

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к лабораторным работам

по программированию на языке Паскаль

часть 2

(для студентов специальности 6.050102
“Компьютерная инженерия”
дневной формы обучения)

Рассмотрено
на заседании кафедры КИ
Протокол № 1
от 31 августа 2010 г.

Утверждено
на заседании учебно-
издательского совета Дон-
НТУ
Протокол № 4
от 07 октября 2010 г.

УДК 681.3(07)

Методические указания к лабораторным работам по программированию на языке Паскаль, часть 2 (для студентов специальности 6.050102 “Компьютерная инженерия” дневной формы обучения). Составители: В.И.Назаренко, К.Б.Юсупова, О.Ю.Чередникова. – Донецк: ДонНТУ, 2010. - 71 с.

Приведены методические указания к двум работам, которые входят в цикл лабораторных работ по программированию на языке Паскаль. В лабораторной работе № 4 рассматриваются методы разработки алгоритмов и программ для задач обработки многомерных массивов, в работе № 5 - методы организации процедур и функций, а также способы их использования при обработке одномерных массивов с произвольными именами типов. Для каждой лабораторной работы приведено 100 вариантов заданий.

Составители: доц. Назаренко В.И.,
 асс. Юсупова К.Б.,
 асс. Чередникова О.Ю.

Ответственный
за выпуск проф. Святный В.А.

Рецензент доц. Федяев О.И.

Лабораторная работа № 4

ОБРАБОТКА МАТРИЦ

Методические указания

Цель работы – практически освоить методы разработки алгоритмов и методы программной реализации задач обработки многомерных массивов.

В любом из вариантов заданий к лабораторной работе № 4 необходимо составить программу, которая предусматривает:

- ввод из текстового файла элементов матрицы;
- вывод введенных данных на принтер и (или) экран;
- обработку матрицы в соответствии с условием задачи;
- вывод результатов.

Вывод результатов нужно организовать как на экран дисплея, так и на принтер. Для организации вывода рекомендуется использовать процедуры ScreenMatrix и PrinterMatrix, приведенные в примере выполнения лабораторной работы.

Ввод матрицы из текстового файла, в отличие от одномерного массива, нельзя организовать по признаку конца файла (по значению функции SeekEof). При вводе с использованием указанного признака матрицы $m \times n$ элементов программа может определить лишь общее количество элементов матрицы, но она не в состоянии отдельно указать значения параметров m и n . Поэтому рекомендуется при формировании текстового файла в первой его строке указать значения m и n (для квадратной матрицы – только значение n), а в других строках – значения элементов матрицы. После ввода из файла переменных m и n ввод элементов матрицы выполняется в цикле **For**, как это сделано в примере.

Обрабатываемая в задании матрица должна иметь от 5 до 10 строк и столбцов.

Общее замечание. При решении ряда задач следует обращать внимание на возможное несоответствие типов элементов массива и выходных результатов. Например, среднее арифметическое значение, среднее квадратичное (см.п.1), дисперсия (см.п.27), другие виды средних значений, как правило, всегда вещественные вне зависимости от того, являются ли целочисленными или вещественными элементы обрабатываемого массива.

Отчет по лабораторной работе №4

В отчете по лабораторной работе необходимо привести:

- титульный лист;
- номер варианта и условие задачи;
- краткое описание программы;
- контрольный просчет (если результаты работы программы не являются очевидными);
- текст программы;
- результаты работы программы.

Печать текста программы рекомендуется выполнять с помощью редактора языка Турбо Паскаль. Если же для этого используется система Word, то печать должна производиться шрифтом Courier New. В этом случае все выводимые символы имеют одинаковую ширину, чем обеспечивается структурированность печатаемой программы. Для других разделов отчета целесообразно использовать шрифт Times New Roman.

Пример выполнения задания

Условие задачи. В заданной квадратной матрице с вещественными элементами сдвинуть каждую ее строку циклически влево или вправо таким образом, чтобы макси-

мальный элемент строки был расположен на главной диагонали. Вычислить след матрицы до и после ее преобразования.

Циклический сдвиг строки предполагает такую перестановку ее элементов, при которой не нарушается их начальное относительное расположение.

Пусть, например, третья строка матрицы имеет вид

5.8 -4.4 10.0 8.7 14.6 11.9

Здесь максимальный элемент $a_{3,5}=14.6$ расположен правее на две позиции от элемента главной диагонали $a_{3,3}=10.0$. Следовательно, необходимо дважды выполнить циклический сдвиг строки влево, после чего она примет вид

10.0 8.7 14.6 11.9 5.8 -4.4

След матрицы – это сумма элементов ее главной диагонали.

Примечание. Главную, как и побочную диагональ может иметь только квадратная матрица.

Задача, которая сформулирована в качестве примера выполнения задания, реализована в приведенной ниже программе Labor4.

При обработке матрицы в каждой ее строке определяется максимальный элемент A_{max} и его порядковый номер (индекс) j_{max} , после чего вычисляется параметр $k=i-j_{max}$, где i – номер строки матрицы. Этот параметр указывает на положение элемента A_{max} относительно главной диагонали матрицы. Если $k > 0$ (максимальный элемент находится слева от элемента главной диагонали), то выполняется циклический сдвиг строки вправо до тех пор, пока параметр k не примет нулевое значение (в каждом цикле значение k уменьшается на 1). Последнее означает, что максимальный элемент переставлен на место элемента главной диагонали. При $k < 0$ сдвиг выполняется влево.

В программе Labor4 предусмотрено два направления вывода данных: на экран дисплея и на принтер. Необходимость использования принтера при очередном запуске программы определяется ответом пользователя на запрос программы.

Процедура ScreenMatrix, которая используется для вывода матрицы на экран дисплея, предполагает, что для размещения матрицы достаточно одной страницы экрана. Если это не так, то эта процедура должна быть дополнена фрагментом, который обеспечивает управление “перелистыванием” страниц экрана. Пример такого дополнения для вывода одномерных массивов приведен в методических указаниях к лабораторной работе №5.

Процедура WaitEscape, которая содержится в программе Labor4, используется для приостановки работы программы до тех пор, пока не будет нажата клавиша Esc (эта клавиша имеет код 27).

Переменная k может изменяться лишь в диапазоне $-N_{max} .. N_{max}$. Так как в программе объявлено $N_{max} = 30$, то для размещения переменной k достаточно отвести один байт памяти. Этому соответствуют целочисленные типы `byte` и `shortint`. Поскольку значение k может быть как положительным, так и отрицательным, то для этой переменной в программе определен тип `shortint`.

```
Program Labor4;  
Uses Crt,Printer;  
Const Nmax = 30;  
Type  
    Matrix = array[1..Nmax,1..Nmax] of real;  
Var  
    i,j,           { параметры цикла }  
    n,             { размер матрицы }
```

```

jmax : byte; { позиция макс.элемента Amax в строке }
k : shortint; { разность между позициями Amax }
           { и элемента a[i,i] }
IndPrinter { индикатор использования принтера }
           : boolean;
ch : char; { символ нажатой клавиши }
Amax, { максимальный элемент в строке }
Buf, { буферная переменная }
Trace : real; { след матрицы }
A : Matrix; { обрабатываемая матрица }
F : text; { исходный файл }
{ ----- }
Procedure WaitEscape;
{ Приостановка программы до нажатия клавиши Esc }
Var ch : char;
Begin
  Repeat
    ch:=ReadKey;
  Until ord(ch)=27;
End { WaitEscape };
{ ----- }
Procedure ReadMatrix;
{ Чтение матрицы из текстового файла }
Var i,j : byte;
Begin
  Reset(F);
  Read(F,n);
  For i:=1 to n do
    For j:=1 to n do
      Read(F,a[i,j]);
  Close(F);
End { ReadMatrix };
{ ----- }
Procedure ScreenMatrix;
{ Вывод матрицы на экран дисплея }
Var i,j,k : byte;
Begin
  For i:=1 to n do
    Begin
      k:=0;
      For j:=1 to n do
        Begin
          Inc(k);
          If k<5 then
            Write(a[i,j]:8:2, '  ')
          Else
            Begin
              k:=0;
              Writeln(a[i,j]:8:2)
            End
        End
      End;
    If k>0 then Writeln;
  End;

```

```

End { ScreenMatrix };
{ ----- }
Procedure PrinterMatrix;
{ Вывод матрицы на принтер }
Var i,j,k : byte;
Begin
  For i:=1 to n do
    Begin
      k:=0;
      For j:=1 to n do
        Begin
          Inc(k);
          If k<5 then
            Write(Lst,a[i,j]:8:2,'  ')
          Else
            Begin
              k:=0;
              Writeln(Lst,a[i,j]:8:2)
            End
          End;
          If k>0 then Writeln(Lst);
        End;
      End { PrinterMatrix };
{ ----- }
Procedure TraceMatrix;
{ Вычисление следа матрицы }
Var i : byte;
Begin
  Trace:=0;
  For i:=1 to n do
    Trace:=Trace+a[i,i];
End { TraceMatrix };
{ ----- }

Begin

{ Установление соответствия между внутренним }
{ и внешним файлами }
Assign(F,'Matrix.dat');

{ Запрос об использовании принтера }
ClrScr;
Writeln('Будет ли использоваться принтер (Да,Нет) ?');
ch:=ReadKey;
If ch in ['Д','д','L','l'] then
  IndPrinter:=true
Else
  IndPrinter:=false;

{ Ввод и печать исходных данных }
ReadMatrix;
Writeln('          Исходная матрица   n = ',n);
ScreenMatrix;

```

```

If IndPrinter then
  Begin
    Writeln(Lst, '                Исходная матрица   n = ', n);
    PrinterMatrix;
  End;

{ Вычисление следа исходной матрицы }
TraceMatrix;
Writeln('След исходной матрицы Trace = ', Trace:8:2);
If IndPrinter then
  Writeln(Lst, 'След исходной матрицы Trace = ', Trace:8:2);
WaitEscape;

{ Преобразование исходной матрицы }
ClrScr;
For i:=1 to n do
  Begin
    Amax:=a[i,1]; jmax:=1;
    For j:=2 to n do
      If a[i,j]>Amax then
        Begin
          Amax:=a[i,j]; jmax:=j
        End;
    k:=i-jmax;
    If k>0 then                { ЦИКЛИЧЕСКИЙ СДВИГ }
      While k>0 do            { строки вправо }
        Begin
          Buf:=a[i,n];
          For j:=n downto 2 do
            a[i,j]:=a[i,j-1];
          a[i,1]:=Buf;
          Dec(k);
        End
      Else
        If k<0 then            { ЦИКЛИЧЕСКИЙ СДВИГ }
          While k<0 do        { строки влево }
            Begin
              Buf:=a[i,1];
              For j:=1 to n-1 do
                a[i,j]:=a[i,j+1];
              a[i,n]:=Buf;
              Inc(k);
            End;
        End;
    Writeln('                Преобразованная матрица');
    ScreenMatrix;
  If IndPrinter then
    Begin
      Writeln(Lst);
      Writeln(Lst, 'Преобразованная матрица');
      PrinterMatrix;
    End;

```

```

{ Вычисление следа преобразованной матрицы }
TraceMatrix;
Writeln('След преобразованной матрицы Trace = ',Trace:8:2);
If IndPrinter then
    Writeln(Lst,'След преобразованной матрицы '+
        'Trace = ',Trace:8:2);
WaitEscape;

```

End.

Результаты работы программы:

```

                Исходная матрица n = 6
1.00    10.00    0.00    3.00    13.00    23.00
5.00    55.00    15.00    7.00    17.00    0.00
9.00    19.00    0.00    81.00   88.00    18.00
6.00    16.00    66.00    33.00   21.00    12.00
99.00   15.00    44.00    0.00   13.00    11.00
10.00   11.00    12.00    17.00   23.00    14.00
След исходной матрицы Trace = 116.00

```

```

                Преобразованная матрица
23.00    1.00    10.00    0.00    3.00    13.00
5.00    55.00    15.00    7.00    17.00    0.00
0.00    81.00    88.00    18.00    9.00    19.00
12.00    6.00    16.00    66.00   33.00   21.00
44.00    0.00    13.00    11.00   99.00   15.00
14.00    10.00   11.00    12.00   17.00   23.00
След преобразованной матрицы Trace = 354.00

```

Варианты заданий

1. Для прямоугольной матрицы $A(m \times n)$ определить среднее арифметическое значение S ее элементов, среднее квадратичное отклонение G , а также процент элементов матрицы, расположенных в интервале $[-G, S + G]$.

$$S = \frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n a_{i,j}; \quad G = \sqrt{\frac{1}{mn-1} \sum_{i=1}^m \sum_{j=1}^n (a_{i,j} - S)^2}$$

2. Задана квадратная матрица $A(n \times n)$. Переставить в обратном порядке элементы тех столбцов матрицы, которые расположены ниже ее главной диагонали.

Примечание. Вполне понятно, что последний столбец матрицы обработке не подвергается, поскольку в этом столбце под главной диагональю нет ни одного элемента. В предпоследнем столбце под главной диагональю находится лишь один элемент, который нет смысла обменивать сам с собою.

3. В каждой строке прямоугольной матрицы $A(m \times n)$ отрицательным элементам присвоить нулевое значение, после чего перенести все положительные элементы в начало строки в порядке их исходного относительного расположения.

4. Каждую строку прямоугольной матрицы $A(m \times n)$ заменить суммой ниже расположенных строк, включая данную строку. Определить значение и местоположение максимального элемента матрицы до и после ее преобразования.

5. Для каждого столбца прямоугольной целочисленной матрицы $A(m \times n)$ подсчитать сумму элементов, которые входят в состав этого столбца и определить, есть ли столбцы с одинаковой суммой. Подсчитать количество пар таких столбцов.

Примечание. Обозначим количество пар столбцов с одинаковой суммой через p . Легко показать, что в общем случае может быть $p > n$.

Пусть $n = 4$ и все суммы одинаковы. Тогда получим пары

$$1 - 2, 1 - 3, 1 - 4, 2 - 3, 2 - 4, 3 - 4.$$

Следовательно, в этом случае $p = 6 > n$.

6. В каждой строке и в каждом столбце квадратной матрицы $A(n \times n)$ помещается строго по одному нулевому элементу. Перестановкой строк добиться того, чтобы все нули были расположены на главной диагонали матрицы. Буферный массив не использовать.

7. Выполнить текущее сглаживание каждой строки прямоугольной матрицы $A(m \times n)$ и определить, как изменилось максимальное отклонение dS_{max} ее элементов от среднего арифметического значения S данной строки до и после сглаживания.

Примечание. При текущем сглаживании массива x_1, x_2, \dots, x_n каждый j -ый элемент массива ($j = 2, 3, \dots, n-1$) заменяется средним арифметическим значением элементов с индексами $j-1, j, j+1$. Отклонение элемента x_i от параметра S равно $dS = |x_i - S|$.

8. В каждой строке прямоугольной матрицы $A(m \times n)$ перенести максимальный элемент в последнюю позицию строки, сдвинув при этом влево расположенные после него элементы. Учесть частный случай, когда максимальный элемент уже находится в последней позиции строки.

Пример. Строка 5 18 21 12 10 24 13 17 8 10 после преобразования будет иметь вид 5 18 21 12 10 13 17 8 10 24.

9. В прямоугольной матрице $A(m \times n)$ каждый нулевой элемент заменить средним арифметическим значением ненулевых элементов той строки, в которой расположен данный элемент. При этом все нулевые элементы строки должны быть заменены одним и тем же значением.

10. В каждом столбце квадратной матрицы $A(n \times n)$ найти максимальный по модулю элемент и, если он не является диагональным, поменять его местами с диагональным элементом. Подсчитать количество таких перестановок.

11. Дана квадратная матрица $A(n \times n)$. Если в ее треугольной части, расположенной выше побочной диагонали, имеются нулевые элементы, то заменить каждый из них минимальным, но отличным от нуля значением элементов столбца, в котором расположен нулевой элемент.

Примечание. Нулевые элементы заменяются лишь выше побочной диагонали, но при поиске минимального элемента рассматривается весь столбец !

Рекомендация.

1) Организовать цикл просмотра столбцов по j от 1 до $n-1$ (при $j = n$ выше побочной диагонали нет ни одного элемента).

2) В j -ом столбце при просмотре его элементов найти минимальное, но не равное нулю значение a_{\min}

3) В том же j -ом столбце просматривать лишь элементы $a_{i,j}$, расположенные выше побочной диагонали. Если $a_{i,j} = 0$, то присвоить $a_{i,j} := a_{\min}$.

12. В каждой строке прямоугольной матрицы $A(m \times n)$ подсчитать количество k изолированных положительных элементов, то есть элементов, окруженных слева и справа хотя бы одним неположительным элементом. Для первого элемента смежными являются второй и последний элементы, для последнего – первый и предпоследний. Определить номера строк, которые имеют максимальное k_{\max} и минимальное k_{\min} количество таких элементов, после чего обменять эти строки местами. Учесть частный случай, когда $k_{\max} = k_{\min}$.

13. Для каждой строки прямоугольной целочисленной матрицы $A(m \times n)$ определить сумму ее положительных элементов, после чего сгруппировать строки в порядке уменьшения этих сумм.

