

Краткие теоретические сведения.

<i>Диапазон констант (шестнадцатеричный)</i>	<i>Диапазон констант (десятичный)</i>	<i>Тип данных</i>
-\$8000000000000000..-\$80000001	-2 ⁶³ .. ²¹⁴⁷⁴⁸³⁶⁴⁹	Int64
-\$80000000..-\$8001	-2147483648.. ³²⁷⁶⁹	Integer
-\$8000..-\$81	-32768.. ¹²⁹	Smallint
-\$80.. ¹	-128.. ¹	Shortint

0..\$7F	0..127	0..127
\$80..\$FF	128..255	Byte
\$0100..\$7FFF	256..32767	0..32767
\$8000..\$FFFF	32768..65535	Word
\$10000..\$7FFFFFFF	65536..2147483647	0..2147483647
\$80000000..\$FFFFFFFF	2147483648..4294967295	Cardinal
\$100000000..\$7FFFFFFFFFFFFFFF	4294967296.. $2^{63}-1$	Int64

Примеры объявления *не типизированных констант*:

const

```

Min = 0;                // 0..127
Max = 100;              // 0..127
Center = (Max - Min) div 2; // 0..127
Beta = Chr(225);        // char
NumChars = Ord('Z') - Ord('A') + 1; // 0..127
Message = 'Out of memory'; // string
ErrStr = 'Error: ' + Message + ' '; // string
ErrPos = 80 - Length(ErrStr) div 2; // 0..127
Ln10 = 2.302585092994045684; // Extended
Ln10R = 1 / Ln10;       // Extended
Numeric = ['0'..'9'];   // Set of
Alpha = ['A'..'Z', 'a'..'z']; // Set of
AlphaNum = Alpha + Numeric; // Set of

```

Посмотреть тип данных константы можно в режиме отладки *run*→*inspect* (см. рис 1).

