4.	Программа перейдет в режим сохранения массивов в файл. Программа предложит имя файла для сохранения.
Ввести такое же имя не существующего файла.	Будет выведено сообщение о том, что файл существует и предложено перезаписать файл.
Ввести n.	Программа предложит вернуться в главное меню.
Нажать любую клавишу для возврата	Возврат в главное меню.

В главное меню.

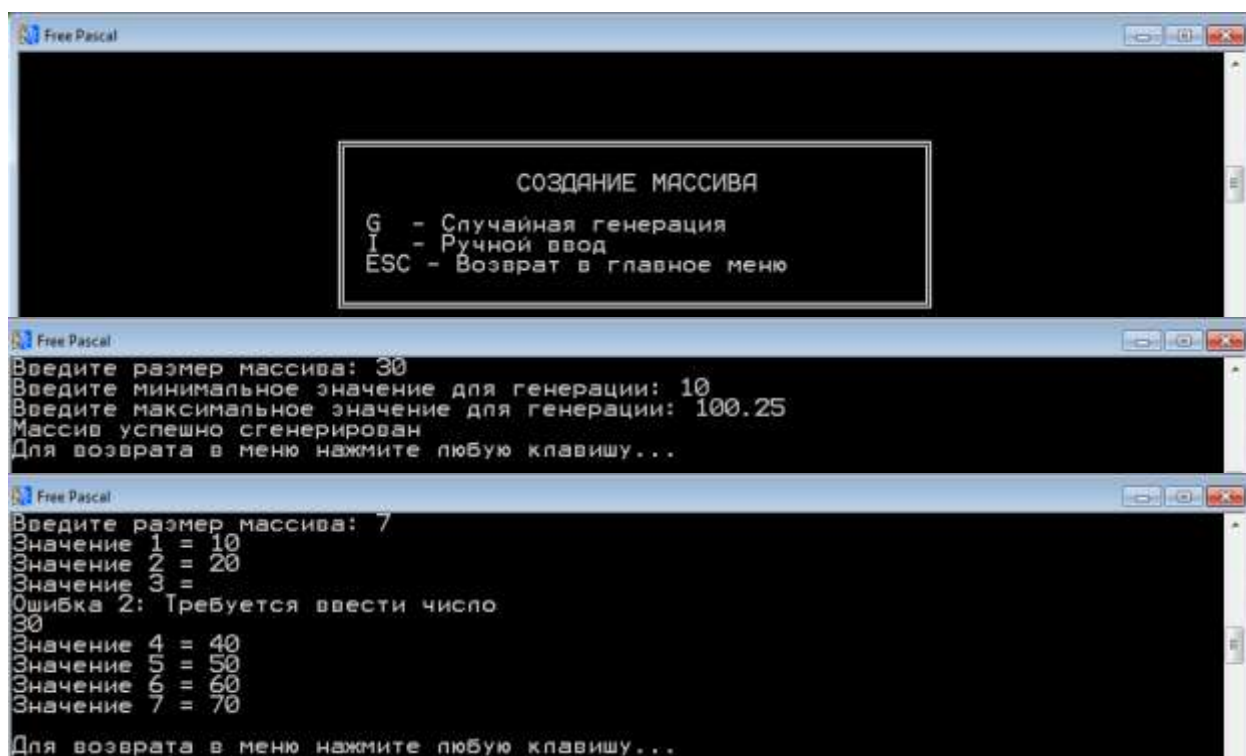


	A	B	C	D	E	F	G	H	I	J
1	36.00	-3.00	34.00	-6.00	26.00	36.00	8.00	5.00	10.00	37.00
2	-11.00	24.00	-19.00	-17.00	-17.00	-19.00	13.00	-3.00	32.00	-16.00
3	15.00	5.00	26.00	0.00	0.00	0.00	12.00	-2.00	28.00	8.00
4	40.00	-15.00	-3.00	37.00	19.00	-7.00	22.00	29.00	21.00	37.00
5	-2.00	22.00	6.00	14.00	1.00	-18.00	-1.00	27.00	-15.00	25.00
6	36.00	22.00	5.00	1.00	-10.00	40.00	34.00	21.00	8.00	6.00
7	11.00	7.00	-15.00	10.00	22.00	-18.00	-15.00	23.00	-10.00	2.00
8	36.00	5.00	29.00	14.00	2.00	-1.00	7.00	16.00	12.00	26.00
9	-7.00	4.00	32.00	11.00	37.00	23.00	40.00	12.00	25.00	-9.00
10	-13.00	26.00	17.00	23.00	35.00	37.00	32.00	39.00	37.00	28.00

## Шаг 6. Создание новых массивов

<i>Действия пользователя</i>	<i>Результат</i>
Нажать F5.	Программа перейдет в режим создания нового массива. Будет выведено подменю создания массива.
Нажать g в английской раскладке клавиатуры.	Программа перейдет в режим генерации массива. Пользователю будет предложено ввести размер массива.
Ввести допустимый размер массива, например 30.	Пользователю будет предложено ввести минимальное значение случайных чисел.
Ввести произвольное вещественное число, например 100.25.	Пользователю будет предложено ввести максимальное значение случайных чисел. Будет сгенерирован новый массив. Программа предложит вернуться в подменю.
Нажать любую клавишу.	Возврат в подменю создания массива.
Нажать i в английской раскладке клавиатуры.	Программа перейдет в режим создания массива путем ручного