14. В прямоугольной матрице $A(m \times n)$ определить значение и местоположение двух элементов, которые имеют соответственно максимальное и минимальное превышение относительно среднего значения соседних элементов, после чего обменять эти элементы местами. Элементы, расположенные на периметре матрицы, не анализировать.

Указание. Для элемента с индексами (i, j) , $i = 2..m-1$; $j = 2..n-1$ соседними элементами считать элементы, смежные с ним по вертикали и по горизонтали.

15. При последовательном просмотре прямоугольной целочисленной матрицы $A(m \times n)$ сформировать из ее элементов одномерный массив $B(k)$, $k \leq m \cdot n$, при этом в массиве B не должно быть одинаковых чисел.

Рекомендация. Перенесем вначале в массив B первый элемент матрицы A , т.е. установим $b_1 = a_{1,1}$, $k = 1$ (k – количество элементов в массиве B). После этого, просматривая в цикле все элементы матрицы A , определить, имеется ли очередной элемент $a_{i,j}$ в массиве B и, если такой элемент не обнаружен, то добавить его в массив B , увеличив одновременно значение параметра k .

16. Рассматривая каждый столбец прямоугольной матрицы $A(m \times n)$ как вектор, определить номера двух векторов-столбцов, длины которых соответственно максимальная и минимальная, после чего обменять эти столбцы местами.

Примечание. Длина вектора, представленного одномерным массивом $X = (x_1, x_2, \dots, x_n)$, равна

$$L = \sqrt{\sum_{i=1}^n x_i^2}$$

17. Определить, есть ли в прямоугольной матрице $A(m \times n)$ линейно зависимые строки и подсчитать количество пар таких строк.

Примечание. Две строки матрицы линейно зависимы, если одну из них можно получить из другой умножением на постоянный коэффициент, отличный от нуля.

Рекомендация. Рассмотрим i -ую и k -ую строки ($i \neq k$). Выберем в этих строках два элемента $a_{i,j}$ и $a_{k,j}$ такие, что $a_{i,j} \neq 0$ и $a_{k,j} \neq 0$. Если эти строки линейно зависимы, то частное $\frac{a_{i,j}}{a_{k,j}}$ или $\frac{a_{k,j}}{a_{i,j}}$ должно иметь целочисленное значение.

Определим $p_1 = a_{i,j} \operatorname{div} a_{k,j}$, $q_1 = a_{i,j} \bmod a_{k,j}$, $p_2 = a_{k,j} \operatorname{div} a_{i,j}$, $q_2 = a_{k,j} \bmod a_{i,j}$. Если $p_1 \neq 0$ и $q_1 = 0$ или $p_2 \neq 0$ и $q_2 = 0$, то строки i и k могут быть линейно зависимыми, в противном случае они наверняка не являются линейно зависимыми. Если $p_1 \neq 0$ и $q_1 = 0$, то для всех элементов k -ой строки должно выполняться $a_{k,j} = p_1 a_{i,j}$. Если $p_2 \neq 0$ и $q_2 = 0$, то аналогично $a_{i,j} = p_2 a_{k,j}$.

Указание. При проверке отношения $a_{k,j} = p_1 a_{i,j}$ цикл обработки строк нужно прекратить, если обнаружено $a_{k,j} \neq p_1 a_{i,j}$. Аналогично поступить при проверке отношения $a_{i,j} = p_2 a_{k,j}$.

18. В прямоугольной матрице $A(m \times n)$ найти два элемента, которые в наименьшей и наибольшей степени отличаются от среднего арифметического значения элементов данной матрицы, после чего обменять их местами.

Примечание. Степень отличия элемента $a_{i,j}$ от среднего арифметического значения S определяется значением параметра $dS = |a_{i,j} - S|$. Следовательно, в задаче идет речь о нахождении таких элементов матрицы, для которых параметр dS имеет соответственно максимальное или минимальное значение.

19. Элементами прямоугольной матрицы $A(m \times n)$ могут быть только числа -1 , 0 и 1 . Для каждого столбца матрицы нужно выполнить следующее: если сумма элементов столбца не равна нулю, то заменить часть нулевых элементов значением $+1$ или -1 таким образом, чтобы указанная сумма как можно меньше отличалась от нуля.

20. К каждому элементу главной диагонали целочисленной квадратной матрицы $A(n \times n)$ прибавить такое значение d , чтобы сумма элементов в соответствующем столбце матрицы была нулевой. Сформировать вектор, i -ый компонент которого равен значению параметра d , после чего сгруппировать столбцы матрицы в порядке возрастания компонент этого вектора.

Примечание. Параметр d может быть как положительным, так и отрицательным.

21. Для каждого столбца прямоугольной матрицы $A(m \times n)$, элементами которой являются целые положительные числа, определить сумму элементов, которые входят в его состав, и если она нечетная, прибавить единицу к значению последнего элемента данного столбца, после чего сгруппировать отдельно элементы каждого столбца в порядке уменьшения их значений.

22. Рассматривая каждую строку прямоугольной матрицы $A(m \times n)$ как вектор, найти номера i -ой и k -ой векторов-строк ($i = 1 \dots m-1$; $k = i+1 \dots m$), для которых модуль их скалярного произведения имеет максимальное значение, после чего обменять эти строки местами.

Примечание. Скалярное произведение двух векторов, которые описываются как одномерные массивы $X = (x_1, x_2, \dots, x_n)$ и $Y = (y_1, y_2, \dots, y_n)$, определяется по формуле

$$S = \sum_{i=1}^n x_i y_i$$

23. В прямоугольной вещественной матрице $A(m \times n)$ каждый нулевой элемент заменить средним арифметическим значением ненулевых элементов того столбца, в котором расположен этот элемент. Просмотр столбца выполнять сверху вниз, при замене очередного нулевого элемента учитывать значения замененных к этому моменту нулевых элементов. Цикл просмотра столбца выполнять только один раз.

24. Главная и побочная диагонали разделяют элементы квадратной матрицы $A(n \times n)$ на четыре сектора, которые условно можно назвать левый, правый, верхний и нижний. Считая, что элементы указанных диагоналей не входят в состав секторов, найти в левом секторе минимальный по модулю, а в правом – максимальный по модулю элементы, после чего обменять их местами.

25. В прямоугольной матрице $A(m \times n)$ рассмотреть квадратные подматрицы размерностью $1, 2, 3, \dots, n$, причем для всех подматриц левым верхним элементом является элемент начальной матрицы с индексами $(1,1)$. Определить номера двух подматриц, среднее арифметическое элементов которых имеет соответственно наибольшее и наименьшее значение.

26. В прямоугольной матрице $A(m \times n)$ найти номера столбцов, которые содержат соответственно максимальное и минимальное количество отрицательных элементов, после чего обменять их местами. Учесть возможные частные случаи (в матрице нет отрицательных элементов; лишь один столбец матрицы содержит такие элементы; лишь два столбца содержат отрицательные элементы, но их количество одинаковое).

27. В прямоугольной матрице $A(m \times n)$ определить среднее арифметическое значение S и дисперсию D отдельно для положительных и отрицательных элементов данной матрицы (нулевые элементы при вычислении S и D не учитывать).

Примечание.

В целом для матрицы

$$S = \frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n a_{i,j}; \quad D = \frac{1}{mn-1} \sum_{i=1}^m \sum_{j=1}^n (a_{i,j} - S)^2$$

Если в матрице нет положительных элементов, то принять $S = 0, D = 0$; если в матрице только один такой элемент со значением b , то принять $S = b, D = 0$. Аналогичные действия выполнить относительно отрицательных элементов.

28. В прямоугольной матрице $A(m \times n)$ определить количество строк, элементы которых полностью упорядочены по возрастанию, после чего перенести эти строки в начальную часть матрицы, сохранив их первоначальное относительное расположение.

Примечание. Элементы массива x_1, x_2, \dots, x_n упорядочены по возрастанию, если $x_1 \leq x_2 \leq x_3 \leq \dots \leq x_n$.

Указание. При просмотре элементов строки в случае выявления пары элементов, которые нарушают ее упорядоченность, т.е. имеет место $a_{i,j} > a_{i,j+1}$ ($j = 1..n-1$), дальнейший анализ элементов строки прекратить.

29. В каждом столбце прямоугольной матрицы $A(m \times n)$ есть по крайней мере один нулевой элемент. Требуется заменить в столбце последний нулевой элемент таким значением, чтобы сумма элементов столбца равнялась нулю. Сформировать вектор, i -ый компонент которого равен разности между суммой элементов i -ой строки (не столбца!) до преобразования матрицы и суммой элементов этой же строки после ее преобразования.

30. Обнулить элементы k -ой строки и l -го столбца прямоугольной матрицы $A(m \times n)$, после чего присвоить этим элементам такие значения, чтобы сумма элементов каждой строки и сумма элементов каждого столбца, за исключением l -го, равнялась нулю. Значения k и l ввести с клавиатуры.

Пример.

Пусть $k = 3$, $l = 2$ и матрица после обнуления k -ой строки и l -го столбца имеет вид:

$$A = \begin{pmatrix} 5 & 0 & -6 & 4 & -10 \\ -3 & 0 & 7 & -6 & 7 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 3 & 3 & 8 \end{pmatrix}$$

Последовательность обработки:

1) Суммируем элементы каждого p -го столбца ($p = 1..n$; $p \neq l$) и заменяем элементы $a_{k,p}$ полученной суммой с обратным знаком:

$$A = \begin{pmatrix} 5 & 0 & -6 & 4 & -10 \\ -3 & 0 & 7 & -6 & 7 \\ -3 & 0 & -10 & -1 & 5 \\ 1 & 0 & 3 & 3 & 8 \end{pmatrix}$$

2) Суммируем элементы каждой i -ой строки ($i = 1..m$) и заменяем элемент $a_{i,l}$ полученной суммой с обратным знаком:

$$A = \begin{pmatrix} 5 & 7 & -6 & 4 & -10 \\ -3 & -5 & 7 & -6 & 7 \\ -3 & 19 & -10 & -1 & 5 \\ 1 & 21 & 3 & 3 & 8 \end{pmatrix}$$

31. В квадратной матрице $A(n \times n)$ определить сумму положительных элементов, расположенных над главной диагональю, и сумму отрицательных элементов, расположенных под главной диагональю. Если первая из них превышает по модулю вторую, то транспонировать матрицу.

32. Для прямоугольной матрицы $A(m \times n)$ найти минимальный из положительных элементов и максимальный из отрицательных элементов, после чего обменять их местами. Нулевые элементы не учитывать.

Указание. Учесть частный случай, когда в матрице нет положительных или отрицательных элементов.

33. В прямоугольной матрице $A(m \times n)$ определить количество столбцов, которые целиком состоят из положительных элементов.

Указание. Последовательный просмотр элементов столбца организовать таким образом, чтобы при обнаружении первого неположительного элемента остальные элементы столбца не проверялись.

34. Часть элементов прямоугольной матрицы $A(m \times n)$ имеет нулевое значение. Заменить каждый такой элемент суммой смежных с ним элементов.

Примечание. Угловые элементы матрицы имеют два смежных элемента; элементы, расположенные на периметре матрицы, но не угловые, имеют три смежных элемента; другие элементы матрицы имеют четыре смежных элемента.

Указание. Для формирования преобразованной матрицы в данном случае необходимо использовать буферный массив (дополнительную матрицу).

Пусть фрагмент матрицы имеет вид

$$\begin{array}{cccc} 10 & 3 & & \\ -4 & 0 & 0 & 6 \\ & 7 & -8 & \end{array}$$

Если преобразование выполняется в этой же матрице, то получим

$$\begin{array}{cccc} 10 & 3 & & \\ -4 & 13 & 14 & 6 \\ & 7 & -8 & \end{array}$$

В этом случае при вычислении нового значения для второго нулевого элемента используется уже вычисленное значение первого нулевого элемента, что неверно. Здесь должно быть

$$\begin{array}{cccc} 10 & 3 & & \\ -4 & 13 & 1 & 6 \\ & 7 & -8 & \end{array}$$

Естественно, после формирования преобразованной матрицы в буферном массиве ее элементы нужно переписать в исходную матрицу.

35. Рассматривая элементы строки прямоугольной вещественной матрицы $A(m \times n)$ как координаты точки в n -мерном пространстве, определить номера точек, расстояние между которыми максимальное, после чего поменять их местами.

Примечание. Если две n -мерные точки заданы двумя одномерными массивами $X = (x_1, x_2, \dots, x_n)$ и $Y = (y_1, y_2, \dots, y_n)$, то расстояние между ними определяется по формуле

$$L = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

36. Дана квадратная целочисленная матрица $A(n \times n)$. Если в ее треугольной части, расположенной ниже главной диагонали, имеются нечетные элементы, то заменить каждый из них средним арифметическим значением S четных элементов всего столбца, в котором расположен нечетный элемент. Среднее арифметическое значение должно быть округлено до ближайшего целого четного значения. Если в столбце под главной диагональю содержится несколько нечетных элементов, то каждый из них должен быть заменен одним и тем же значением.

Примечание. Округление значения S до ближайшего четного значения может быть выполнено следующим образом:

$$S := \text{trunc}(S)$$

$$a_{i,j} = \begin{cases} S + 1, & \text{если } \text{odd}(S) = \text{true} \\ S, & \text{если } \text{odd}(S) = \text{false} \end{cases}$$

Пример.

S	Trunc(S)	$a_{i,j}$
3.8	3	4
3.2	3	4
2.8	2	2
2.2	2	2

37. В состав прямоугольной матрицы $A(m \times n)$ входят целые положительные числа. Требуется в каждой строке матрицы определить сумму элементов, которые являются удвоенными нечетными числами, после чего сгруппировать строки в порядке уменьшения этих сумм.

Примечание. Определить, является ли элемент $a_{i,j}$ удвоенным нечетным числом, можно следующим образом:

$$b := a_{i,j}; \quad d := b \text{ div } 2$$

Если $\text{odd}(b) = \text{false}$ и $\text{odd}(d) = \text{true}$, то элемент $a_{i,j}$ является удвоенным нечетным числом.

38. Для прямоугольной матрицы $A(m \times n)$, элементами которой являются целые положительные числа, определить среднее арифметическое значение S и среднее квадратичное отклонение G отдельно для четных и нечетных элементов. Учесть, что переменные S и G по своей природе являются вещественными числами.

Примечание. См. п.1 и п.27.

39. Для прямоугольной матрицы $A(m \times n)$ сформировать целочисленный вектор $V = (b_1, b_2, \dots, b_m)$. Элементу b_i присвоить значения 1, если i -ая строка имеет не меньше двух подряд расположенных нулевых элементов, в противном случае $b_i = 0$. Строки, для которых определено значение $b_i = 1$, перенести в начало матрицы, сохранив их исходный относительный порядок.

Указание. Если при анализе строки обнаружены два смежных нулевых элемента, дальнейший просмотр строки прекратить.

40. Для заданной целочисленной квадратной матрицы $A(n \times n)$ проверить, имеет ли место совпадение k -ой строки и k -го столбца ($k = 1..n$). Напечатать номера совпадающих строк и столбцов.

41. Задана целочисленная квадратная матрица $A(n \times n)$. Сгруппировать элементы каждой строки, расположенные выше главной диагонали, в порядке уменьшения их абсолютных значений.

42. Если максимальный элемент i -ой строки прямоугольной матрицы $A(m \times n)$, $i = 1..m$ больше суммы других элементов данной строки, а модуль минимального элемента меньше суммы других элементов строки, то заменить максимальный и минимальный элементы полусуммой их значений, в противном случае строку матрицы

оставить без изменений. Подсчитать количество строк матрицы, в которых сделана указанная замена.

43. Нулевые элементы каждого столбца прямоугольной матрицы $A(m \times n)$ переместить в начало этого же столбца, сохранив без изменения относительную последовательность других элементов столбца.

44. В каждой строке целочисленной прямоугольной матрицы $A(m \times n)$ найти максимальный по модулю элемент и заменить его средним арифметическим значением S других элементов данной строки. Параметр S должен быть округлен до ближайшего целого значения.

45. Рассматривая каждый столбец прямоугольной вещественной матрицы $A(m \times n)$ как вектор, определить номера j и k ($j = 1 \dots n-1$; $k = j+1 \dots n$) векторов-столбцов, угол между которыми максимальный, после чего обменять эти столбцы местами.

Примечание. Косинус угла между двумя векторами - это скалярное произведение векторов, разделенное на произведение их модулей (модуль вектора - это его длина). Если два вектора описаны одномерными массивами $X = (x_1, x_2, \dots, x_n)$ и $Y = (y_1, y_2, \dots, y_n)$, то тогда имеем

$$\cos \alpha = \frac{\sum_{i=1}^n x_i y_i}{\sqrt{\sum_{i=1}^n x_i^2} \sqrt{\sum_{i=1}^n y_i^2}}$$

46. В каждой строке прямоугольной матрицы $A(m \times n)$ определить среднее арифметическое значение S его элементов, после чего элементам $a_{i,j} < S$ присвоить нулевое значение. Сгруппировать строки матрицы в порядке уменьшения параметра S .

47. Задана цело численная прямоугольная матрица $A(m \times n)$. Если сумма элементов ее i -ой строки ($i = 1..m$) нечетная, то добавить единицу к значению последнего элемента $a_{i,n}$, после чего сгруппировать элементы этой строки в порядке уменьшения их абсолютных значений.