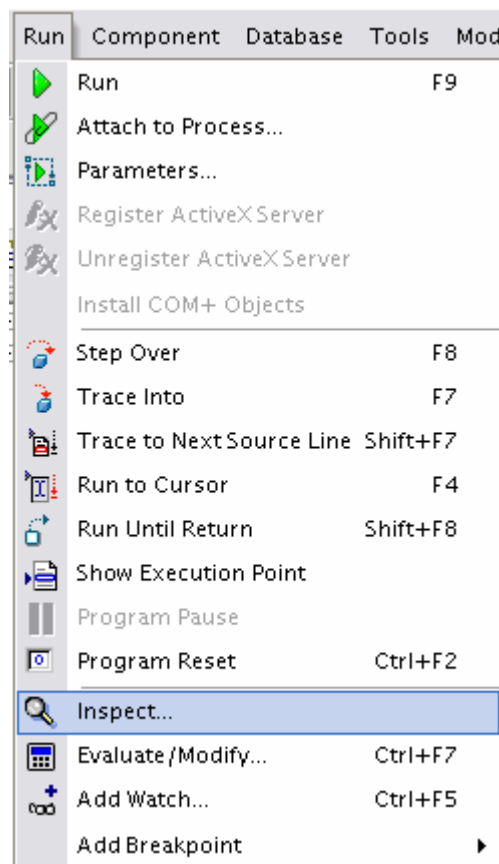
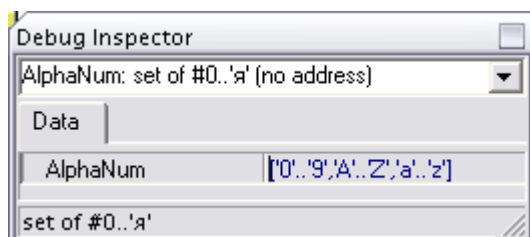
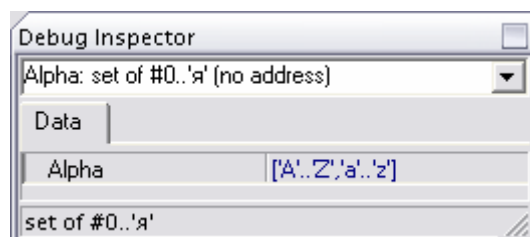
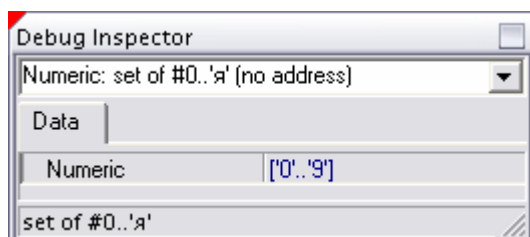
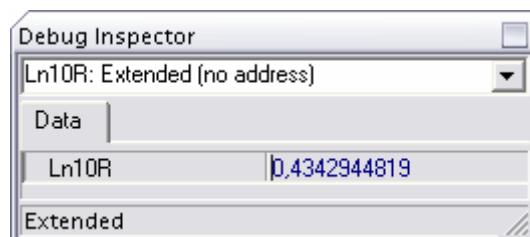
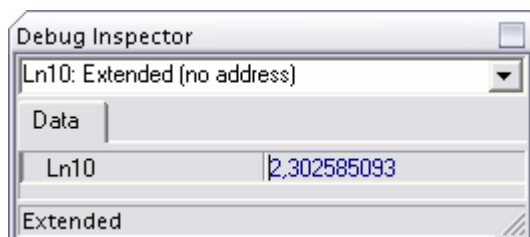
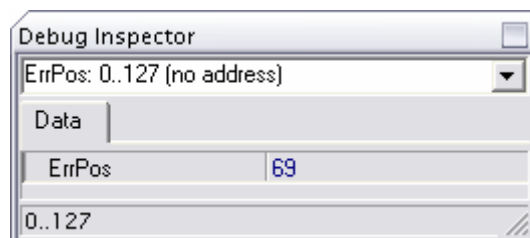
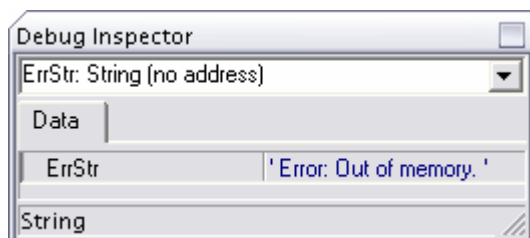
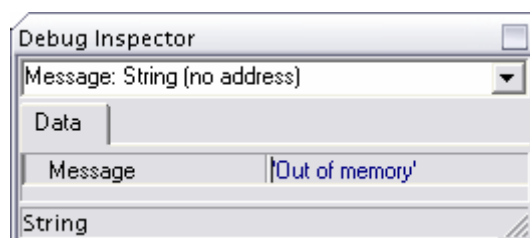
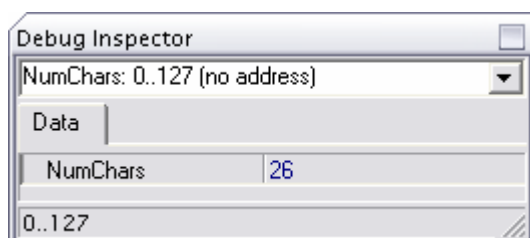
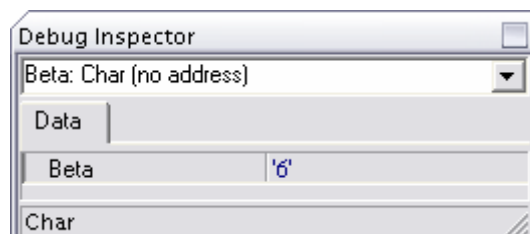
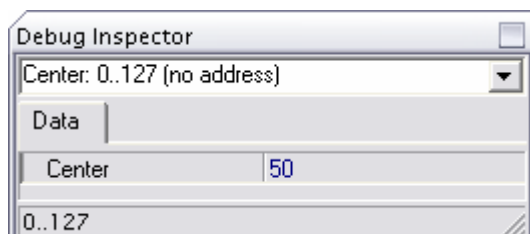
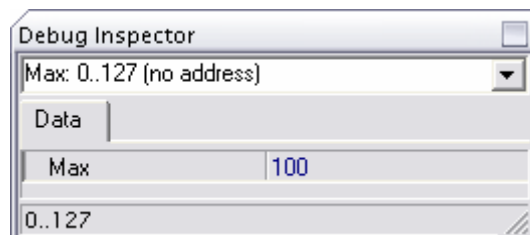
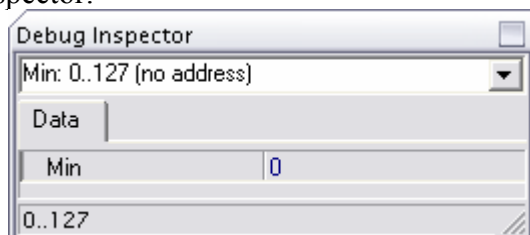


Рис 1 Вызов окна Debug inspector

Ниже представлены все константы из примера, просмотренные в окне Debug inspector:



Типизированные константы

Типизированные константы устанавливают значения массивов, записей, процедурных и указательных типов данных, они не могут участвовать в константных выражениях.

Формат описания типизированной константы:

const *identifier*: *type* = *value*

где,

- *identifier* – идентификатор;
- *type* – любой тип, кроме файлового и вариантного;
- *Value* – значение типа *type* (в большинстве случаев может быть константным выражением, но для массивов, записей, процедурных и указательных типов, определяются специальные правила объявления).