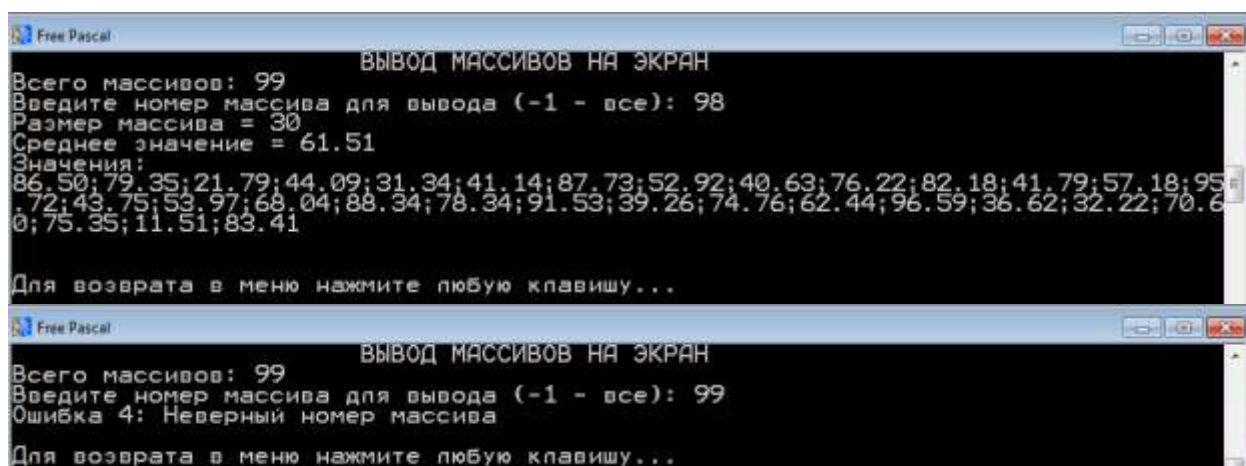
	<p>ввода.</p> <p>Пользователю будет предложено ввести размер массива.</p>
Ввести допустимый размер массива, например 7.	Пользователю будет предложено ввести значения массива.
<p>Последовательно ввести вещественные значения массива, например</p> <p>10</p> <p>20</p> <p>30</p> <p>40</p> <p>50</p> <p>60</p> <p>70</p> <p>В случае ошибочного ввода, повторить ввод.</p>	<p>Будет создан массив.</p> <p>Программа предложит вернуться в подменю.</p>
Нажать любую клавишу.	Программа вернется в подменю.
Дважды нажать Esc для возврата в главное меню.	Отобразится главное меню.



## Шаг 7. Просмотр созданных массивов

Действия пользователя	Результат
-----------------------	-----------

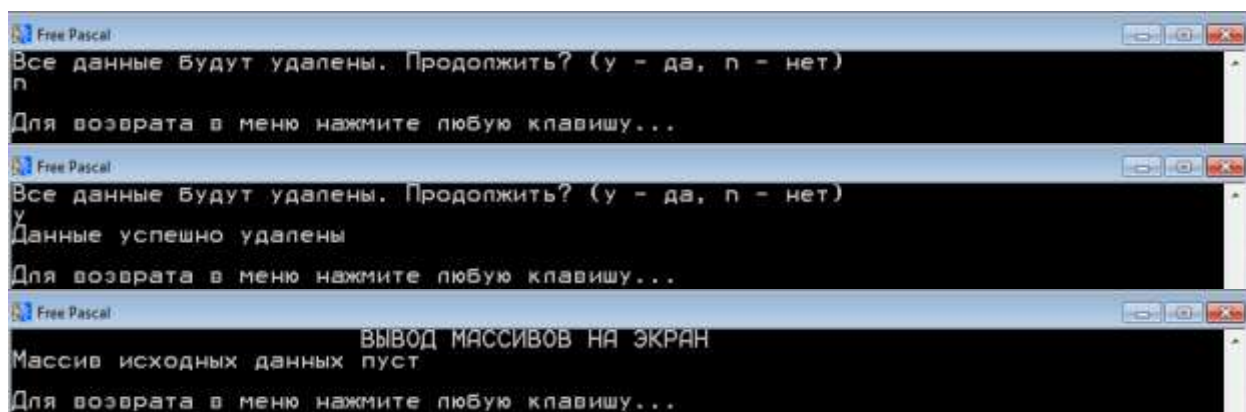
Нажать F2.	Программа перейдет в режим просмотра массивов. Будет выведено общее количество массивов. Программа предложит ввести номер массива для отображения.
Ввести номер сгенерированного на предыдущем шаге массива, в примере 98.	Будет выведено среднее значение и содержимое сгенерированного массива. Программа предложит вернуться в главное меню.
Нажать любую клавишу.	Возврат в главное меню.
Нажать F2.	Программа перейдет в режим просмотра массивов. Будет выведено общее количество массивов. Программа предложит ввести номер массива для отображения.
Ввести номер введенного на предыдущем шаге массива, в примере 99.	Будет выведено среднее значение и содержимое введенного массива. Программа предложит вернуться в главное меню.
Нажать любую клавишу.	Возврат в главное меню.



## Шаг 8. Удаление массивов

<i>Действия пользователя</i>	<i>Результат</i>
Нажать Del.	Программа перейдет в режим удаления массивов. Будет запрошено .

	Программа предложит ввести номер массива для отображения.
Ввести n.	Программа предложит вернуться в главное меню.
Нажать любую клавишу.	Возврат в главное меню.
Ввести у.	Все данные будут удалены. Программа предложит вернуться в главное меню.
Нажать F2 для проверки удаления.	Будет выведено сообщение о том, что массив данных пуст. Программа предложит вернуться в главное меню.
Нажать любую клавишу.	Возврат в главное меню.



### Шаг 9. Выход из программы

<i>Действия пользователя</i>	<i>Результат</i>
Нажать Esc.	Программа завершит работу.