48. В каждой строке прямоугольной матрицы $A(m \times n)$ удалить максимальный элемент, сдвинув на одну позицию влево расположенные после него элементы. Последнему элементу строки присвоить нулевое значение. Определить, как изменилось при этом среднее арифметическое значение элементов матрицы.

49. Задана верхняя треугольная матрица $A(n \times n)$ ($a_{i,j} = 0$ для всех $i > j$; $i, j = 1..n$) с вещественными элементами. В каждой строке этой матрицы, кроме первой, заменить нулевые элементы такими равными друг другу значениями, чтобы сумма элементов данной строки стала равной нулю.

Примечание.. Элементами верхней треугольной матрицы считаются элементы, расположенные над главной диагональю, и элементы главной диагонали. Элементы под главной диагональю при этом должны иметь нулевое значение.

50. В прямоугольной матрице $A(m \times n)$ найти три минимальных элемента и заменить их нулевым значением. Определить, как изменилось при этом среднее арифметическое значение элементов матрицы.

Указание. Поиск трех минимальных элементов выполнять при однократном просмотре элементов матрицы.

Рекомендация. Для упрощения программы будем считать, что матрица имеет не менее трех столбцов. Последовательность решения задачи:

1) $\text{amin1} := a_{1,1}; \text{amin2} := a_{1,2}; \text{amin3} := a_{1,3};$

$\text{imin1} := 1; \text{jmin1} := 1; \text{imin2} := 1; \text{jmin2} := 2; \text{imin3} := 1; \text{jmin3} := 3;$

2) Сгруппировать переменные $\text{amin1}, \text{amin2}, \text{amin3}$ по увеличению их значений. При перестановке указанных переменных одновременно переставлять принадлежащие им индексы $\text{imin1}, \text{jmin1}, \text{imin2}, \text{jmin2}, \text{imin3}, \text{jmin3}$.

3) Последовательно просматривать элементы матрицы.

Если $a_{i,j} < \text{amin1}$, то

$\text{amin3} := \text{amin2}; \text{imin3} := \text{imin2}; \text{jmin3} := \text{jmin2};$

$\text{amin2} := \text{amin1}; \text{imin2} := \text{imin1}; \text{jmin2} := \text{jmin1};$

$\text{amin1} := a_{i,j}; \text{imin1} := i; \text{jmin1} := j;$

иначе

если $a_{i,j} < \text{amin2}$, то

$\text{amin3} := \text{amin2}; \text{imin3} := \text{imin2}; \text{jmin3} := \text{jmin2};$

$\text{amin2} := a_{i,j}; \text{imin2} := i; \text{jmin2} := j;$

иначе

если $a_{i,j} < \text{amin3}$, то

$\text{amin3} := a_{i,j}; \text{imin3} := i; \text{jmin3} := j.$

4) Заменить нулями элементы $a_{i \min 1, j \min 1}, a_{i \min 2, j \min 2}, a_{i \min 3, j \min 3}$.

51. В каждой i -ой строке прямоугольной матрицы $A(m \times n)$ найти количество k элементов $a_{i,j}$, которые превышают полусумму элементов $a_{i,j-1}$ и $a_{i,j+1}$, смежных с элементом $a_{i,j}$, после чего сгруппировать строки матрицы в порядке возрастания параметра k . Первый и последний элементы строки не рассматривать.

52. Элементы каждой строки прямоугольной матрицы $A(m \times n)$ сдвинуть циклически вправо, не изменяя при этом положение максимального элемента данной строки. Например, для строки

8 12 -6 14 11 -3 7 9

получим

9 8 12 14 -6 11 -3 7.

Рекомендация. Обозначим значение максимального элемента в i -ой строке через a_{\max} , его положение (индекс в строке) через j_{\max} .

Для решения поставленной задачи необходимо рассматривать три частных случая: $j_{\max} = 1$; $j_{\max} = n$; $1 < j_{\max} < n$. Ниже представлена возможная последовательность обработки исходного массива.

1) $j_{\max} = 1$

$R = a_{i,n}$

$a_{i,j} = a_{i,j-1}; j = n - 1..3$

$$a_{i,2} := R$$

2) $j_{\max} = n$

$$R := a_{i,n-1}$$

$$a_{i,j} := a_{i,j-1}; \quad j = n-2..2$$

$$a_{i,1} := R$$

3) $1 < j_{\max} < n$

$$R := a_{i,n}$$

$$a_{i,j} := a_{i,j-1}; \quad j = n-1 .. j_{\max} + 1$$

$$a_{i,j_{\max}+1} := a_{i,j_{\max}-1}$$

$$a_{i,j} := a_{i,j-1}; \quad j = j_{\max} - 1 .. 2$$

$$a_{i,1} := R$$

53. В каждом столбце прямоугольной матрицы $A(m \times n)$ поменять местами минимальный элемент с элементом главной диагонали, если элемент главной диагонали, относящийся к данному столбцу, имеет отрицательное значение. Подсчитать количество таких обменов.

54. В квадратной матрице $A(n \times n)$ все элементы, расположенные ниже главной диагонали, равны нулю. В каждом столбце, кроме последнего, присвоить элементу, расположенному непосредственно ниже главной диагонали, такое значение, чтобы сумма элементов в столбце без учета элемента главной диагонали равнялась нулю.

55. Прямоугольная матрица $A(m \times 3)$ имеет m строк и 3 столбца. Считая элементы строки длинами отрезков, определить, сколько треугольников может быть построено из отрезков, которые размещаются в строках матрицы. Определить при этом номер треугольника, который имеет наибольшую площадь, для чего использовать формулу

$$S = \sqrt{d}; \quad d = p(p-a)(p-b)(p-c),$$

где a, b, c – длины сторон треугольника, p – его полупериметр.

Примечание. Треугольник не может быть построен, если $d \leq 0$.

56. В каждой строке прямоугольной матрицы $A(m \times n)$ определить разность d между средним арифметическим значением S_1 элементов, которые находятся на четных местах, и средним арифметическим значением S_2 элементов, которые находятся на нечетных местах. При этом учесть, что количество столбцов может быть как четным, так и нечетным. Сгруппировать строки в порядке уменьшения параметра d .

57. Каждая строка прямоугольной матрицы $A(m \times n)$ определяет координаты точки в n -мерном пространстве. Точки принадлежат ломаной линии, начало и конец которой определяется точками, содержащимися в первой и последней строках матрицы. Вычислить длину ломаной и определить номер ее отрезка, который имеет наибольшую длину.

Примечание. Длина отрезка, начальная и конечная точки которого заданы i -ой и $(i+1)$ -ой строками матрицы ($i = 1 .. m-1$; $k = i+1 .. m$) определяется по формуле

$$L_i = \sqrt{\sum_{j=1}^n (a_{i,j} - a_{i+1,j})^2}$$

58. Для каждой строки прямоугольной матрицы $A(m \times n)$ определить количество нарушений k условия упорядоченности ее элементов по возрастанию, то есть условия $a_{i,j} \leq a_{i,j+1}$ ($j = 1 \dots n-1$). Сгруппировать строки в порядке возрастания параметра k .

59. Рассматривая в квадратной матрице $A(n \times n)$ диагональ, которая соединяет левый нижний элемент с правым верхним элементом, определить минимальное по модулю число среди элементов, расположенных выше данной диагонали, и максимальное по модулю число среди элементов, расположенных ниже этой же диагонали, после чего обменять местами соответствующие этим числам элементы матрицы.

60. В каждом столбце прямоугольной матрицы $A(m \times n)$ перенести максимальный по модулю элемент в последнюю позицию столбца, сдвинув при этом вверх расположенные после него элементы. Учесть частный случай, когда максимальный элемент уже находится в последней позиции столбца.

Пример. Столбец

5 -18 21 -12 10 -24 13 17 -8 10

после преобразования будет иметь вид

5 -18 21 -12 10 13 17 -8 10 -24 .

61. Для прямоугольной матрицы $A(m \times n)$ сформировать одномерный массив $B(m)$, i -ый элемент которого равняется номеру первого отрицательного элемента в i -ой строке. Сгруппировать строки матрицы в порядке уменьшения элементов b_i .

62. В прямоугольной матрице $A(m \times n)$ определить номера строк, сумма модулей элементов которых наибольшая и наименьшая по сравнению с другими строками, после чего обменять эти строки местами.

63. К минимальному по модулю элементу каждой строки матрицы $A(m \times n)$ прибавить такое значение, чтобы сумма элементов строки была нулевой. Сформировать вектор $B(n)$, j -ая компонента которого равна разности d между суммой элементов в j -ом столбце (не строке!) до преобразования матрицы и суммой элементов в этом же столбце после преобразования. Сгруппировать столбцы в порядке уменьшения параметра d .

64. Дана квадратная матрица $A(n \times n)$. Если в ее треугольной части, расположенной ниже побочной диагонали, имеются нулевые элементы, то заменить каждый из них минимальным, но отличным от нуля элементом строки, в которой расположен нулевой элемент.

Примечание. Нулевые элементы заменяются лишь ниже побочной диагонали, но при поиске минимального элемента рассматривается вся строка!

65. Если в j -ом столбце квадратной матрицы $A(n \times n)$, $j=1..n$ максимальный по модулю элемент находится на главной диагонали, то разделить все элементы данного столбца на значение максимального элемента (не на модуль этого значения!); в противном случае столбец матрицы оставить без изменения. Если расположенный на главной диагонали максимальный по модулю элемент равен нулю, то для преобразования столбца найти в нем ближайший по значению ненулевой элемент. Если все элементы

столбца нулевые, то преобразование этого столбца не производить. Подсчитать общее количество преобразованных столбцов матрицы.

66. Для каждого столбца прямоугольной матрицы $A(m \times n)$, все элементы которого положительные числа, вычислить среднее гармоническое S , после чего сгруппировать столбцы в порядке уменьшения значения S .

Примечание. Для массива x_1, x_2, \dots, x_n $(x_i > 0)$

$$S = \frac{n}{\frac{1}{x_1} + \frac{1}{x_2} + \dots + \frac{1}{x_n}}$$

67. Для каждого столбца прямоугольной матрицы $A(m \times n)$ определить разность d между суммой модулей элементов, которые находятся на четных местах, и суммой модулей элементов, которые находятся на нечетных местах. При этом учесть, что количество строк в матрице может быть как четным, так и нечетным. Сгруппировать столбцы матрицы в порядке уменьшения параметра d .

68. В прямоугольной целочисленной матрице $A(m \times n)$ определить значение и местоположение максимального по модулю и минимального по модулю четных элементов, после чего обменять их местами. Учесть частные случаи (в матрице нет четных элементов; в матрице лишь один четный элемент; минимальный и максимальный по модулю четные элементы одинаковы).

69. В прямоугольной матрице $A(m \times n)$ определить количество столбцов, которые содержат только числа одного знака (положительные или отрицательные) и не имеют нулевых элементов. Переместить такие столбцы в начальную часть матрицы, сохранив их первоначальный относительный порядок.

Указание. Два смежных элемента в j -ом столбце имеют одинаковый знак, если $a_{i,j} \cdot a_{i+1,j} > 0$ ($i = 1..m-1$). Если при просмотре столбца будет обнаружено, что $a_{i,j} \cdot a_{i+1,j} \leq 0$, то дальнейший анализ этого столбца не производить.

70. Для каждой строки прямоугольной матрицы $A(m \times n)$ определить количество k элементов, которые совпадают по знаку с соответствующими элементами последней строки. Сгруппировать строки матрицы в порядке возрастания параметра k (вполне очевидно, что для последней строки $k = n$).

Указание. Для определения знака элемента $a_{i,j}$ использовать функцию

$$\text{sign}(a_{i,j}) = \begin{cases} 1, & \text{если } a_{i,j} > 0 \\ 0, & \text{если } a_{i,j} = 0 \\ -1, & \text{если } a_{i,j} < 0 \end{cases}$$

71. Элементами прямоугольной матрицы $A(m \times n)$ являются целые числа (положительные, отрицательные и нулевые). В каждой строке матрицы проанализировать элементы, симметричные относительно середины строки (1 и n , 2 и $n-1$ и т.д.) и, если они равны по модулю, но противоположны по знаку, то обменять их местами. Подсчитать общее количество k перестановок в каждой строке. Строки, для которых $k = 0$, перенести в начальную часть матрицы, сохранив первоначальный относительный порядок этих строк.

72. Рассматривая каждый столбец прямоугольной вещественной матрицы $A(m \times n)$ как вектор, определить, есть ли в матрице ортогональные векторы-столбцы. Подсчитать количество таких пар векторов.

Примечание. Для ортогональных векторов их скалярное произведение S равняется нулю. Значение S принять равным нулю, если $|S| < \varepsilon$, где ε - малое число (например, 0.001).

О вычислении скалярного произведения см.п.22.

73. Для прямоугольной матрицы $A(m \times n)$ определить значение и местоположение элемента, который является седловой точкой матрицы (если имеются такие точки). Первую седловую точку обменять местами с ближайшим элементом, расположенным на периметре матрицы. Если расстояние от седловой точки до двух или более сторон периметра одинаковые, то для обмена использовать любую из таких сторон периметра.

Указание. Седловой точкой матрицы считают не расположенный на ее периметре элемент с индексом (i, j) , который является минимальным в i -ой строке и одновременно максимальным в j -ом столбце.

74. Для каждой строки прямоугольной матрицы $A(m \times n)$ определить максимальное значение S_{\max} из сумм пар рядом расположенных элементов строки (1 и 2, 3 и 4 и т.д.). Если количество столбцов n матрицы нечетное, то в качестве последней суммы принять значение n -го элемента. Сгруппировать строки матрицы в порядке уменьшения параметра S_{\max} .

75. Для каждого столбца прямоугольной вещественной матрицы $A(m \times n)$ определить сумму модулей его элементов, а потом сгруппировать столбцы в порядке возрастания этих сумм.

76. Для каждой строки прямоугольной матрицы $A(m \times n)$ определить количество k элементов, значение которых расположено в заданном интервале $[a, b]$, $a \leq b$. Параметры a и b ввести с клавиатуры. Сгруппировать строки матрицы в порядке уменьшения значения k .

77. В каждом столбце прямоугольной матрицы $A(m \times n)$ определить значение d_{\max} и номер k_{\max} элемента, который в наибольшей степени отличается от среднего арифметического значения S элементов данного столбца, после чего сгруппировать столбцы в порядке уменьшения параметра d_{\max} .

Указание. При перестановке двух элементов массива d_{\max} одновременно переставлять соответствующие им элементы массива k_{\max} .

Об определении степени отличия элемента $a_{i, j}$ от значения S см.п.18.

78. Главная и побочная диагонали разделяют элементы квадратной матрицы $A(n \times n)$ на четыре сектора, которые условно можно назвать левый, правый, верхний и нижний. Считая, что элементы указанных диагоналей входят в состав секторов, найти в верхнем секторе минимальный по модулю, а в нижнем – максимальный по модулю элементы, после чего обменять их местами.

79. Каждый столбец прямоугольной матрицы $A(m \times n)$ определяет координаты точки в m -мерном пространстве. Определить номера точек, наименее удаленных одна от другой, после чего обменять эти точки местами, если они не совпадают друг с другом, т.е. если расстояние между ними не равно нулю.

Указание. Об определении расстояния L между двумя n -мерными точками см. п.35. При этом считать $L = 0$, если $L \leq \varepsilon$, где ε - малое число (например, 0.001).

80. Задана вещественная квадратная матрица $A(n \times n)$. Сгруппировать элементы каждого столбца, расположенные ниже главной диагонали, в порядке возрастания их абсолютных значений. Определить, как при этом изменилось положение максимального элемента среди тех, которые расположены ниже главной диагонали.

81. В прямоугольной матрице $A(m \times n)$ переставить каждую пару строк (1,2), (3,4),... таким образом, чтобы первой была строка с меньшей суммой элементов. Определить количество таких перестановок. Если в матрице нечетное количество строк, то последняя строка при преобразовании матрицы не обрабатывается.

82. В каждой строке прямоугольной целочисленной матрицы $A(m \times n)$ определить процент p четных положительных чисел относительно общего количества положительных чисел в данной строке, после чего сгруппировать строки в порядке уменьшения параметра p .

Примечание. Учесть случай, когда в строке нет положительных элементов (для исключения деления на нуль).

83. Элементы каждой строки прямоугольной матрицы $A(m \times n)$ заменить их дополнениями до максимального элемента этой же строки. Определить, насколько при этом изменится общая сумма элементов матрицы (абсолютное изменение и относительное в процентах).

Примечание. Пусть мы имеем две суммы $S1$ и $S2$. Тогда абсолютное изменение этих сумм равно $dS = S2 - S1$, относительное - $pS = \frac{dS}{S1} 100$. Поскольку в частном случае может быть $S1 = 0$, то для исключения деления на нуль при определении pS лучше использовать формулу

$$pS = \frac{dS}{\max(S1, S2)} 100$$

84. Для прямоугольной матрицы $A(m \times n)$ определить, сколько пар ее столбцов содержат одинаковое количество отрицательных элементов. После этого столбцы, в составе которых нет отрицательных элементов, переставить в начало матрицы, сохранив их исходный относительный порядок.