Примеры объявления типизированных констант:

Const

```
Max: Integer = 100;  
Ln10 Extended = 2.3025850929940456840179914546844;  
Sqrt2 Extended = 1.4142135623730950488016887242097;
```

Константные массивы

Объявление константного массива включает значения разделенные запятыми, заключенные в круглые скобки.

Пример:

```
const Digits: array[0..9] of Char = ('0', '1', '2', '3', '4', '5', '6', '7', '8', '9');
```

Объявлен типизированный константный массив названный Digits, значениями которого являются символы '0'..'9'.

Для работы с **null-terminated** строками (стандарт представления строк в языке C, требуется для вызовов процедур и функций операционной системы, вызовов процедур и функций расположенных в dll-библиотеках, и т.д.), строковая константа может быть инициализирована символьным массивом.

Пример:

```
const Digits: array[0..9] of Char = '0123456789';
```

При определении многомерных константных массивов, значения для каждого измерения отделяются круглыми скобками, и разделяются запятыми.

Пример:

```
type TCube = array [0..1, 0..1, 0..1] of Integer;  
const Maze: TCube = ( ( (0, 1), (2, 3) ), ( (4, 5), (6, 7) ) );
```

Объявлен новый тип данных TCube – (трехмерный массив, элементы имеют тип Integer), объявлен константный массив Maze, значения элементов массива представлены ниже, порядок присвоения значений элементам массива представлен на рис 2.

```
Maze[0,0,0] = 0  
Maze[0,0,1] = 1  
Maze[0,1,0] = 2  
Maze[0,1,1] = 3  
Maze[1,0,0] = 4  
Maze[1,0,1] = 5  
Maze[1,1,0] = 6
```

Maze[1,1,1] = 7

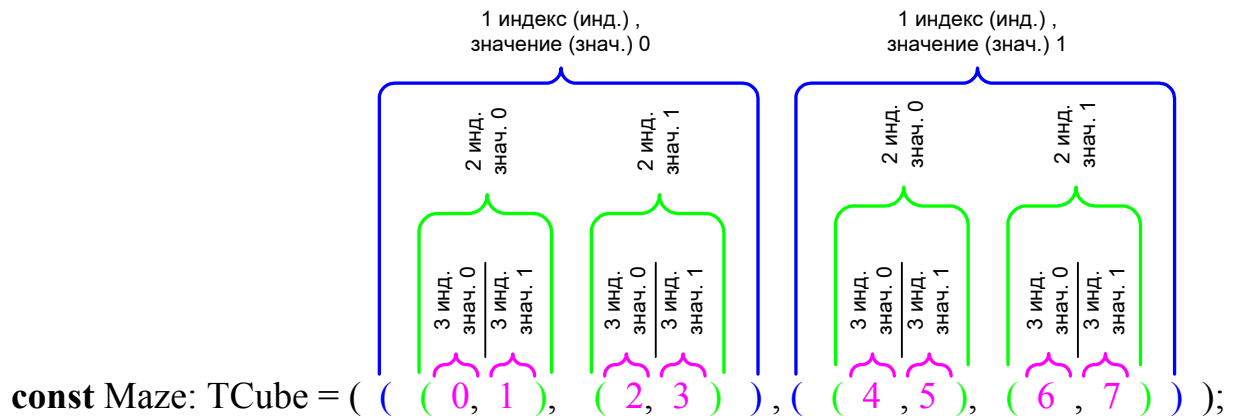


Рис 2 Распределение значений элементов массива, согласно индексам.

Процедуры и функции для работы со случайными числами.

Функция **Randomize**;

Модуль: System

Описание

Процедура инициализирует генерацию случайных чисел со случайного значения, используя текущее значение системного времени.

Если не инициализировать генератор случайных чисел с помощью данной процедуры, то при каждом запуске программы будет генерироваться одна и та же последовательность чисел. Для генерации определенной последовательности, необходимо системной переменной **RandSeed**: *LongInt* присвоить конкретное значение.

Пример

Var

i: Integer;

begin

...

Randomize;

for i:= 1 to 100 **do**

 Writeln(Random(200));

...

End.

Функция **Random** [(Range: Integer)];

Модуль: System

Описание

Функция генерирует случайное целое число в заданном диапазоне [0...Range).

Параметр Range является необязательным. Если данный параметр опущен, то функция возвращает число действительного типа в диапазоне [0...1).

Генератор случайных чисел инициализируется процедурой **Randomize**. Если не производить инициализацию, то при каждом запуске программы, функция **Random** будет возвращать одну и ту же последовательность чисел. Если необходимо сгенерировать определенную последовательность, то для этого нужно присвоить системной переменной **RandSeed**: *LongInt* конкретное значение. При этом следует помнить, что в разных версиях

компилятора используются различные алгоритмы генерирования последовательностей значений. По этому не рекомендуется использовать функцию **Random** для шифровки-дешифровки данных (т.е. когда необходимо генерировать одинаковые псевдослучайные последовательности чисел).