Примечание. В общем случае количество пар элементов p может превышать количество сравниваемых элементов m . Например, если в массиве $X = (x_1, x_2, x_3, x_4)$, $m = 4$ все элементы одинаковые, то получим пары элементов 1 - 2, 1 - 3, 1 - 4, 2 - 3, 2 - 3 4, 3 - 4. т.е. $p = 6 > m = 4$.

85. В прямоугольной матрице $A(m \times n)$ определить номера столбцов, среднее арифметическое значение модулей элементов которых наибольшее и наименьшее по сравнению с другими столбцами, после чего обменять эти столбцы местами.

86. Для каждого столбца прямоугольной матрицы $A(m \times n)$ найти разность d между максимальным и минимальным элементами, после чего сгруппировать столбцы в порядке уменьшения параметра d .

87. Элементами прямоугольной матрицы $A(m \times n)$ являются целые положительные числа. Требуется сгруппировать элементы каждой строки матрицы по уменьшению их значений и определить после этого, для скольких элементов выполняется условие $a_{i,j} > j$.

88. Задана нижняя треугольная матрица $A(n \times n)$ ($a_{i,j} = 0$ для всех $i < j$; $i, j = 1..n$) с вещественными элементами. В каждом столбце этой матрицы, кроме первого, заменить нулевые элементы такими равными друг другу значениями, чтобы сумма элементов данного столбца стала равной нулю.

Указание. Элементами нижней треугольной матрицы считаются элементы, расположенные под главной диагональю, и элементы главной диагонали. Элементы над главной диагональю при этом должны иметь нулевое значение.

89. Задана квадратная целочисленная матрица $A(n \times n)$. Если в ее треугольной части, расположенной выше главной диагонали, имеются нечетные элементы, то заменить каждый из них средним арифметическим значением четных элементов строки, в которой расположен нечетный элемент. Среднее арифметическое значение должно быть округлено до ближайшего целого четного значения. При наличии в строке над главной диагональю нескольких нечетных элементов их замена должна быть произведена одним и тем же значением. Если в строке нет четных элементов, замену не производить.

Примечание. Об округлении до ближайшего четного значения см. п.36.

90. Для прямоугольной матрицы $A(m \times n)$ сформировать одномерный массив $B(m)$, i -му элементу которого присвоить значение 1, если в i -ой строке матрицы имеются по крайней мере три элемента, упорядоченные по убыванию ($a_{i,j} \geq a_{i,j+1} \geq a_{i,j+2}$, $j = 1..n-2$), и значение 0 в противном случае. Переместить в начальную часть матрицы строки, для которых определены значения $b_i = 1$, сохранив при этом относительное расположение этих строк в исходной матрице.

Указание. Если при анализе элементов строки будет определено выполнение условия упорядоченности для трех элементов, дальнейший перебор элементов строки прекратить.

91. В каждом столбце прямоугольной матрицы $A(m \times n)$ определить минимальное по модулю значение S_{\min} для разностей пар расположенных рядом элементов столбца ($a_{1,j} - a_{2,j}$, $a_{3,j} - a_{4,j}$ и т.д.). Если количество строк матрицы нечетное, то в качестве последней пары принять значение последнего элемента столбца. Сгруппировать столбцы в порядке уменьшения параметра S_{\min} .

92. Элементы каждой строки квадратной матрицы $A(n \times n)$ переставить в обратном порядке. Определить после этого сумму элементов $S1$ над побочной диагональю и сумму элементов $S2$ под побочной диагональю, Если $S1 > S2$, то транспонировать матрицу.

93. Для прямоугольной вещественной матрицы $A(m \times n)$ сформировать одномерный целочисленный массив $B(n)$, j -ый элемент которого равен номеру первого

нулевого элемента в j -ом столбце матрицы, после чего сгруппировать столбцы в порядке уменьшения элементов b_j .

94. В прямоугольной матрице $A(m \times n)$ определить номера столбцов, которые имеют максимальную и минимальную сумму модулей своих элементов, и поменять местами эти столбцы.

95. Для квадратной вещественной матрицы $A(n \times n)$ вычислить скалярное произведение векторов X и Y , компоненты которых определяются следующим образом:

$$\begin{aligned}x_i &= \max(a_{i,1}, a_{i,2}, \dots, a_{i,n}), \\y_i &= \min(a_{1,i}, a_{2,i}, \dots, a_{n,i}),\end{aligned} \quad i = 1..n,$$

где x_i - максимальный элемент в i -ой строке;

y_i - минимальный элемент в i -ом столбце.

Примечание. О вычислении скалярного произведения см.п.22.

96. Каждая строка прямоугольной матрицы $A(m \times n)$ - это координаты точки в m -мерном пространстве. Определить, сколько точек расположено внутри гиперсферы с координатами центра (c_1, c_2, \dots, c_m) и радиусом R , вне гиперсферы и на ее поверхности. Параметр R и значения c_1, c_2, \dots, c_m вводит из отдельного файла.

Примечание. Точка находится внутри гиперсферы, если расстояние от нее до центра гиперсферы меньше радиуса гиперсферы.

О вычислении расстояния между двумя многомерными точками см.п.35.

97. Для каждого столбца прямоугольной матрицы $A(m \times n)$ вычислить среднее геометрическое значение P модулей его элементов и определить номера столбцов, для которых P имеет соответственно максимальное и минимальное значение (матрица не должна иметь нулевых элементов), после чего обменять эти столбцы местами.

Примечание. Для одномерного массива x_1, x_2, \dots, x_n

$$P = \sqrt[n]{\prod_{i=1}^n |x_i|}$$

98. В прямоугольной матрице $A(m \times n)$ определить количество строк, которые целиком состоят из отрицательных элементов, после чего переместить эти строки в начальную часть матрицы, сохранив их первоначальный относительный порядок.

Указание. Последовательный просмотр элементов строки организовать таким образом, чтобы при обнаружении первого неотрицательного элемента остальные элементы строки не проверялись.

99. Дана квадратная целочисленная матрица $A(n \times n)$. Если в ее треугольной части, расположенной ниже побочной диагонали, имеются четные элементы, то заменить каждый из них средним арифметическим значением нечетных элементов строки, в которой расположен четный элемент. Среднее арифметическое значение должно быть округлено до ближайшего целого нечетного значения. Если в строке нет нечетных элементов, замену не производить. Все преобразуемые в строке четные элементы должны быть заменены на одно и то же нечетное значение.

Примечание. Округление вещественного числа до ближайшего четного значения рассмотрено в п.36. Округление до ближайшего нечетного значения выполняется в основном аналогично.

100. Столбцы прямоугольной матрицы $A(m \times n)$ обменивать местами относительно вертикальной оси симметрии таким образом, чтобы первым был столбец с меньшей суммой элементов.

Лабораторная работа № 5

ИСПОЛЬЗОВАНИЕ ПРОЦЕДУР И ФУНКЦИЙ

Методические указания

Цель работы - практически освоить методы организации процедур и функций, а также способы их использования при обработке одномерных массивов с разными именами типов.

В любой из приведенных ниже задач необходимо составить программу, включающую в себя по крайней мере одну подпрограмму (процедуру или функцию), которая выполняет основную обработку исходных данных в соответствии с условием задачи. Все задачи сформулированы относительно обработки одного одномерного массива или пары одновременно обрабатываемых массивов. В программе, которая реализует лабораторную работу № 5, должна быть предусмотрена в первом случае последовательная обработка **трех разных массивов**, а во втором случае - **трех разных пар массивов**. При этом для обеспечения универсальности разработанной подпрограммы предполагается, что последовательно обрабатываемые в ней **массивы или пары массивов имеют разные имена типов**.

Условия некоторых задач ориентированы на сложение или вычитание двоичных чисел или, в общем случае, чисел, которые записаны в системе счисления с основанием $1 < q \leq 10$. При этом предполагается, что числа представлены в виде массива цифр. В этом случае настоятельно рекомендуется внимательно просмотреть приложение «Числа и системы счисления» в учебном пособии «Основы программирования на языке Турбо Паскаль» (в указанном приложении рассматриваются решения аналогичных задач).

Лабораторную работу № 5 рекомендуется выполнять в три этапа.

Этап 1. Программа составляется в предположении, что в ней обрабатывается только один массив (или одна пара массивов). На этом этапе должна быть тщательно проверена правильность реализации алгоритма решения задачи путем прогона системы тестовых входных данных. В систему тестов должны быть включены не только "нормальные" массивы, но и массивы, которые имеют некоторые предельные свойства (например, массив, который состоит из одних нулей, или массив, который состоит из одного элемента). Для реализации однотипных фрагментов программы рекомендуется применять процедуры и функции; в составе программы целесообразно применять также инструментальные процедуры и функции, которые могут быть использованы практически без изменения в разных программах (например, вывод одномерного массива, контроль размера страницы, которая выводится на экран и т.п.).

Этап 2. Разработанная программа перерабатывается в процедуру или функцию в предположении, что все последовательно обрабатываемые массивы (или пары массивов) имеют одинаковые имена типов. В основной программе в этом случае выполняется ввод и вывод исходных данных, последовательное обращение к подпрограмме, вывод результатов. Каждый из обрабатываемых массивов должен вводиться из отдельного текстового файла.

Этап 3. Выполняется преобразование основной программы и подпрограмм для общего случая, который предусматривает обработку массивов с разными именами типов.

В некоторых задачах в лабораторной работе № 5 нужно выполнять поиск серий в одномерном массиве (серия - это группа последовательно расположенных однотипных элементов). Рассмотрим методику такого поиска на конкретном примере.

Пример. В целочисленном массиве определить положение наиболее длинной серии отрицательных элементов, причем в состав серии должны входить не менее трех элементов.

Пусть нам задан массив $X = (x_1, x_2, \dots, x_n)$. Тогда признак начала серии - это истинность отношения $x_i < 0$, а признак конца серии - истинность отношения $x_i \geq 0$.

В программе GroupSearchе для искомой серии ставятся в соответствие две переменные: переменная k_1 , определяющая позицию первого элемента серии, и переменная k_2 , которая соответствует концу серии (позиции элемента, который расположен после последнего элемента серии). В этом случае длина серии (количество элементов, которые входят в ее состав) равна $k_2 - k_1$. Поиск начала очередной серии выполняется с позиции $k_2 + 1$, где k_2 определяет конечную границу предшествующей серии. Поиск конца текущей серии осуществляется, начиная с позиции $k_1 + 1$, где k_1 определяет начало данной серии.

В программе используются функции SignBegin и SignEnd, которые выполняют поиск начала и конца серии по соответствующим признакам. Если при обращении к функции поиска необходимый признак не найден, то выходное значение функции будет равно нулю.

Общий поиск осуществляется в цикле **While** под управлением булевской переменной Cond, которая имеет значение true до тех пор, пока продолжается поиск.

На старте общего поиска переменной k_2 присваивается нулевое значение, переменной Cond - значение true.

В начальной части цикла **While** определяется положение очередной серии: $k_1 := \text{SignBegin}(k_2 + 1)$. Если $k_1 = 0$, то это означает, что в массиве таких серий больше нет. Тогда переменной Cond присваивается значение false, что ведет к прекращению работы цикла **While**. Если $k_1 > 0$, то выполняется поиск конца серии: $k_2 := \text{SignEnd}(k_1 + 1)$. Если $k_2 = 0$, то это означает, что последний элемент массива принадлежит искомой серии. Тогда переменной k_2 присваивается значение $n + 1$, где n - количество элементов в массиве, а переменной Cond - значение false. После этого в соответствии с условием задачи выполняется обработка серии, ограниченной значениями переменных k_1 и k_2 .

В учебном пособии «Основы программирования на языке Турбо Паскаль» методика поиска серий рассматривается более подробно. В частности, здесь отмечено, что серии могут быть разделены на два типа:

1) такие, в которых начало и конец очередной серии однозначно определяются одним или двумя смежными элементами массива;

2) такие, в которых утверждение о наличии или отсутствии серии может быть сделано лишь в целом по отношению к фрагменту массива.

В вариантах заданий к лабораторной работе № 5 имеются задачи о поиске серий обоих типов.

```

Program GroupSearch;
Const Nmax = 500;
Type Ar = array[1..Nmax] of integer;
Var i, k1, k2,
    l,           { размер серии }
    lmax,       { размер наиболее длинной серии }
    kmax        { позиция нач.элемента наиболее }
                { длинной серии }
    : word;
    Cond : boolean; { управляющая переменная }
    X : Ar;         { обрабатываемый массив }
{ ----- }
Function SignBegin(k:word):word;
{ Поиск начала серии }
Var i : word;
Begin
    SignBegin:=0;
    For i:=k to n do
        If x[i]<0 then
            Begin
                SignBegin:=i; Exit
            End;
End { SignBegin };
{ ----- }
Function SignEnd(k:word):word;
{ Поиск конца серии }
Var i : word;
Begin
    SignEnd:=0;
    For i:=k to n do
        If x[i]>=0 then
            Begin
                SignEnd:=i; Exit
            End;
End { SignEnd };
{ ----- }
Begin
    Ввод n, X
    k2:=0; Cond:=true;
    lmax:=0; kmax:=0;
    While Cond do
        Begin
            k1:=SignBegin(k2+1);
            If k1=0 then
                Cond:=false
            Else

```

```

Begin
  k2:=SignEnd(k1+1);
  If k2=0 then
    Begin
      k2:=n+1;
      Cond:=false
    End;
  l:=k2-k1;
  If (l>2) and (l>lmax) then
    Begin
      lmax:=l; kmax:=k1
    End;
  End;
End;
Вывод  lmax, kmax
End.

```

В некоторых задачах нужно вычислять значения различных полиномов. Такие вычисления должны быть организованы по схеме Горнера, пример программной реализации которой содержится в методических указаниях к лабораторной работе № 3. В частности, при переводе числа из системы счисления с основанием q в десятичную систему также нужно применять схему Горнера (отдельно для целой и дробной частей числа).

Отчет по лабораторной работе № 5

В отчете по лабораторной работе необходимо привести:

- титульный лист;
- номер варианта и условие задачи;
- краткое описание программы;
- контрольный просчет (если результаты работы программы не являются очевидными);
- текст программы;
- результаты работы программы.

Текст программы должен содержать лишь третий этап решения задачи. В данном этапе рекомендуется использовать абсолютные переменные, которые в этом случае более универсальны по сравнению с аппаратом преобразования типов. Этапы 1 и 2, как записано в методических указаниях, рекомендуются для надежного перехода от сравнительно простой реализации к более сложной.

Пример выполнения задания

Ниже на конкретном примере приводится иллюстрация этапов выполнения лабораторной работы № 5.

Условие задачи. В вещественном массиве $X = (x_1, x_2, \dots, x_n)$ поменять местами первый положительный и последний отрицательный элементы, второй положительный и предпоследний отрицательный элементы и т.д.

Этап 1. Данный этап для рассматриваемого примера реализован в приведенной ниже программе Labor5a.

В программе Labor5a обрабатывается массив X , включающий в себя n элементов. Алгоритм решения задачи состоит в следующем.

Сначала для массива X определяется количество содержащихся в нем элементов: положительных `NumberPos` и отрицательных `NumberNeg`. Вполне очевидно, что количество циклов обмена `NumberExchange` будет равно $\min(\text{NumberPos}, \text{NumberNeg})$.

В цикле, который выполняется `NumberExchange` раз, массив X просматривается слева направо, начиная с позиции k_1 (начальное значение $k_1=1$) до тех пор, пока не будет найден положительный элемент. Положение этого элемента запоминается в переменной `IndexPos`. Аналогично выполняется просмотр массива X справа налево, начиная с позиции k_2 (начальное значение $k_2=n$), для обнаружения отрицательного элемента, положение которого запоминается в переменной `IndexNeg`. Обмен элементов с индексами `IndexPos` и `IndexNeg` выполняется в буферном массиве `Buffer`, который является копией исходного массива X . Перед переходом к следующему циклу поиска и обмена элементов значения k_1 и k_2 изменяются ($k_1=\text{IndexPos}+1$; $k_2=\text{IndexNeg}-1$). После завершения циклов обмена буферный массив `Buffer` переписывается в массив X .

Определение индексов положительного и отрицательного элементов выполняется в процедуре `SearchIndex`. Процедуры `WaitEnter`, `PrintString`, `PrintKeyAndWaitEnter`, `ControlPageScreen`, `ReadArray`, `ScreenArray` и `PrinterArray` являются инструментальными; выполняемая этими процедурами работа описана в комментариях к программе.

В программе `Labor5a` предусмотрено два вида печати: на экране дисплея и на принтере. Необходимость использования принтера при очередном запуске программы определяется ответом пользователя на запрос программы.

Процедура `ScreenArray` обеспечивает вывод на экран одномерного массива X любого размера. При этом в одной строке экрана печатается пять элементов массива X . Размер страницы экрана (количество строк на странице) определяется константой `LengthPage` процедуры `ControlPageScreen`. Переход к следующей странице экрана выполняется при нажатии клавиши `Enter`. При нажатии клавиши `Esc` происходит выход из процедуры, при этом оставшаяся часть массива на экран не выводится.

Примечание. Более совершенным способом просмотра больших массивов является способ, который используется в промышленных программах, где при нажатии клавиши \uparrow (Up) или \downarrow (Down) выполняется сдвиг экрана вверх или вниз на одну строку (скроллинг), а нажатие клавиши `PgUp` или `PgDn` вызывает перелистывание страниц на экране. Однако программы, которые реализуют такой способ просмотра, достаточно сложные и громоздкие, а потому в лабораторном практикуме по программированию не рассматриваются.

В описании этапа 3, которое приведено ниже на примере выполнения конкретного задания по лабораторной работе №5, говорится, в частности, о нежелательности применения буферных массивов при реализации задач по лабораторной работе. Тем не менее в программе `Labor5a` такой массив используется.

Предположим, что массив $X = (x_1, x_2, \dots, x_{12})$ содержит такие компоненты:

14 -3 -7 8 -12 -10 -5 7 9 12 15 18

Будем выполнять обмен элементов непосредственно в этом массиве, т.е. без применения буферного массива. В первом цикле, после обмена элементов $x_1 = 14$ и $x_7 = -5$, получим:

-5 -3 -7 8 -12 -10 14 7 9 12 15 18

Во втором цикле будут обменены элементы $x_4 = 8$ и $x_6 = -10$:

-5 -3 -7 -10 -12 8 14 7 9 12 15 18

В третьем цикле должны быть обменены элементы $x_6 = 8$ и $x_5 = -12$. Но элемент 8 уже принимал участие в обмене, изменение его положения приведет к неправильному преобразованию массива X . Поэтому в программе Labor5a в буферном массиве *Buffer* вначале создается копия исходного массива X , после чего при обработке исходного массива перестановка элементов производится лишь в буферном массиве. Это исключает возможность некорректного обмена типа того, что был отмечен в приведенном выше примере. После окончания обработки элементы буферного массива переписываются обратно в исходный массив.

Реализацию рассмотренного примера можно выполнить и без применения буферного массива, но для этого пришлось бы использовать стеки, которые рассматриваются лишь в лабораторной работе № 8.

```

Program Labor5a;
Uses Crt,Printer;
Const
    Nmax = 100;          { макс.количество эл-тов в массиве X }
    Enter = 13;         { код клавиши Enter }
    Escape = 27;       { код клавиши Esc }
    PressKey = 'Нажмите клавишу Enter';
Type
    Xar = array[1..Nmax] of real;
Var
    i,                  { параметр цикла }
    n,                  { количество эл-тов в массиве X }
    k1,                 { индекс начала поиска положит.эл-та }
    k2,                 { то же для отрицательного эл-та }
    IndexPos,           { позиция положительного элемента }
    IndexNeg,           { позиция отрицательного элемента }
    NumberPos,          { количество положительных элементов }
    NumberNeg,          { количество отрицательных элементов }
    NumberExchange      { количество обменов элементов }
        : integer;
    Buf : real;         { буферная переменная }
    IndPrinter : boolean; { индикатор использования принтера }
    Reply : char;       { символ ответа на запрос программы }
    X,Buffer : Xar;     { исходный и буферный массивы }
    FileX : text;       { исходный файл }
{ ----- }

Procedure WaitEnter;
{ Задержка выполнения программы, пока }
{ не будет нажата клавиша Enter }
Var ch : char;
Begin
    Repeat
        ch:=ReadKey;
    Until ord(ch) = Enter;
End { WaitEnter };
{ ----- }

Procedure PrintString(x,y:integer; S:string);
{ Вывод строки S с позиции x строки экрана с номером y }
Begin

```

```

    GotoXY(x,y);
    Write(S);
End { PrintString };
{ ----- }
Procedure PrintKeyAndWaitEnter;
{ Вывод строки-константы PressKey с позиции 1 строки }
{ экрана 25 и задержка программы до нажатия клавиши Enter }
Begin
    PrintString(1,25,PressKey);
    WaitEnter;
    ClrScr;
End { PrintKeyAndWaitEnter };
{ ----- }
Procedure ControlPageScreen(Var j,KeyExit:byte);
{ Контроль размера страницы на экране }
Const LengthPage = 23; { количество строк на одной странице }
    S = 'След.страница - Enter, конец просмотра - Esc';
Var ch : char;
Begin
    Inc(j); KeyExit:=0;
    If j=LengthPage then
        Begin
            j:=0;
            PrintString(1,25,S);
            Repeat
                ch:=ReadKey; KeyExit:=ord(ch);
            Until (KeyExit=Enter) or (KeyExit=Escape);
            ClrScr;
        End;
End { ControlPageScreen };
{ ----- }
Procedure ReadArray;
{ Ввод исходного массива }
Var i : integer;
Begin
    Reset(FileX);
    n:=0;
    While not SeekEof(FileX) do
        Begin
            Inc(n);
            Read(FileX,x[n]);
        End;
    Close(FileX);
End { ReadArray };
{ ----- }
Procedure ScreenArray(S:string);
{ Вывод на экран массива X }
Var i : integer;
    j,k,KeyExit : byte;
Begin
    j:=0; k:=0;
    Write(S); Writeln('      n = ',n);
    For i:=1 to n do

```

```

    Begin
      Inc(k);
      If k<5 then
        Write(x[i]:8:2, ' ':4)
      Else
        Begin
          k:=0; Inc(j);
          Writeln(x[i]:8:2);
          ControlPageScreen(j,KeyExit);
          If KeyExit=Escape then Exit;
        End;
      End;
      If k>0 then Writeln;
      PrintKeyAndWaitEnter;
End { ScreenArray };
{ ----- }
Procedure PrinterArray(S:string);
{ Печать на принтере массива X }
Var i : integer;
      k : byte;
Begin
  k:=0;
  Writeln(Lst);
  Write(Lst,S); Writeln(Lst, '   n = ',n);
  For i:=1 to n do
    Begin
      Inc(k);
      If k<5 then
        Write(Lst,x[i]:8:2, ' ':4)
      Else
        Begin
          k:=0;
          Writeln(Lst,x[i]:8:2);
        End;
    End;
  If k>0 then Writeln(Lst);
End { PrinterArray };
{ ----- }
Procedure SearchIndex;
{ Поиск в массиве X очередной пары положительного и отрица- }
{ тельного элементов, начиная с позиций k1 и k2           }
Label 10;
Var i : integer;
Begin
  IndexPos:=0; IndexNeg:=0;
  For i:=k1 to n do
    If x[i]>0 then
      Begin
        IndexPos:=i; Goto 10
      End;
  10:
  If IndexPos>0 then
    For i:=k2 downto 1 do

```



```

        If x[i]<0 then
            Begin
                IndexNeg:=i; Exit
            End;
End { SearchIndex };
{ ----- }
Begin

{ Установление соответствия между внутренним и }
{ внешним файлами }
Assign(FileX, 'X.dat');

{ Запрос об использовании принтера }
ClrScr; IndPrinter:=false;
Writeln('Будет ли использоваться принтер (Да,Нет) ?');
Reply:=ReadKey;
If Reply in ['Д','д','L','l'] then
    IndPrinter:=true;

{ Ввод и вывод исходных данных }
ReadArray;
ScreenArray('        Исходный массив X');
If IndPrinter then
    PrinterArray('        Исходный массив X');

{ Формирование буферного массива }
For i:=1 to n do
    buffer[i]:=x[i];

{ Определение количества обменов элементов в массиве }
NumberPos:=0; NumberNeg:=0;
For i:=1 to n do
    If x[i]>0 then
        Inc(NumberPos)
    Else
        If x[i]<0 then
            Inc(NumberNeg);
NumberExchange:=abs(NumberPos-NumberNeg);

{ Организация обменов элементов в массиве }
k1:=1; k2:=n;
For i:=1 to NumberExchange do
    Begin
        SearchIndex;
        Buf:=buffer[IndexPos];
        buffer[IndexPos]:=buffer[IndexNeg];
        buffer[IndexNeg]:=Buf;
        k1:=IndexPos+1; k2:=IndexNeg-1;
    End;
For i:=1 to n do
    x[i]:=buffer[i];

```

```

{ Вывод преобразованного массива }
ScreenArray('      Преобразованный массив X');
If IndPrinter then
    PrinterArray('      Преобразованный массив X');

```

End.

Этап 2. Преобразование программы Labor5a в подпрограмму в предположении, что все обрабатываемые массивы имеют одинаковое имя типа, в достаточной мере проиллюстрировано в программе Labor5b, которая приводится ниже. Здесь следует обратить внимание на следующее.

1. Ввод каждого из обрабатываемых массивов X, Y, Z выполняется из отдельного файла.

2. Ввод и печать обрабатываемых массивов реализуется в основной программе при обращении к процедурам ReadArray, ScreenArray и PrinterArray.

3. Все инструментальные подпрограммы независимы от процедуры Exchange и являются внутренними только по отношению к основной программе.

4. При обработке любого из массивов X, Y, Z по описанному выше алгоритму различие между этими массивами состоит лишь в имени и размере массива, в связи с чем для процедуры Exchange в качестве формальных параметров взяты имя обрабатываемого массива A и его размер n.

5. Описания переменных, которые содержались в разделе **Var** программы Labor5a, разделены на глобальные (раздел **Var** программы Labor5b) и локальные (раздел **Var** процедуры Exchange). К локальным переменным отнесены те переменные, которые используются только в процедуре обмена элементов обрабатываемого массива. В частности, для локального массива Buffer, как и для других локальных переменных, память выделяется лишь на период работы процедуры Exchange.

6. В процедуре ReadArray возвращаемыми параметрами являются имя массива A и его размер n. Поэтому перед именем этих формальных параметров записано слово **Var**.

7. При работе процедуры ScreenArray элементы формального массива A и его размер n не изменяются. Поэтому оба эти параметра можно было бы объявить как параметры-значения. Однако в этом случае при обращении к процедуре была бы произведена передача всех элементов фактического массива, в чем нет необходимости. Наличие слова **Var** только перед именем формального массива A приводит к тому, что в процедуру ScreenArray передаются не элементы массива, а лишь его адрес, имеющий размер 4 байта.

Program Labor5b;

Uses Crt, Printer;

Const

Nmax = 300; { макс. количество эл-тов в массивах X, Y, Z }

Enter = 13; { код клавиши Enter }

Escape = 27; { код клавиши Esc }

PressKey = 'Нажмите клавишу Enter';

Type

XYZar = **array**[1..Nmax] **of** real;

Var

nx, ny, nz: integer; { количество эл-тов в массивах X, Y, Z }

```

    IndPrinter          { индикатор использования принтера }
        : boolean;
    Reply : char;        { символ ответа на запрос программы }
    X,Y,Z : XYZar;      { исходные массивы X,Y,Z }
    FileX,FileY,        { исходные файлы }
    FileZ : text;
{ ----- }
Procedure WaitEnter;
Procedure PrintString(x,y:integer; S:string);
Procedure PrintKeyAndWaitEnter;
Procedure ControlPageScreen(Var j,KeyExit:byte);
{ ----- }
Procedure ReadArray(Var FileArray:text; Var A:XYZar;
                    Var n:integer);
{ Ввод исходного массива }
Var i : integer;
Begin
    Reset(FileArray);
    n:=0;
    While not SeekEof(FileArray) do
        Begin
            Inc(n);
            Read(FileArray,a[n]);
        End;
    Close(FileArray);
End { ReadArray };
{ ----- }
Procedure ScreenArray(Var A:XYZar; n:integer; S:string);
{ Вывод исходного массива на экран }
Var i : integer;
    j,k,KeyExit : byte;
Begin
    ClrScr; j:=0; k:=0;
    Writeln(S);
    For i:=1 to n do
        Begin
            Inc(k);
            If k<5 then
                Write(a[i]:8:2,' ':4)
            Else
                Begin
                    k:=0; Inc(j);
                    Writeln(a[i]:8:2);
                    ControlPageScreen(j,KeyExit);
                    If KeyExit=Escape then Exit;
                End;
            End;
        If k>0 then Writeln;
        PrintKeyAndWaitEnter;
    End { ScreenArray };
{ ----- }
Procedure PrinterArray(Var A:XYZar; n:integer; S:string);
{ Печать исходного массива на принтере }

```

```

Var i : integer;
      k : byte;
Begin
  k:=0;
  Writeln(Lst); Writeln(Lst,S);
  For i:=1 to n do
    Begin
      Inc(k);
      If k<5 then
        Write(Lst,a[i]:8:2,' ':4)
      Else
        Begin
          k:=0;
          Writeln(Lst,a[i]:8:2);
        End;
    End;
  If k>0 then Writeln(Lst);
End { PrinterArray };
{ ----- }
Procedure Exchange(Var A:XYZar; n:integer);
{ Обмен положительных и отрицательных элементов в массиве }
Var
  i,           { параметр цикла }
  k1,         { индекс начала поиска положит. эл-та }
  k2,         { то же для отрицательного элемента }
  IndexPos,   { позиция положительного элемента }
  IndexNeg,   { позиция отрицательного элемента }
  NumberPos,  { количество положительных элементов }
  NumberNeg,  { количество отрицательных элементов }
  NumberExchange { количество обменов элементов }
  : integer;
  Buf : real;   { буферная переменная }
  Buffer : XYZar; { буферный массив }
{ ----- }
Procedure SearchIndex;
{ Поиск положительного и отрицательного элементов }
Label 10;
Var i : integer;
Begin
  IndexPos:=0; IndexNeg:=0;
  For i:=k1 to n do
    If a[i]>0 then
      Begin
        IndexPos:=i; Goto 10
      End;
  10:
  If IndexPos>0 then
    For i:=k2 downto 1 do
      If a[i]<0 then
        Begin
          IndexNeg:=i; Exit
        End;
  End { SearchIndex };

```

```

{ ----- }
Begin

{ Формирование буферного массива }
For i:=1 to n do
    buffer[i]:=a[i];

{ Определение количества обменов в массиве }
NumberPos:=0; NumberNeg:=0;
For i:=1 to n do
    If a[i]>0 then
        Inc(NumberPos)
    Else
        If a[i]<0 then
            Inc(NumberNeg);
NumberExchange:=abs(NumberPos-NumberNeg);

{ Организация обменов элементов в массиве }
k1:=1; k2:=n;
For i:=1 to NumberExchange do
    Begin
        SearchIndex;
        Buf:=buffer[IndexPos];
        buffer[IndexPos]:=buffer[IndexNeg];
        buffer[IndexNeg]:=Buf;
        k1:=IndexPos+1; k2:=IndexNeg-1;
    End;
For i:=1 to n do
    a[i]:=buffer[i];
End { Exchange };
{ ----- }
Begin

{ Установление соответствия между внутренними и }
{ внешними файлами }
Assign(FileX,'X.dat');
Assign(FileY,'Y.dat');
Assign(FileZ,'Z.dat');

{ Запрос об использовании принтера }
ClrScr; IndPrinter:=false;
Writeln('Будет ли использоваться принтер (Да,Нет) ?');
Reply:=ReadKey;
If Reply in ['Д','д','L','l'] then
    IndPrinter:=true;

{ Ввод, обработка и вывод массива X }
ReadArray(FileX,X,nx);
ScreenArray(X,nx,'          Исходный массив X');
If IndPrinter then
    PrinterArray(X,nx,'          Исходный массив X');
Exchange(X,nx);
ScreenArray(X,nx,'          Преобразованный массив X');

```

```

If IndPrinter then
  PrinterArray(X,nx, '      Преобразованный массив X');

{ Ввод, обработка и вывод массива Y }
ReadArray(FileY,Y,ny);
ScreenArray(Y,ny, '      Исходный массив Y');
If IndPrinter then
  PrinterArray(Y,ny, '      Исходный массив Y');
Exchange (Y,ny);
ScreenArray(Y,ny, '      Преобразованный массив Y');
If IndPrinter then
  PrinterArray(Y,ny, '      Преобразованный массив Y');

{ Ввод, обработка и вывод массива Z }
ReadArray(FileZ,Z,nz);
ScreenArray(Z,nz, '      Исходный массив Z');
If IndPrinter then
  PrinterArray(Z,nz, '      Исходный массив Z');
Exchange (Z,nz);
ScreenArray(Z,nz, '      Преобразованный массив Z');
If IndPrinter then
  PrinterArray(Z,nz, '      Преобразованный массив Z');
End.

```

Замечание 3. На данном этапе выполняется преобразование программы Labor5b в предположении, что обрабатываемые массивы X, Y, Z имеют разные имена типов.

Практически для каждого языка программирования создаются пакеты прикладных программ (ППП), которые реализуют разнообразные процедуры численного анализа (решение нелинейных уравнений, определение корней полиномов, операции над матрицами, решение системы дифференциальных уравнений и т.п.). Прикладная программа разрабатывается в виде подпрограммы, обращение к которой выполняется из программы пользователя. При этом, как правило, ППП поставляется в виде объектных модулей, что исключает возможность изменения их текста пользователем.

Паскаль-подпрограмма, которая обрабатывает массив, должна в списке формальных параметров содержать имя этого массива с указанием соответствующего имени типа. В общем случае имя типа формального массива не совпадает с именем типа фактического массива в программе пользователя. Таким образом, прикладная подпрограмма должна обеспечивать совместимость типов формального и фактического массивов.

В Паскаль-программе массив всегда имеет фиксированный размер, обусловленный его именем типа. Необходимость использования разных имен типов связана главным образом с тем, что формальный и фактический массивы в общем случае имеют разные размеры. Типы элементов этих массивов, естественно, должны быть одинаковыми.