Пример

Var

X: Real;

Begin

...

Randomize;

X:= Random; $\{ 0 \leq X < 1 \}$

...

end;

Функция **RandG** (Mean, StdDev: *Extended*): *Extended*;

Модуль: Math

Описание

Функция генерирует случайное число с отклонением по Гауссу от среднего значения Mean. Возвращаемое значение будет лежать в диапазоне Mean-StdDev ... Mean+StdDev.

Пример

Uses

SysUtils,Math;

Var

 X: Real;

Begin

...

Randomize;

X:= RandG(100,10); $\{ 90 \leq X \leq 110 \}$

...

end;

Примеры решения заданий

Пример №1

Задание:

В массиве M(15) заменить нулевые элементы квадратами их индексов. (с использованием статического массива).

Решение:

program Zadanie1;

 {\$APPTYPE CONSOLE}

uses

 SysUtils;

Const

```

maxIndex = 15;
var
  i: Integer;
  m: array [1..maxIndex] of double;
begin
  // Инициализация элементов массива
  for i := 1 to maxIndex do
    begin
      Write('Enter m[' , i , ']:'); // Приглашение для ввода m[i]
      Readln(m[i]); // Ввод значения m[i] с консоли
    end;

  // Перебираем все элементы
  for i := 1 to maxIndex do
    // Если m[i]=0, то присваиваем значение квадрата индекса
    if m[i]=0 then m[i]:=i*i;

  // Выводим результат
  Writeln;
  Writeln('-----');
  Writeln('----- Result -----');
  Writeln;
  for i := 1 to maxIndex do
    Writeln('m[' , i , ']=' , m[i]:5:0);

  Writeln;
  Writeln('Press any key');
  Readln;
end.

```

```

C:\Program Files\Borland\Delphi7\Project
Enter m[1]:4
Enter m[2]:7
Enter m[3]:9
Enter m[4]:0
Enter m[5]:3.789
Enter m[6]:0
Enter m[7]:0
Enter m[8]:0
Enter m[9]:0
Enter m[10]:678.99
Enter m[11]:0
Enter m[12]:0
Enter m[13]:0
Enter m[14]:2
Enter m[15]:6

----- Result -----

m[1]=    4
m[2]=    7
m[3]=    9
m[4]=   16
m[5]=    4
m[6]=   36
m[7]=   49
m[8]=   64
m[9]=   81
m[10]= 679
m[11]=  121
m[12]=  144
m[13]=  169
m[14]=    2
m[15]=    6

Press any key

```

Рис 3 Результат решения примера №1.

Пример №2

Задание:

Задан массив $A(n)$. Вычислить сумму произведений всех пар соседних чисел $sum = \sum_{i=2}^n a_{i-1} * a_i$, (решить, используя динамические массивы).

Решение:

```
program Zadanie2;

{$APPTYPE CONSOLE}

uses
  SysUtils;

var
  i, n : Integer;
  sum : double;
  a: array of double;
begin
  while (n<=2) do
    Begin
      Write('Enter kolvo elementov >2 :');
      Readln(n);
      if n<3 then Writeln('Kolvo >2 !!!!!!!!!!!');
    end;
  // Инициализация массива
  SetLength(a,n);
  // Инициализация элементов массива
  for i := 0 to n-1 do
    begin
      Write('Enter a[' ,i ,']:'); // Приглашение для ввода a[i]
      Readln(a[i]);             // Ввод значения a[i] с консоли
    end;

  // Расчет суммы
  for i := 1 to n-1 do
    sum := sum + a[i-1]*a[i];    // Если a[i]=0, то присваиваем значение квадрата
  // индекса

  // Выводим результат
  Writeln;
  Writeln('-----');
  Writeln('----- Result -----');
  Writeln;
  Writeln(sum);

  Writeln;
  Writeln('Press enter');
  Readln;
end.
```