Предположим, что для массивов X, Y и Z, которые имеют разные имена типов, необходимо вычислить среднее арифметическое их элементов. Тогда программа может иметь следующий вид:

```

Program Example;
Type  Xar = array[1..50] of real;
        Yar = array[1..500] of real;

```

```

        Zar = array[1..5000] of real;
Var i,nx,ny,nz : word;
        Sx,Sy,Sz : real;
        X : Xar;
        Y : Yar;
        Z : Zar;
Procedure MiddleAr(Var Buf:Xar; Var S:real; n:word);
Var i : word;
Begin
    S:=0;
    For i:=1 to n do
        S:=S+Buf[i];
    S:=S/n;
End { MiddleAr };
Begin
    Ввод и печать nx,ny,nz,X,Y,Z
    MiddleAr(X,Sx,nx);
    MiddleAr(Y,Sy,ny);
    MiddleAr(Z,Sz,nz);
    Печать Sx,Sy,Sz
End.

```

Здесь при трансляции программы будет правильно воспринято лишь первое обращение к процедуре MiddleAr; для других обращений будет выдано сообщение о несоответствии типов фактических массивов Y, Z и формального массива Buf.

Возможность обработки массивов с разными именами типов может быть обеспечена одним из двух способов: путем использования аппарата приведения типов переменных или с помощью абсолютных переменных.

Рассмотрим суть приведения типов на таких двух примерах.

Пример 1.

```

Program ByteChar;
Var ch : char;
    b1,b2 : byte;
Begin
    ch:='A'; b1:=byte(ch); b2:=ord(ch);
    Writeln('b1=',b1,' b2=',b2);
End.

```

Результаты работы программы: b1=65 b2=65.

Выражение byte(ch) определяет "наложение" типа byte на переменную ch. В этом случае биты, которые содержатся в поле памяти ch, интерпретируются как значение переменной типа byte (ch = 01000001 (двоичное) = 41 (шестнадцатеричное) = 65 (десятичное)).

Пример 2.

```

Program PutType;
Type ByteAr = array[1..2] of byte;
Var W : word;
Begin
    ByteAr(W) [1]:=10; ByteAr(W) [2]:=20;

```

```

        Writeln(W, '      ', ByteAr(W) [1], '      ', ByteAr(W) [2]);
        W:=2500;
        Writeln(W, '      ', ByteAr(W) [1], '      ', ByteAr(W) [2]);
        Readln;
End.

```

В результате работы программы будет напечатано:

```

        5130  10    20
        2500  196   9

```

В программе PutType происходит наложение "формального" поля ByteAr длиной два байта на реальное поле W, которое также имеет длину два байта. При этом байты, входящие в состав поля W, рассматриваются как отдельные элементы байтового массива типа ByteAr.

Напечатанные выше элементы в 16 с/с имеют следующие значения:

```

        140A    OA    14
        09C4    C4    09

```

Соответствие между элементами полей W и ByteAr легко увидеть, если учесть, что в переменной типа word старшим считается правый байт.

В Турбо Паскале допускаются формальные параметры без типа. Таким формальным параметрам могут соответствовать фактические параметры любого типа. Это позволяет с помощью аппарата приведения типа переменной организовать в процедуре обработку массивов разного размера. В частности, программа Example в этом случае будет иметь следующий вид:

```

Program Example1;
Type  Xar = array[1..50] of real;
        Yar = array[1..500] of real;
        Zar = array[1..5000] of real;
Var i, nx, ny, nz : word;
        Sx, Sy, Sz : real;
        X : Xar;
        Y : Yar;
        Z : Zar;

Procedure MiddleAr(Var Buf; Var S:real; n:word);
Type  RealAr = array[1..10000] of real;
Var i : word;
Begin
        S:=0;
        For i:=1 to n do
            S:=S+RealAr(Buf) [i];
        S:=S/n;
End { MiddleAr };
Begin
        Ввод и печать nx, ny, nz, X, Y, Z
        MiddleAr(X, Sx, nx);
        MiddleAr(Y, Sy, ny);
        MiddleAr(Z, Sz, nz);
        Печать Sx, Sy, Sz
End.

```


Формальный параметр Buf - это параметр-переменная, поскольку перед именем Buf расположено слово **Var**. Это означает, что при обращении к процедуре MiddleAr фиктивный адрес переменной Buf замещается реальным адресом переменной X, Y или Z. Как известно, адрес поля памяти - это адрес его крайнего левого байта. Однако знания адреса поля X в программе еще недостаточно для его обработки, поскольку при этом неизвестна ни длина, ни структура поля памяти. Необходимую дополнительную информацию предоставляет конструкция RealAr(Buf), которая в данном случае указывает, что поле памяти, которое соответствует формальной переменной Buf, надо обрабатывать как переменную типа RealAr, то есть как одномерный массив с вещественными компонентами.

Хотя "интерпретирующее" поле RealAr, которое накладывается на формальную переменную Buf без типа, больше чем фактические переменные X, Y или Z, в процедуре MiddleAr выход за границы соответствующих массивов не состоится, если переменные nx, ny, nz не превышают соответственно значений 50, 500, 5000.

Вышеприведенную программу Example можно не менее эффективно реализовать с помощью абсолютных переменных:

```

Program Example2;
Type Xar = array[1..50] of real;
      Yar = array[1..500] of real;
      Zar = array[1..5000] of real;
Var i, nx, ny, nz : word;
      Sx, Sy, Sz : real;
      X : Xar;
      Y : Yar;
      Z : Zar;
Procedure MiddleAr(Var Buf; Var S:real; n:word);
Type RealAr = array[1..10000] of real;
Var i : word;
      A : RealAr absolute Buf;
Begin
  S:=0;
  For i:=1 to n do
    S:=S+a[i];
  S:=S/n;
End { MiddleAr };
Begin
  Ввод и печать nx, ny, nz, X, Y, Z
  MiddleAr(X, Sx, nx);
  MiddleAr(Y, Sy, ny);
  MiddleAr(Z, Sz, nz);
  Печать Sx, Sy, Sz
End.

```

В разделе **Var** процедуры MiddleAr объявлена переменная A типа RealAr. Тем не менее этой переменной, в отличие от других локальных переменных, память не выделяется. Ей присваивается тот же адрес, который получает формальная переменная Buf при обращении к процедуре MiddleAr. Следовательно, при первом обращении к этой процедуре имена A, Buf, X - это три имени одного и того же поля памяти, а

поскольку тип переменной A указан явно, то программа располагает всей необходимой информацией для обработки поля X, непосредственно используя для этого имя A.

В программе Labor5c, которая реализует обработку массивов с разными именами типов, в процедурах ReadArray, ScreenArray и других на формальный параметр A без имени типа накладывается массив с типом RealAr. При обращении к процедуре формальное имя A заменяется именем фактического массива. Так как RealAr - это имя типа, а не описание переменной, то объекту с именем RealAr никакой памяти не выделяется. Как и любое имя типа, RealAr определяет множество значений, которые имеет право принимать переменная с этим именем типа. Поскольку массив не может занимать более чем 64 Кбайт (65536 байт) памяти, то для типа элемента real максимальный размер массива может иметь описание

```
RealAr = array[1..10922] of real или  
RealAr = array[1..2*MaxInt div SizeOf(real)] of real.
```

В программе Labor5d аналогичная обработка массивов с разными именами типа выполняется с помощью аппарата приведения типа.

Итак, в процедуре Exchange выполняется обработка одномерного вещественного массива с любым именем типа, что обеспечивается интерпретацией формального параметра A как переменной типа RealAr. Несколько иная ситуация с буферным массивом Buffer, в который "перекачиваются" элементы обрабатываемого массива. Массив Buffer является локальным. При старте процедуры Exchange этому массиву должно выделяться вполне определенное поле памяти. Поэтому естественно возникает вопрос: "Сколько памяти выделять для массива Buffer?" Поскольку по определению процедура Exchange должна обрабатывать массивы с любым именем типа (массивы любого размера), то на этот вопрос не может быть однозначного ответа. При любом конкретном размере массива Buffer выделяемой ему памяти может не хватить при обработке большого массива; и наоборот, эта память будет избыточной при обработке небольших массивов. Следовательно, использование буферных массивов существенным образом ограничивает универсальность процедур, предназначенных для обработки массивов с разными именами типов.

Процедуру Exchange можно реализовать без применения буферного массива, для чего можно использовать стек, в который последовательно записываются индексы обмениваемых элементов массива. Стек - это динамическая структура, для которой память выделяется в процессе работы программы по мере необходимости, поэтому при его использовании не нарушается универсальность процедуры Exchange. Но поскольку стеки рассматриваются лишь в лабораторной работе № 8, то при выполнении работы № 5 допускается применение буферных массивов, если в конкретной задаче нет фразы "Буферный массив не использовать".

```
Program Labor5c;  
{ Этап 3 с использованием абсолютных переменных }  
Uses Crt,Printer;  
Const  
  NmaxX = 100;   { макс.количество эл-тов в массиве X }  
  NmaxY = 200;   { макс.количество эл-тов в массиве Y }  
  NmaxZ = 300;   { макс.количество эл-тов в массиве Z }  
  Enter = 13;    { код клавиши Enter }  
  Escape = 27;   { код клавиши Esc }
```

```

    PressKey = 'Нажмите клавишу Enter';
Type
    Xar = array[1..NmaxX] of real;
    Yar = array[1..NmaxY] of real;
    Zar = array[1..NmaxZ] of real;
Var
    nx,ny,nz : integer; { количество эл-тов в массивах X,Y,Z }
    IndPrinter      { индикатор использования принтера }
        : boolean;
    Reply : char;      { символ ответа на запрос программы }
    X : Xar;           { ИСХОДНЫЙ МАССИВ X }
    Y : Yar;           { ИСХОДНЫЙ МАССИВ Y }
    Z : Zar;           { ИСХОДНЫЙ МАССИВ Z }
    FileX,FileY,     { ИСХОДНЫЕ ФАЙЛЫ }
    FileZ : text;
{ ----- }
Procedure WaitEnter;
Procedure PrintString(x,y:integer; S:string);
Procedure PrintKeyAndWaitEnter;
Procedure ControlPageScreen(Var j,KeyExit:byte);
{ ----- }
Procedure ReadArray(Var FileArray:text; Var A; Var n:integer);
{ Ввод исходного массива }
Type RealAr = array[1..2*MaxInt div SizeOf(real)] of real;
Var i : integer;
    B : RealAr absolute A;
Begin
    Reset(FileArray);
    n:=0;
    While not SeekEof(FileArray) do
        Begin
            Inc(n);
            Read(FileArray,b[n]);
        End;
    Close(FileArray);
End { ReadArray };
{ ----- }
Procedure ScreenArray(Var A; n:integer; S:string);
{ Вывод исходного массива на экран }
Type RealAr = array[1..2*MaxInt div SizeOf(real)] of real;
Var i : integer;
    j,k,KeyExit : byte;
    B : RealAr absolute A;
Begin
    ClrScr; j:=0; k:=0;
    Writeln(S);
    For i:=1 to n do
        Begin
            Inc(k);
            If k<5 then
                Write(b[i]:8:2,' ':4)
            Else
                Begin

```

```

        k:=0; Inc(j);
        Writeln(b[i]:8:2);
        ControlPageScreen(j,KeyExit);
        If KeyExit=Escape then Exit;
    End;
End;
If k>0 then Writeln;
PrintKeyAndWaitEnter;
End { ScreenArray };
{ ----- }
Procedure PrinterArray(Var A; n:integer; S:string);
{ Печать исходного массива на принтере }
Type RealAr = array[1..2*MaxInt div SizeOf(real)] of real;
Var i : integer;
    k : byte;
    B : RealAr absolute A;
Begin
    k:=0;
    Writeln(Lst); Writeln(Lst,S);
    For i:=1 to n do
        Begin
            Inc(k);
            If k<5 then
                Write(Lst,b[i]:8:2,' ':4)
            Else
                Begin
                    k:=0;
                    Writeln(Lst,b[i]:8:2);
                End;
            End;
        End;
    If k>0 then Writeln(Lst);
End { PrinterArray };
{ ----- }
Procedure Exchange(Var A; n:integer);
{ Обмен положительных и отрицательных элементов в массиве }
Type RealAr = array[1..2*MaxInt div SizeOf(real)] of real;
    BufAr = array[1..1000] of real;
Var
    i,           { параметр цикла }
    k1,         { индекс начала поиска положит.эл-та }
    k2,         { то же для отрицательного элемента }
    IndexPos,   { позиция положительного элемента }
    IndexNeg,   { позиция отрицательного элемента }
    NumberPos,  { количество положительных элементов }
    NumberNeg,  { количество отрицательных элементов }
    NumberExchange { количество обменов элементов }
    : integer;
    Buf : real; { буферная переменная }
    Buffer : BufAr; { буферный массив }
    B : RealAr absolute A;
{ ----- }
Procedure SearchIndex;
{ Поиск положительного и отрицательного элементов }

```

```

Label 10;
Var i : integer;
Begin
  IndexPos:=0; IndexNeg:=0;
  For i:=k1 to n do
    If b[i]>0 then
      Begin
        IndexPos:=i; Goto 10
      End;
  10:
  If IndexPos>0 then
    For i:=k2 downto 1 do
      If b[i]<0 then
        Begin
          IndexNeg:=i; Exit
        End;
  End { SearchIndex };
  { ----- }
Begin

  { Формирование буферного массива }
  For i:=1 to n do
    buffer[i]:=b[i];

  { Определение количества обменов в массиве }
  NumberPos:=0; NumberNeg:=0;
  For i:=1 to n do
    If b[i]>0 then
      Inc(NumberPos)
    Else
      If b[i]<0 then
        Inc(NumberNeg);
  NumberExchange:=abs (NumberPos-NumberNeg);

  { Организация обменов элементов в массиве }
  k1:=1; k2:=n;
  For i:=1 to NumberExchange do
    Begin
      SearchIndex;
      Buf:=buffer[IndexPos];
      buffer[IndexPos]:=buffer[IndexNeg];
      buffer[IndexNeg]:=Buf;
      k1:=IndexPos+1; k2:=IndexNeg-1;
    End;
  For i:=1 to n do
    b[i]:=buffer[i];
  End { Exchange };
  { ----- }
Begin
  Раздел операторов в программах Labor5b и Labor5c одинаковый
End.

```

```

Program Labor5d;
{ Этап 3 с использованием аппарата приведения типов }
Uses Crt,Printer;
Const
    NmaxX = 100;    { макс.количество эл-тов в массиве X }
    NmaxY = 200;    { макс.количество эл-тов в массиве Y }
    NmaxZ = 300;    { макс.количество эл-тов в массиве Z }
    Enter = 13;     { код клавиши Enter }
    Escape = 27;    { код клавиши Esc }
    PressKey = 'Нажмите клавишу Enter';
Type
    Xar = array[1..NmaxX] of real;
    Yar = array[1..NmaxY] of real;
    Zar = array[1..NmaxZ] of real;
Var
    nx,ny,nz : integer; { кол-во эл-тов в массивах X,Y,Z }
    IndPrinter
        : boolean;
    Reply : char;       { символ ответа на запрос программы }
    X : Xar;            { ИСХОДНЫЙ МАССИВ X }
    Y : Yar;            { ИСХОДНЫЙ МАССИВ Y }
    Z : Zar;            { ИСХОДНЫЙ МАССИВ Z }
    FileX,FileY,
    FileZ : text;
{ ----- }
Procedure WaitEnter;
Procedure PrintString(x,y:integer; S:string);
Procedure PrintKeyAndWaitEnter;
Procedure ControlPageScreen(Var j,KeyExit:byte);
{ ----- }
Procedure ReadArray(Var FileArray:text; Var A; Var n:integer);
{ Ввод исходного массива }
Type RealAr = array[1..2*MaxInt div SizeOf(real)] of real;
Var i : integer;
Begin
    Reset(FileArray);
    n:=0;
    While not SeekEof(FileArray) do
        Begin
            Inc(n);
            Read(FileArray,RealAr(A)[n]);
        End;
    Close(FileArray);
End { ReadArray };
{ ----- }
Procedure ScreenArray(Var A; n:integer; S:string);
{ Вывод исходного массива на экран }
Type RealAr = array[1..2*MaxInt div SizeOf(real)] of real;
Var i : integer;
    j,k,KeyExit : byte;
Begin
    ClrScr; j:=0; k:=0;
    Writeln(S);

```

```

For i:=1 to n do
  Begin
    Inc(k);
    If k<5 then
      Write(RealAr(A)[i]:8:2, ' ':4)
    Else
      Begin
        k:=0; Inc(j);
        Writeln(RealAr(A)[i]:8:2);
        ControlPageScreen(j,KeyExit);
        If KeyExit=Escape then Exit;
      End;
    End;
    If k>0 then Writeln;
    PrintKeyAndWaitEnter;
End { ScreenArray };
{ ----- }
Procedure PrinterArray(Var A; n:integer; S:string);
{ Печать исходного массива на принтере }
Type RealAr = array[1..2*MaxInt div SizeOf(real)] of real;
Var i : integer;
    k : byte;
Begin
  k:=0;
  Writeln(Lst); Writeln(Lst,S);
  For i:=1 to n do
    Begin
      Inc(k);
      If k<5 then
        Write(Lst,RealAr(A)[i]:8:2, ' ':4)
      Else
        Begin
          k:=0;
          Writeln(Lst,RealAr(A)[i]:8:2);
        End;
    End;
    If k>0 then Writeln(Lst);
End { PrinterArray };
{ ----- }
Procedure Exchange(Var A; n:integer);
{ Обмен положительных и отрицательных элементов в массиве }
Type RealAr = array[1..2*MaxInt div SizeOf(real)] of real;
    BufAr = array[1..1000] of real;
Var
  i,          { параметр цикла }
  k1,        { индекс начала поиска положит. эл-та }
  k2,        { то же для отрицательного элемента }
  IndexPos,  { позиция положительного элемента }
  IndexNeg,  { позиция отрицательного элемента }
  NumberPos, { количество положительных элементов }
  NumberNeg, { количество отрицательных элементов }
  NumberExchange { количество обменов элементов }
  : integer;

```

```

    Buf : real;      { буферная переменная }
    Buffer : BufAr; { буферный массив }
{ ----- }
Procedure SearchIndex;
{ Поиск положительного и отрицательного элементов }
Label 10;
Var i : integer;
Begin
    IndexPos:=0; IndexNeg:=0;
    For i:=k1 to n do
        If RealAr(A)[i]>0 then
            Begin
                IndexPos:=i; Goto 10
            End;
    10:
    If IndexPos>0 then
        For i:=k2 downto 1 do
            If RealAr(A)[i]<0 then
                Begin
                    IndexNeg:=i; Exit
                End;
    End { SearchIndex };
{ ----- }
Begin

{ Формирование буферного массива }
    For i:=1 to n do
        buffer[i]:=RealAr(A)[i];

{ Определение количества обменов в массиве }
    NumberPos:=0; NumberNeg:=0;
    For i:=1 to n do
        If RealAr(A)[i]>0 then
            Inc(NumberPos)
        Else
            If RealAr(A)[i]<0 then
                Inc(NumberNeg);
    NumberExchange:=abs(NumberPos-NumberNeg);

{ Организация обменов элементов в массиве }
    k1:=1; k2:=n;
    For i:=1 to NumberExchange do
        Begin
            SearchIndex;
            Buf:=buffer[IndexPos];
            buffer[IndexPos]:=buffer[IndexNeg];
            buffer[IndexNeg]:=Buf;
            k1:=IndexPos+1; k2:=IndexNeg-1;
        End;
    For i:=1 to n do
        RealAr(A)[i]:=buffer[i];
    End { Exchange };
{ ----- }

```


Begin

Раздел операторов в программах Labor5b и Labor5d одинаковый

End.**Результаты работы программы:**

```

Исходный массив X      n = 25
  1.00      2.00     -3.00      0.00      5.00
-11.00     12.00      0.00     -14.00     15.00
 21.00    -22.00     23.00    -24.00     25.00
 18.00    -44.00    -65.00      0.00      0.00
  0.00     17.00    -33.00    -48.00     22.00

Преобразованный массив X
-48.00    -33.00     18.00      0.00    -65.00
 25.00    -44.00      0.00     23.00    -24.00
-22.00     21.00    -14.00     15.00    -11.00
 -3.00     12.00      5.00      0.00      0.00
  0.00     17.00      2.00      1.00     22.00

Исходный массив Y      n = 15
 31.00  -222.00      73.00     64.00    -85.00
-11.00   12.00    -13.00     14.00     15.00
 21.00   -22.00     23.00    -24.00     25.00

Преобразованный массив Y
-24.00   15.00    -22.00    -13.00     14.00
 12.00   -11.00     64.00    -85.00   -222.00
 21.00   73.00     23.00     31.00     25.00

Исходный массив Z      n = 20
441.00   332.00     83.00     74.00     55.00
  0.00     1.00     12.00     13.00     14.00
  1.00     0.00      5.00     21.00     22.00
273.00   24.00      0.00      0.00     25.00

Преобразованный массив Z
441.00   332.00     83.00     74.00     55.00
  0.00    11.00     12.00     13.00     14.00
  1.00     0.00      5.00     21.00     22.00
273.00   24.00      0.00      0.00     25.00

```

Варианты заданий

1. Задан массив целых чисел $X = (x_1, x_2, \dots, x_n)$. Сформировать массив $Y = (y_1, y_2, \dots, y_m)$, поместив в него в порядке уменьшения все разные числа, которые содержатся в массиве X . Определить, насколько отличаются средние арифметические значения элементов массивов X и Y .

Примечание. “Разные” и “неповторяемые” числа – это не одно и то же. Например, в массиве $X = (5, 8, 18, -3, 10, 14, 14, 8, 12, 7, 5, 12, 8, 4, 12)$ разные числа – это массив $Y = (18, 14, 12, 10, 8, 7, 5, 4, -3)$, а неповторяемые – массив $Z = (18, 10, 7, 4, -3)$.

Рекомендация.

- 1) Записать $y_1 = x_1; m = 1$.
- 2) Просматривая элементы $x_i (i = 2 .. n)$, определить, имеется ли элемент x_i в массиве Y . Если такой элемент не обнаружен, то добавить его в массив Y , увеличивая при этом значение переменной m .

2. Массив $X = (x_1, x_2, \dots, x_n)$ содержит большое количество нулевых элементов. Определить положение и размер наиболее длинной серии таких элементов и удалить ее из состава массива.

Примечание. Алгоритм поиска серии в одномерном массиве рассмотрен в методических указаниях к лабораторной работе № 5.

3. Задано два массива $X = (x_1, x_2, \dots, x_n)$ и $Y = (y_1, y_2, \dots, y_m)$, в состав которых входят натуральные числа, причем в каждом из этих массивов нет повторяющихся элементов. Сформировать массив Z , включив в него все элементы, которые одновременно содержатся в массивах X и Y .

Рекомендация.

- 1) В массив Z полностью переписать массив X .
- 2) Перебирая элементы массива Y , определить, имеется ли элемент y_i в массиве X , и, если такой элемент не обнаружен, добавить его в массив Z .

4. Задан целочисленный массив $X = (x_1, x_2, \dots, x_n)$, в котором могут быть одинаковые числа. Найти максимальный и минимальный элементы среди неповторяющихся чисел и обменять их местами. Учесть частные случаи:

- в массиве нет неповторяющихся чисел;
- массив содержит лишь одно неповторяющееся число.

5. Из массива целых положительных чисел $X = (x_1, x_2, \dots, x_n)$ удалить все четные по значению элементы, кроме последнего. Числа, которые остались после удаления четных элементов, расположить в порядке возрастания. Учесть частные случаи (в массиве нет четных элементов, есть только один четный элемент, все элементы - четные). Буферный массив не использовать.

Примечание. Массив X рекомендуется просматривать справа налево. Тогда первый встреченный четный элемент не будет подлежать удалению, остальные четные элементы должны быть удалены.

6. Задан целочисленный массив $X = (x_1, x_2, \dots, x_n)$. Если в этом массиве имеются пары одинаковых, но противоположных по знаку элементов, то обменять их местами при условии, что в массиве больше нет равных им элементов.

Например, в массиве

$$X = (1, 8, 14, 12, -8, -7, 6, 8, -14, -7, 12, 4)$$

нужно обменять местами элементы 14 и -14.

Рекомендация.

Пусть обрабатываемый массив имеет вид

$$X = (5, 8, 3, -5, 7, -8, 4, 8, 6, -12)$$

Предположим, что для элемента x_i проверяются элементы $x_j, j = i+1..n$, т.е. расположенные после него элементы массива X . Тогда при анализе $x_1 = 5$ однозначно вытекает, что должны быть обменены элементы x_1 и x_4 . При анализе $x_2 = 8$ обмен не производится, поскольку в остальной части массива имеются еще два элемента, равные по модулю x_2 . Однако при анализе $x_6 = -8$ будет произведен обмен, что не соответствует условию задачи.

Следовательно, при анализе элемента x_i нужно учитывать не только элементы, которые находятся после него, но все элементы массива. Тогда ситуация, указанная для $x_6 = -8$, не будет возникать. Однако при этом появляется другой источник ошибки. После обмена x_1 и x_4 при анализе x_4 будет повторно произведен обмен этих же элементов, что, естественно, неверно.

Предлагается следующий алгоритм решения задачи.

- 1) Создать булевский индексный массив *Index*, равный по размеру массиву X .
- 2) Всем элементам массива *Index* присвоить значение *false*.
- 3) Оформить функцию. `Exchange(k:integer):boolean`, где k – индекс анализируемого элемента массива. Эта функция проверяет, сколько в массиве имеется элементов, равных по модулю x_k . Если найден лишь один $|x_p| = |x_k|$ и при этом `Index[p] = false`, то производится обмен x_k и x_p , после чего устанавливается `Index[p] := true` и `Index[k] := true`.

7. В массиве $X = (x_1, x_2, \dots, x_n)$ переставить местами первый и второй отрицательные элементы, третий и четвертый отрицательные элементы и т.д. Если количество отрицательных элементов в массиве меньше двух, преобразование массива не производить. Определить, как изменилось положение минимального и максимального элементов массива X при его преобразовании.

Рекомендация.

Здесь целесообразно создать процедуру `Searche (Var k, k1, k2: integer)`, где k – индекс начала поиска отрицательного элемента, $k1$ и $k2$ – индексы двух ближайших отрицательных элементов. В начальной части процедуры `Searche` устанавливаются $k1 = 0$ и $k2 = 0$. Обмен элементов массива X производится, если $k1 \neq 0$ и $k2 \neq 0$. Если же $k1 = 0$ или $k2 = 0$, то дальнейший анализ массива X прекращается.

8. Задан вещественный массив $X = (x_1, x_2, \dots, x_n)$, его элементы – это значения углов в радианах. Сформировать для каждого значения x_i элементы целочисленных массивов g_i, m_i, s_i , которые определяют эти же углы в градусах, минутах и секундах.

Примечание. Секунды должны быть округлены до ближайшего целого значения. Интервалы значений: $0 \leq g_i < 360$; $0 \leq m_i < 60$; $0 \leq s_i < 60$.

Рекомендация.

Обозначим через $y = x_i$ исходное значения угла в радианах, через b – то же в градусах (x и b – вещественные значения), через g, m, s – целочисленные значения градусов, минут и секунд. Тогда требуемое преобразование можно выполнить в следующем порядке:

```

b:=180*y/pi
g:=trunc(b)
b:=180*(b-g)/pi
m:=trunc(b)
b:=180*(b-m)/pi

```

$s := \text{round}(b)$

9. Определить наибольший общий делитель всех чисел, которые содержатся в заданной последовательности $X = (x_1, x_2, \dots, x_n)$ целых положительных чисел.

Примечание.

Наибольший общий делитель $d(a, b)$ для массива $X = (x_1, x_2, \dots, x_n)$ определяется по алгоритму Евклида. Последовательность действий:

$$\begin{aligned} p &= d(x_1, x_2) \\ p &= d(p, x_3) \\ &\dots \dots \dots \\ p &= d(p, x_n) \end{aligned}$$

10. Элементы массива $A = (a_0, a_1, \dots, a_n)$ - это коэффициенты полинома $P(x)$ степени n . Определить коэффициенты полинома $P^2(x)$.

Указание. Коэффициенты a'_m , $m = 0..2n$ для полинома $P^2(x)$ вычисляются по следующим формулам:

$$a'_m = \begin{cases} \sum_{i=0}^m a_i a_{m-i}, & m = 0..n \\ \sum_{i=n+1}^{2n} a_{n-i} a_{m-n+i}, & m = (n+1)..2n \end{cases}$$

Задание рекомендуется выполнять в следующем порядке:

- 1) По коэффициентам a_i , $i = 0..n$ полинома $P(x)$, используя приведенные выше формулы, вычислить коэффициенты a'_j , $j = 0..2n$ полинома $P^2(x)$.
- 2) Ввести с клавиатуры конкретное значение аргумента x (например, $x = 2.0$) и по схеме Горнера вычислить значения полиномов $y_1 = P(x)$ и $y_2 = P^2(x)$.
- 3) Если y_1^2 и y_2 отличаются между собою не больше чем на малую величину ε , то программная реализация задачи выполнена верно.
Значение n для $P(x)$ брать в диапазоне 4 .. 10.

11. Элементы массива $X = (x_1, x_2, \dots, x_n)$ - это последовательность цифр целого числа, записанного в системе счисления с основанием q , $2 \leq q \leq 10$, $0 \leq x_i < q$. Переставить цифры числа в обратном порядке, удалить незначащие нули (если они имеются) и напечатать десятичное значение этого числа до и после перестановки. Значение q ввести с клавиатуры.

Указание. Десятичное значение числа вычислять по схеме Горнера.

Рекомендация.

Предположим, что после перестановки цифр числа и удаления незначащих нулей мы получили

$$X = (x_1, x_2, \dots, x_m),$$

где x_i - цифры числа в системе счисления с основанием q . Тогда это число можно представить в виде

$$P = x_1 q^{m-1} + x_2 q^{m-2} + x_3 q^{m-3} + \dots + x_{m-1} q + x_m$$

или

$$P = (\dots(x_1q + x_2)q + x_3)q + \dots + x_{m-1})q + x_m$$

Последнее выражение – это и есть схема Горнера для вычисления полинома.

12. Известно, что в целочисленном массиве $X = (x_1, x_2, \dots, x_n)$ три и только три числа равны между собой. Найти эти числа и переместить их в начало массива, сдвинув другие числа к концу этого массива.

13. За однократный просмотр целочисленного массива $X = (x_1, x_2, \dots, x_n)$ найти его максимальный положительный элемент x_{\max} и определить среднее арифметическое значение всех элементов массива, за исключением элементов, равных x_{\max} (в общем случае в массиве может быть несколько элементов, равных x_{\max}).

Указание. В программе должны быть учтены частные случаи, в том числе

- в массиве нет положительных элементов;
- все элементы массива положительные и равны друг другу.

14. В целочисленном массиве $X = (x_1, x_2, \dots, x_n)$ каждую пару x_i и x_j ($i, j = 1, 2, \dots, n; i \neq j$) нечетных элементов преобразовать в четные элементы по формулам: $x_i := x_i + 1$, $x_j := x_j - 1$. Пары элементов i, j выбирать в порядке их прохождения в массиве X . Определить, как при этом изменились максимальный и минимальный элементы массива X (в абсолютном отношении и в процентах).

Рекомендация.

Разработать процедуру `Transform(Var k, k1, k2: integer)`, где k – начало поиска нечетных элементов, $k1$ и $k2$ – индексы ближайших нечетных элементов. В начальной части процедуры установить $k1 = 0$ и $k2 = 0$. При первом обращении к процедуре `Transform` задать $k = 1$, при последующих обращениях $k = k2 + 1$. Если после окончания работы процедуры `Transform` будет получено $k1 = 0$ или $k2 = 0$, то дальнейший анализ массива X прекратить.

15. В заданном массиве целых чисел $X = (x_1, x_2, \dots, x_n)$ найти самый длинный подмассив, который является арифметической прогрессией.

Примечание. Если подмассив является арифметической прогрессией, то для его элементов должно выполняться соотношение

$$2x_k = x_{k-1} + x_{k+1}, \quad k = 2 \dots n-1$$

В общем случае начало серии – элемент с индексом $k-1$, конец серии – элемент с индексом k .

Пример.

Пусть мы имеем $X = (8, 7, 5, 7, 9, 11, 13, 4, 6)$.

Здесь начало серии выполняется при $k = 4$: $2x_4 = x_3 + x_5 \Rightarrow 14 = 5 + 9$.

Следовательно, начало серии – это элемент с индексом $k-1 = 3 \Rightarrow x_3 = 5$. Конец серии обнаруживается при $k = 7$: $2x_7 \neq x_6 + x_8 \Rightarrow 26 \neq 11 + 4$. Следовательно, конец серии – это элемент с индексом $k = 7 \Rightarrow x_7 = 13$.

Особый случай. Если последний элемент массива входит в состав последней серии, то его принадлежность к этой серии нужно проверять отдельно: если $2x_{n-1} = x_{n-2} + x_n$, то концом серии является элемент x_n .

16. По заданному целочисленному массиву $X = (x_1, x_2, \dots, x_n)$ сформировать массив $Y = (y_1, y_2, \dots, y_n)$ такой, что y_i - это количество элементов массива X , которые не превышают значение x_i (т.е. меньше или равно значения x_i) на конечном отрезке этого массива от элемента с индексом $i+1$ до элемента с индексом n . Вполне очевидно, что для y_n должно быть записано нулевое значение.

Пример. Для массива

$$X = (5, 8, 3, 7, 5, 4, 10, 8, 12, 15, 3, 7)$$

будем иметь

$$Y = (4, 7, 1, 4, 2, 1, 3, 2, 2, 2, 0, 0).$$

17. Заданы два целочисленных массива $X = (x_1, x_2, \dots, x_n)$ и $Y = (y_1, y_2, \dots, y_m)$. Пусть в массиве X содержится k_1 четных элементов, а в массиве Y - k_2 нечетных элементов. Обменять местами $k = \min(k_1, k_2)$ четных элементов массива X с нечетными элементами массива Y (в порядке их прохождения в массивах X и Y). Учсть, что в частном случае может быть $k = 0$.