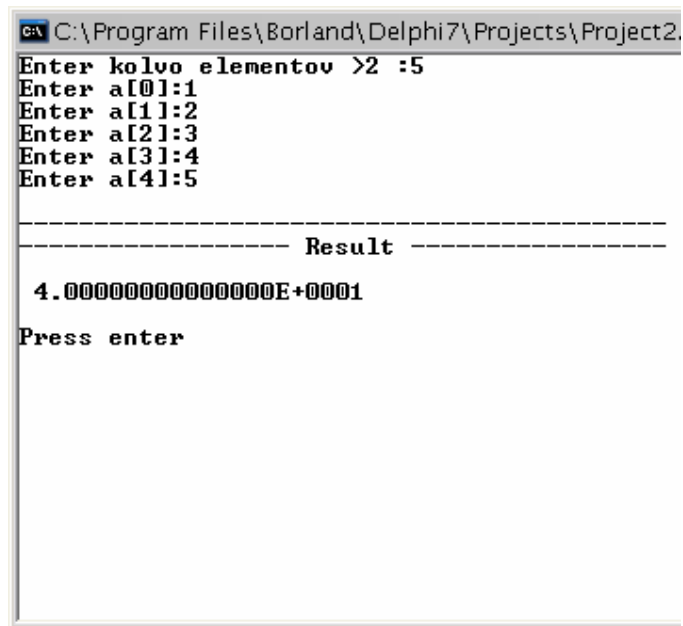



Рис 4 Результат решения примера №2

Пример №3

Задание:

Дан массив Y(10), вывести на экран все пары совпадающих элементов (решить с применением статических массивов).

Решение:

```

program Zadaniel;

{$APPTYPE CONSOLE}

uses
  SysUtils;
Const
  maxIndex = 10;
var
  i,j: Integer;
  y: array [1..maxIndex] of double;
begin
  // Инициализация элементов массива
  for i := 1 to maxIndex do
    begin
      Write('Enter y[' ,i, ']:'); // Приглашение для ввода y[i]
      Readln(y[i]);             // Ввод значения y[i] с консоли
    end;

  // Выводим результат
  Writeln;
  Writeln('-----');
  Writeln('----- Result -----');
  Writeln;

  // Перебираем все элементы - 1
  // т.к. последний элемент с самим собой сравнивать бесполезно
  for i := 1 to maxIndex-1 do
    for j := i+1 to maxIndex do // Ищем пару для, y[i], начиная с y[i+1]
      if y[i]=y[j] then

```

```

        Writeln('y[' , i , ']=y[' , j , ']= ' , y[i]); // Пара найдена, выводим её
Writeln;
Writeln('Press enter');
Readln;
end.

```

```

C:\Program Files\Borland\Delphi7\Projects
Enter y[1]:1
Enter y[2]:7
Enter y[3]:3
Enter y[4]:6
Enter y[5]:3
Enter y[6]:3
Enter y[7]:6
Enter y[8]:7
Enter y[9]:1
Enter y[10]:2

----- Result -----
y[1]=y[9]= 1.00000000000000E+0000
y[2]=y[8]= 7.00000000000000E+0000
y[3]=y[5]= 3.00000000000000E+0000
y[3]=y[6]= 3.00000000000000E+0000
y[4]=y[7]= 6.00000000000000E+0000
y[5]=y[6]= 3.00000000000000E+0000
Press enter
_

```

Рис 5 Результат решения примера №3

Пример №4

Задание:

Сформировать одномерный массив по следующему принципу: четные элементы равны квадрату индекса, а нечетные его обратной величине (решить с применением динамических массивов).

Решение:

```

program Zadanie4;

{$APPTYPE CONSOLE}

uses
  SysUtils;

var
  i, n : Integer;
  a: array of double;
begin
  while (n<=2) do
    Begin
      Write('Enter kolvo elementov >2 :');
      Readln(n);
      if n<3 then Writeln('Kolvo >2 !!!!!!!!!');
    end;
  // Инициализация массива
  SetLength(a,n);
  // Инициализация элементов массива
  for i := 0 to n-1 do

```

```

if (i mod 2)=1 then
    a[i]:= -i                // Для элементов с четным индексом
else
    a[i]:= i*i;              // Для элементов с не четным индексом

// Выводим результат
Writeln;
Writeln('-----');
Writeln('----- Result -----');
Writeln;
for i := 0 to n-1 do
    writeln('a[' , i , ']=' , a[i]:5:0);

Writeln;
Writeln('Press enter');
Readln;
end.

```

```

C:\Program Files\Borland\Delphi7\Projects\Project2.exe
Enter kolvo elementov >2 :20

-----
Result -----

a[0]=    0
a[1]=   -1
a[2]=    4
a[3]=   -3
a[4]=   16
a[5]=   -5
a[6]=   36
a[7]=   -7
a[8]=   64
a[9]=   -9
a[10]=  100
a[11]=  -11
a[12]=  144
a[13]=  -13
a[14]=  196
a[15]=  -15
a[16]=  256
a[17]=  -17
a[18]=  324
a[19]=  -19

Press enter

```

Рис 6 Результат решения примера №4

Пример №5

Задание:

Дан массив $Y(n)$, содержащий большое количество нулевых элементов. Заменить все группы подряд встречающихся нулей на элемент, состоящий из двух цифр, где первая цифра - 1, а вторая - количество нулей в группе (решить задание с использованием динамических массивов).

Решение:

Вариант №1

```

program Zadanie5;

{$APPTYPE CONSOLE}

```

```

uses
    SysUtils;

var
    i,j,n : Integer;
    kol : Integer;           // Количество элементов с учетом замены групп
    KolGroup: Integer;       // Количество элементов в группе
    ai : Integer;           // Индекс для массива a
    a,y: array of double;
begin
while (n<=2) do
    Begin
        Write('Enter kolvo elementov >2 :');
        Readln(n);
        if n<3 then Writeln('Kolvo >2 !!!!!!!!!!!');
        end;
    // Инициализация массивов
    SetLength(y,n);
    SetLength(a,n);
    // Инициализация элементов массива
    for i := 0 to n-1 do
        begin
            Write('Enter y['i,']:'); // Приглашение для ввода y[i]
            Readln(y[i]);           // Ввод значения y[i] с консоли
        end;

    i:=0;
    while (i<n) do
        begin
            if y[i]=0 then
                // Этот элемент нулевой, значит, возможно, начало группы
                begin
                    kolgroup:=1;           // На каждом шаге, изначально кол-во в группе 1
                    j:=i+1;               // начинаем со второго элемента
                    while ( (j<n) and (y[j]=0) ) do // Ищем группу
                        begin
                            inc(j);
                            inc(kolgroup); // увеличиваем кол-во элементов в группе
                        end;
                    if kolgroup>1 then
                        a[ai]:=StrToInt('1'+IntToStr(kolgroup))
                    else
                        a[ai]:=0;
                        i:=j;
                    end
                else
                    // Этот элемент не нулевой, значит, группа не началась
                    begin
                        a[ai]:=y[i];
                        inc(i);
                    end;
                inc(ai);
            end;

    // Изменяем, длины массивов - освобождаем память
    n:=ai;
    SetLength(a,n);
    SetLength(y,n);
    // Копируем массив a в y
    y := copy(a);
    // Выводим результат
    Writeln;
    Writeln('-----');
    Writeln('----- Result -----');

```

```

Writeln;

for i := 0 to n-1 do
    writeln('y[' , i, ']=' , y[i]:5:5);
Writeln;
Writeln('Press enter');
Readln;
end.

```

```

C:\Program Files\Borland\Delphi7\Projects\Project2.exe
Enter kolvo elementov >2 :15
Enter y[0]:0
Enter y[1]:0
Enter y[2]:0
Enter y[3]:1
Enter y[4]:1
Enter y[5]:1
Enter y[6]:0
Enter y[7]:0
Enter y[8]:0
Enter y[9]:1
Enter y[10]:1
Enter y[11]:0
Enter y[12]:0
Enter y[13]:0
Enter y[14]:0

----- Result -----
y[0]=13.000000
y[1]=1.000000
y[2]=1.000000
y[3]=1.000000
y[4]=13.000000
y[5]=1.000000
y[6]=1.000000
y[7]=14.000000
Press enter

```

Рис 7 Результат решения примера №5 (вариант №1)

Примечание: - Обратите внимание на функцию *Сору*

Анализ решения.

Данное решение является правильным, но у него есть «подводные камни» - двойное использование памяти. Ведь мы используем два массива одинаковой длины, и пока массивы малы наша программа будет работать корректно, но при увеличении длины может получиться, что для выполнения программы не хватит памяти. На ваших компьютерах 64 Мб ОЗУ, допустим, мы работаем с массивом в 6 000 000 элементов типа Double, для размещения такого массива потребуется $6\,000\,000 \times 8 = 48\,000\,000$ байт, на один массив хватит, а на второй нет! **Результат программа не работает.**

Вариант №2

```

program Zadanie5;

{$APPTYPE CONSOLE}

uses
    SysUtils;

var
    i, j, n : Integer;
    kol : Integer;           // Количество элементов с учетом замены групп
    KolGroup: Integer;       // Количество элементов в группе
    ai : Integer;           // Второй индекс для массива y
    y: array of double;

```

```

begin
while (n<=2) do
  Begin
    Write('Enter kolvo elementov >2 :');
    Readln(n);
    if n<3 then Writeln('Kolvo >2 !!!!!!!!!!!');
    end;
  // Инициализация массивов
  SetLength(y,n);
  // Инициализация элементов массива
  for i := 0 to n-1 do
    begin
      Write('Enter y['',i,'']:'); // Приглашение для ввода y[i]
      Readln(y[i]);              // Ввод значения y[i] с консоли
    end;

i:=0;
while (i<n) do
  begin
    if y[i]=0 then
      // Этот элемент нулевой, значит, возможно, начало группы
      begin
        kolgroup:=1;           // На каждом шаге, изначально кол-во в группе 1
        j:=i+1;                // начинаем со второго элемента
        while ( (j<n) and (y[j]=0) ) do // Ищем группу
          begin
            inc(j);
            inc(kolgroup);      // увеличиваем кол-во элементов в группе
          end;
        if kolgroup>1 then
          y[ai]:=StrToInt('1'+IntToStr(kolgroup))
        else
          y[ai]:=0;
          i:=j;
        end
      else
        // Этот элемент не нулевой, значит, группа не началась
        begin
          y[ai]:=y[i];
          inc(i);
        end;
        inc(ai);
      end;
  // Изменяем длины массивов - освобождаем память
  n:=ai;
  SetLength(y,n);
  // Выводим результат
  Writeln;
  Writeln('-----');
  Writeln('----- Result -----');
  Writeln;

  for i := 0 to n-1 do
    writeln('y['',i,'']='',y[i]:5:5);
  Writeln;
  Writeln('Press enter');
  Readln;
end.

```

```
C:\Program Files\Borland\Delphi7\Projects\Project2.  
Enter kolvo elementov >2 :20  
Enter y[0]:0  
Enter y[1]:0  
Enter y[2]:0  
Enter y[3]:0  
Enter y[4]:0  
Enter y[5]:1  
Enter y[6]:1  
Enter y[7]:0  
Enter y[8]:0  
Enter y[9]:1  
Enter y[10]:0  
Enter y[11]:0  
Enter y[12]:1  
Enter y[13]:0  
Enter y[14]:1  
Enter y[15]:0  
Enter y[16]:1  
Enter y[17]:0  
Enter y[18]:1  
Enter y[19]:0  
  
----- Result -----  
  
y[0]=15.00000  
y[1]=1.00000  
y[2]=1.00000  
y[3]=12.00000  
y[4]=1.00000  
y[5]=12.00000  
y[6]=1.00000  
y[7]=0.00000  
y[8]=1.00000  
y[9]=0.00000  
y[10]=1.00000  
y[11]=0.00000  
y[12]=1.00000  
y[13]=0.00000  
  
Press enter
```

Рис 8 Результат решения примера №5 (вариант №2)

Пример №6

Задание:

Заполнить динамический одномерный массив случайными числами в диапазоне [3,7)

Решение:

```
program Zadanie6;  
  
{$APPTYPE CONSOLE}  
  
uses  
    SysUtils;  
Const  
    RMin = 3;  
    RMax = 7;  
var  
    i,n : Integer;  
    a: array of double;  
begin  
    while (n<=2) do  
        Begin  
            Write('Enter kolvo elementov >2 :');
```

```

Readln(n);
if n<3 then Writeln('Kolvo >2 !!!!!!!!!');
end;
// Инициализация массива
SetLength(a,n);
// Инициализация генератора случайных чисел
Randomize;
// Инициализация элементов массива
for i := 0 to n-1 do
  a[i]:=RMin+Random*(RMax-RMin);

// Выводим результат
Writeln;
Writeln('-----');
Writeln('----- Result -----');
Writeln;

for i := 0 to n-1 do
  writeln('a[' ,i, ']=' ,a[i]);
Writeln;
Writeln('Press enter');
Readln;
end.

```

```

C:\Program Files\Borland\Delphi7\Projects\Project2.
Enter kolvo elementov >2 :20

-----
----- Result -----

a[0]= 5.20080131944269E+0000
a[1]= 6.07936117425561E+0000
a[2]= 4.78093444276601E+0000
a[3]= 6.42349107004702E+0000
a[4]= 3.26382719259709E+0000
a[5]= 3.37378054857254E+0000
a[6]= 4.31745010707527E+0000
a[7]= 4.26800710521638E+0000
a[8]= 3.49531468655914E+0000
a[9]= 6.57184835895896E+0000
a[10]= 5.49140961188823E+0000
a[11]= 6.95825064368546E+0000
a[12]= 3.56048120092601E+0000
a[13]= 4.52601879090071E+0000
a[14]= 6.19691982213408E+0000
a[15]= 3.12393569760025E+0000
a[16]= 5.40379523392767E+0000
a[17]= 3.93812652304769E+0000
a[18]= 3.84061506856233E+0000
a[19]= 6.28553860820830E+0000

Press enter

```

Рис 9 Результат решения примера №6

Пример №7

Задание:

Дано два массива:

A = (1, 7, 98 , 3 ,7 ,8 ,3 ,12 ,435);

B = (1.56 ,3.7 ,9.8 ,3.67 ,67 ,8.78 ,3 , 45.12, 4.67);

Заполнить массив C значениями $C[i]=A[i]*B[i]$

Решение:

```

program Zadanie7;

```



```

{$APPTYPE CONSOLE}

uses
  SysUtils;
Const
  MaxIndex = 9;
Type
  mas = array [1..MaxIndex] of double;
Const
  a : mas = (1, 7, 98, 3, 7, 8, 3, 12, 435);
  b : mas = (1.56, 3.7, 9.8, 3.67, 67, 8.78, 3, 45.12, 4.67);
var
  i : Integer;
  c : mas;
begin
  // Заполнения элементов массива C
  for i := 1 to MaxIndex do
    c[i] := a[i] * b[i];

  // Выводим результат
  Writeln;
  Writeln('-----');
  Writeln('----- Result -----');
  Writeln;

  for i := 0 to MaxIndex do
    writeln('c[' , i, ']=', c[i]);
  Writeln;
  Writeln('Press enter');
  Readln;
end.

```

```

C:\Program Files\Borland\Delphi7\Project
-----
----- Result -----
c[0]= 4.33934892348492E-0317
c[1]= 1.56000000000000E+0000
c[2]= 2.59000000000000E+0001
c[3]= 9.60400000000000E+0002
c[4]= 1.10100000000000E+0001
c[5]= 4.69000000000000E+0002
c[6]= 7.02400000000000E+0001
c[7]= 9.00000000000000E+0000
c[8]= 5.41440000000000E+0002
c[9]= 2.03145000000000E+0003
Press enter

```

Рис 10 Результат решения примера №7

Задания

Вариант №1

Задание №1

Дан целочисленный вектор $T(m)$. Изменить знак всех элементов с четными индексами на противоположный, а с нечетными заменить на случайное число из диапазона $[1,15)$ (решить с использованием динамических массивов).

Задание №2

Заполнить элементы массива $C(50)$ используя следующую зависимость:

Для четных элементов $C[i] = \sum a[i]^{b[i]}$ и $C[i] = \sum \frac{a[i]}{b[i]}$ для нечетных, где

A – константный массив, $A = (23, 56.45, 24.12, 73.36, 29.19, 8.09, 21.61, 72.91, 89.31, 53.83)$,
 B – массив заполненный случайными числами.

Задание №3

Дан массив $X(k)$, содержащий большое количество нулевых элементов. Заменить группы элементов, состоящие из нечетного количества нулей, на один нулевой элемент, а из четного - на два.

Вариант №2

Задание №1

Массив $K(30)$ заполнен случайными числами от -15 до 15. Определить количество отрицательных элементов и их индексы (решить с использованием статических массивов).

Задание №2

Заполнить элементы массива C используя следующую зависимость:

$C[i] = \sum a[i] * b[i]$, где

A – константный массив, $A = (23, 56.45, 24.12, 73.36, 29.19, 8.09, 21.61, 72.91, 89.31, 53.83)$,
 B – динамический массив значения вводятся с консоли.

Задание №3

Задан вектор $A(10)$. Определить, сколько содержится в нем различных чисел.

Вариант №3

Задание №1

Заполнить одномерный массив $B(25)$ так, чтобы каждый элемент с четным индексом был равен половине своего номера, а каждый элемент с нечетным индексом – случайному числу из диапазона $[300,567)$ (решить с использованием статических массивов).

Задание №2

Задан вектор:
 $M=(16,34,67,2345,567,897,34,2,6,89,08,57,345,87,46,989,67,456,535,679,346,657,89,57,35)$.
Определить произведение нечетных элементов, имеющих четные индексы.

Задание №3

Дан целочисленный вектор $M(15)$. Определить число соседств из двух чисел разного знака.

Вариант №4

Задание №1

Заполнить одномерный массив $M(n)$ так, чтобы каждый третий элемент был равен сумме двух предыдущих, а остальные случайному числу из диапазона $[34.67, 103405)$ (решить с использованием динамических массивов).

Задание №2

Задан вектор:
 $M=(10.89, 38.32, 78.11, 96.28, 21.73, 26.99, 63.56, 81.67, 76.54, 49.35, 24.66, 40.53, 58.18, 27.85, 63.78, 29.30, 53.55, 33.72, 67.87, 56.36)$.
Определить сумму произведений элементов, у которых индексы кратны 1,2,3,4,5.

Задание №3

Определить в данном векторе $P(m)$ количество пар соседних чисел, являющихся противоположными.

Задание повышенной сложности

Заданы два одномерных массива $A(15)$ и $B(15)$. Сформировать массив $C(30)$, содержащий элементы обоих массивов, расположенные в порядке возрастания.

Задания для самостоятельной подготовки

Задание №1

В массиве $A(20)$ поменять местами соседние четные и нечетные по номеру элементы. Дополнительные массивы не использовать.

Задание №2

Дан одномерный числовой массив $T(k)$. Вычислить сумму произведений всех троек соседних чисел.

Задание №3

Дан одномерный массив $H(n)$. Определить индексы всех равных элементов.