18. Преобразовать массив $X = (x_1, x_2, \dots, x_n)$, расположив сначала его отрицательные, а потом неотрицательные элементы, сохранив при этом в группе отрицательных элементов их первоначальный относительный порядок, а в группе неотрицательных элементов изменив его на обратный. Определить, как при этом изменилось положение минимального по модулю элемента массива X . Буферный массив не использовать.

Пример.

Для массива

$$X = (5, 8, -3, 0, -4, 7, -10, -6, 0, -1)$$

получим

$$X = (-3, -4, -10, -6, -1, 5, 8, 0, 7, 0).$$

Примечание.

Эту задачу можно решить как с использованием буферного массива, так и без его использования. В последнем случае алгоритм решения будет несколько сложнее. Тем не менее, учитывая замечания в методических указаниях по поводу использования буферного массива, второй вариант решения задачи является более предпочтительным.

Кстати, примеры решения аналогичных задач имеются в приложениях к учебному пособию «Основы программирования на языке Турбо Паскаль».

19. Заданный вещественный массив $X = (x_1, x_2, \dots, x_n)$ осреднить следующим образом: максимальный и минимальный элементы заменить их средним арифметическим значением, то же сделать относительно максимального и минимального элементов преобразованного массива X и т.д. $n/2$ раз. Если в очередном цикле обработки массива X окажется, что его максимальный и минимальный элементы отличаются между собою не больше чем на значение ε (ε - достаточно малая величина), то дальнейшее преобразование массива X не производить. Определить, как изменились среднее арифметическое значение S и среднее квадратичное отклонение G элементов массива X после его преобразования. Изменения параметров S и G выразить в процентах, с учетом знака отклонения.

$$S = \frac{1}{n} \sum_{i=1}^n x_i; \quad G = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - S)^2}$$

20. Элементы массива $Y = (y_1, y_2, \dots, y_m)$ - это ординаты точек ломаной линии в равноотстоящих узлах по оси X . Среди "зубцов" ломаной, для которых выполняется отношение $y_{i-1} < y_i > y_{i+1}$ найти "вершину зубца" с максимальным превышением над соседними точками $\Delta y = y_i - \frac{y_{i-1} + y_{i+1}}{2}$ и удалить ее из состава ломаной. Определить, как при этом изменилась общая длина ломаной линии (считать, что расстояние между узлами по оси X равно 1). Изменение длины ломаной выразить в процентах.

Примечание.

Длина отрезка ломаной при шаге $h = 1$ по оси абсцисс

$$l_i = \sqrt{(y_{i+1} - y_i)^2 + 1}$$

21. Определить, содержатся ли в заданной последовательности целых положительных чисел $X = (x_1, x_2, \dots, x_n)$ числа Фибоначчи и, если это так, определить значение и положение максимального и минимального из таких чисел, после чего обменять их местами.

Указание. В программе использовать процедуру генерации чисел Фибоначчи, не превышающих значения анализируемого элемента x_i .

Числа Фибоначчи получают по формуле

$$f_{i+2} = f_{i+1} + f_i; \quad f_1 = 1; \quad f_2 = 1$$

22. Удалить из вещественного массива $X = (x_1, x_2, \dots, x_n)$ все элементы, которые превышают его среднее арифметическое значение S , кроме первого такого элемента, и определить, как при этом изменилось значение S (в процентах). Буферный массив не использовать.

23. При однократном просмотре целочисленного массива $X = (x_1, x_2, \dots, x_n)$ найти два максимальных по модулю элемента, кратных одновременно числам 2 и 3, и, если такие элементы существуют, обменять их местами в массиве.

Примечание. Если максимальный элемент кратен одновременно числам 2 и 3, необходимо произвести поиск другого элемента, кратного числам 2 и 3. Обмен не производить, если

- в массиве менее двух элементов, кратных числам 2 и 3;
- два максимальных элемента, которые кратны одновременно числам 2 и 3, в то же время равны друг другу..

Указание.

Для проверки делимости числа на 2 использовать функцию `odd`, для проверки делимости на 3 – операцию `mod`.

Вместо отдельной проверки кратности числам 2 и 3 можно проверять на кратность числу 6.

24. В массиве $X = (x_1, x_2, \dots, x_n)$ определить положение и размер первой и последней серии положительных четных элементов, в состав которых входит не менее четырех элементов, и если эти серии существуют, обменять их местами. Учесть, что длины серий, которые обмениваются, могут быть разными.

Примечание.

О методике поиска серии см.п.2.

25. Все положительные числа в вещественном массиве $X = (x_1, x_2, \dots, x_n)$ переставить в обратном порядке, не изменяя положения других чисел. Буферный массив не использовать.

Примечание. Если в анализируемой паре x_i и x_j имеет место $x_i = x_j$, то очевидно, что обмен таких элементов не имеет смысла.

Рекомендация.

Здесь целесообразно разработать две функции: `FromLeft(k1:integer):integer` и `FromRight(k2:integer):integer`. Первая из них производит поиск ближайшего справа положительного числа, начиная с позиции $k1$; вторая – ближайшего слева, начиная с позиции $k2$. Поиск прекращается, когда $k1 \geq k2$.

26. Вещественный массив $X = (x_1, x_2, \dots, x_n)$ содержит несколько отрицательных элементов, которые разделяют его на отдельные подмассивы. Сгруппировать элементы каждого подмассива в порядке возрастания. Учесть частные случаи (в массиве нет отрицательных элементов; подмассив пустой или содержит только один элемент).

Рекомендация.

Здесь целесообразно для группировки подмассива отдельно разработать процедуру `Group(k1, k2:integer)`, где $k1$ и $k2$ – начало и конец группируемого подмассива. Кроме того, должна быть функция `Negativ(k:integer):integer`, определяющая положение ближайшего отрицательного числа, начиная с позиции k .

27. Задан массив $X = (x_1, x_2, \dots, x_n)$, элементами которого являются натуральные числа. В составе массива X могут быть повторяющиеся элементы. Сформировать массив Y , включив в него все неповторяющиеся элементы из массива X .

Рекомендация. Целесообразно в программе предусмотреть функцию `Repeat(k:integer):boolean`, где k – индекс проверяемого элемента массива. Если выходное значение этой функции равно `false`, то такой элемент нужно включить в состав формируемого массива Y .

28. В вещественном массиве $X = (x_1, x_2, \dots, x_n)$ каждую пару x_i и x_j ($i, j = 1, 2, \dots, n; i \neq j$) отрицательных элементов преобразовать в положительные по формуле $x_i, x_j = \sqrt{x_i x_j}$. Пары элементов (i, j) выбирать в порядке их прохождения в массиве X . Определить, как при этом изменилось среднее арифметическое значение элементов массива X (в процентах).

29. Значение целого неотрицательного числа задано в виде массива его двоичных цифр $A = (a_n, a_{n-1}, \dots, a_1, a_0)$. Прибавить к двоичному представлению числа единицу и напечатать полученное десятичное значение.

Рекомендация. См. приложение «Числа и системы счисления» к учебному пособию «Основы программирования на языке Турбо Паскаль».

30. Задано два массива $X = (x_1, x_2, \dots, x_n)$ и $Y = (y_1, y_2, \dots, y_m)$, в состав которых входят натуральные числа, причем в каждом из этих массивов нет повторяющихся элементов. Сформировать массив Z , объединив массивы X и Y , при этом в массиве Z также не должно быть повторяющихся элементов.

Рекомендация.

1) Переписать в массив Z все элементы массива X .

2) Просматривая элементы $y_i, i = 1..m$, определить, имеется ли в массиве X элемент, равный значению y_i . Если это не обнаружено, то добавить элемент y_i в состав массива Z .

31. Задан вещественный массив $X = (x_1, x_2, \dots, x_n)$, его элементы - это значения расстояний в милях. Нужно сформировать для каждого значения x_i элементы целочисленных массивов k_i, m_i, s_i , которые определяют расстояние в километрах, метрах и сантиметрах.

Примечание. 1 миля = 1,609344 км.

Сантиметры должны быть округлены до ближайшего целого значения.

См. рекомендации к п.8.

32. Элементы массива $X = (x_1, x_2, \dots, x_n)$ - это последовательность цифр целого числа, записанного в системе счисления с основанием $q, 2 \leq q \leq 10, 0 \leq x_i < q$. Сгруппировать цифры числа в порядке уменьшения и напечатать десятичное значение этого числа до и после группировки. Значение q ввести с клавиатуры.

Указание. Десятичное значение числа вычислять по схеме Горнера.

См. рекомендации к п.11.

33. В целочисленном массиве $X = (x_1, x_2, \dots, x_n)$ найти наиболее длинную серию одинаковых элементов и удалить ее из состава массива. При этом искомая серия должна содержать не менее пяти элементов.

Примечание. См. п.2.

34. В целочисленном массиве $X = (x_1, x_2, \dots, x_n)$, не содержащем одинаковых элементов, удалить m наименьших элементов, а оставшиеся при этом элементы сгруппировать по возрастанию. Значение m ввести с клавиатуры, $m \ll n$.

Указание. Рекомендуется сначала сгруппировать исходный массив по возрастанию, а затем удалить из него первые m элементов.

35. Найти максимальную по длине строго монотонную (то есть убывающую или возрастающую) подпоследовательность заданной последовательности целых чисел $X = (x_1, x_2, \dots, x_n)$, после чего удалить ее из массива.

Примечание. Для элементов строго монотонной последовательности выполняются такие соотношения:

$$x_1 < x_2 < x_3 < \dots < x_m \quad \text{или} \quad x_1 > x_2 > x_3 > \dots > x_m.$$

О методике поиска серии см. п.2.

36. Из массива целых положительных чисел $X = (x_1, x_2, \dots, x_n)$ удалить все нечетные элементы, кроме первого такого элемента, после чего числа, которые остались, расположить в порядке уменьшения. Буферный массив не использовать.

37. Числовая последовательность формируется по правилу:

$$u_0 = \cos(x); u_1 = \cos(x+h); u_2 = \cos(x+2h); \dots; u_n = \cos(x+nh)$$

(значения x, n, h заданы). Среди тех элементов последовательности $U = (u_0, u_1, \dots, u_n)$, которые превышают по модулю заданное значение b , найти максимальный и минимальный элементы, после чего обменять их местами. Учесть, что в частном случае

таких элементов может быть меньше двух. Значения x, n, h, b должны быть введены с клавиатуры.

38. Элементы массива $A = (a_0, a_1, \dots, a_n)$ - это коэффициенты полинома $P(x)$ степени n . Вычислить коэффициенты полинома $P'(x)$ как производной от полинома $P(x)$.

Примечание. Коэффициенты нового полинома вычислять по формуле

$$a'_m = (m+1)a_{m+1}, \quad m = 0..n-1$$

Последовательность выполнения задания – см. п.10.

39. Если в вещественном массиве $X = (x_1, x_2, \dots, x_n)$ имеются серии, которые состоят не менее чем из трех отрицательных элементов, то переставить элементы этих серий в обратном порядке.

Примечание. О методике поиска серии см. п.2.

40. В массиве целых чисел $X = (x_1, x_2, \dots, x_n)$ найти наиболее длинную серию подряд расположенных попарно разных элементов и удалить ее из состава массива.

Примечание. Условие задачи означает, что в искомой серии нет одинаковых смежных элементов, хотя сам массив и может иметь одинаковые элементы. Например, в массиве

$$X = (7, 3, 3, 4, 8, 9, 11, 11, 12, 16, 23, 11, 23, 12, 15, 15, 14, 14, 5)$$

такой серией является (11, 12, 16, 23, 11, 23, 12, 15).

О методике поиска серии см. п.2.

41. Определить наименьшее общее кратное всех чисел, которые содержатся в заданной последовательности $X = (x_1, x_2, \dots, x_n)$ целых положительных чисел.

Примечание. Наименьшее общее кратное двух чисел m и n

$$k(m, n) = \frac{m \cdot n}{d(m, n)},$$

где $d(m, n)$ - наибольший общий делитель, вычисляемый по алгоритму Евклида.

Для переменных m, n, k рекомендуется использовать тип `longint`.

О методике поиска серии см.п.2.

42. Заданы два целочисленных массива $X = (x_1, x_2, \dots, x_n)$ и $Y = (y_1, y_2, \dots, y_m)$. В состав массива X дополнительно включить те элементы из массива Y , которые отсутствуют в массиве X . Определить, как при этом изменилось среднее арифметическое значение элементов массива X (в процентах).

43. В начальной части заданного массива $X = (x_1, x_2, \dots, x_n)$ расположить нулевые элементы, которые входят в его состав, а потом в порядке возрастания положительные элементы и в порядке убывания отрицательные элементы. Определить, как при этом изменилось положение максимального и минимального элементов массива X . Буферный массив не использовать.

Пример.

Для массива

$$X = (3, 8, -5, 0, 4, -1, 0, 7, -5, 0, 8, 10, -3)$$

получим

$$X = (0, 0, 0, 3, 4, 7, 8, 8, 10, -1), -3, -5, -5).$$

44. Значение целого неотрицательного числа задано в виде массива его троичных цифр $A = (a_n, a_{n-1}, \dots, a_1, a_0)$, $0 \leq a_i < 3$. Переставить троичные цифры числа в обратном порядке, удалить незначащие нули (если они имеются) и напечатать полученное десятичное значение.

Указание. Десятичное значение числа вычислять по схеме Горнера.

См. рекомендации к п.11.

45. Заданы вещественные массивы $X = (x_1, x_2, \dots, x_n)$ и $Y = (y_1, y_2, \dots, y_m)$. Пусть массив X имеет k_1 положительных элементов, а массив Y - k_2 отрицательных элементов. Обменять местами $k = \min(k_1, k_2)$ положительных элементов массива X с отрицательными элементами массива Y (в порядке их прохождения в массивах X и Y). Учесть, что в частном случае может быть $k = 0$.

46. Элементы массива $Y = (y_1, y_2, \dots, y_n)$ - это ординаты точек ломаной линии в равноотстоящих узлах по оси X . Выполнить сглаживание ломаной линии по формуле $y'_i = \frac{y_{i-1} + 2y_i + y_{i+1}}{4}$, ($i = 2, 3, \dots, n-1$); $y'_1 = \frac{y_1 + y_2}{2}$; $y'_n = \frac{y_{n-1} + y_n}{2}$, где y'_i - ординаты преобразованной линии. Определить, насколько изменилось среднее отклонение зубцов ломаной относительно соседних точек.

Примечание. Вершины зубцов ломаной - это точки, для которых выполняется отношение $y_{i-1} < y_i > y_{i+1}$. Отклонение зуба относительно соседних точек

$$h = y_i - \frac{y_{i-1} + y_{i+1}}{2}.$$

47. В массиве $X = (x_1, x_2, \dots, x_n)$ определить положение и размер последней серии положительных нечетных элементов, в состав которой входит от двух до пяти элементов, после чего удалить эту серию из состава массива.

Указание. См. п.2.

48. Удалить из целочисленного массива $X = (x_1, x_2, \dots, x_n)$ нулевые элементы и переставить в обратном порядке элементы, которые остались. Определить, как изменились при этом значение и положение минимального x_{\min} и максимального x_{\max} элементов данного массива, а также его среднее арифметическое значение S . Изменение значения параметра S выразить в процентах. Буферный массив не использовать.

49. Задан целочисленный массив $X = (x_1, x_2, \dots, x_n)$, в котором могут быть одинаковые числа. Найти ближайшую от начала массива пару одинаковых чисел, разность индексов которых минимальная, после чего удалить эти числа из массива.

Примечание. Вполне очевидно, что при обнаружении двух смежных одинаковых чисел минимальная разность индексов становится равной единице. Поэтому при обнаружении указанной ситуации дальнейший поиск заданных чисел нужно прекратить.

50. Заданы два массива $X = (x_1, x_2, \dots, x_n)$ и $Y = (y_1, y_2, \dots, y_m)$, в состав которых входят натуральные числа, причем в каждом из этих массивов нет повторяющихся элементов. Сформировать массив Z , включив в него те элементы из массива X , которые отсутствуют в массиве Y . Числа, перенесенные в массив Z , удалить из массива X .

Пример. Пусть мы имеем

$$X = (4, 8, 5, 7, 1^*0, 18, 11, 14), \quad n = 8$$

и $Y = (6, 3, 7, 4, 9, 12, 16, 14, 11)$, $m = 9$.

Тогда получим

$Z = (8, 5, 10, 18, 15)$ и $X = (4, 7, 11)$.

51. При однократном просмотре целочисленного массива $X = (x_1, x_2, \dots, x_n)$ найти два максимальных числа x_{\max_1} и x_{\max_2} соответственно среди четных и нечетных по значению элементов массива, после чего переставить в обратном порядке элементы подмассива, позиционно расположенные между этими числами, включая элементы x_{\max_1} и x_{\max_2} . Учесть частные случаи:

- в массиве нет четных или нечетных элементов;
 - x_{\max_1} позиционно расположен перед x_{\max_2} или наоборот.
-

52. Все четные числа в целочисленном массиве $X = (x_1, x_2, \dots, x_n)$ переставить в обратном порядке, не изменяя положения других чисел.

Примечание. Если в анализируемой паре x_i и x_j имеет место $x_i = x_j$, то очевидно, что обмен таких элементов не имеет смысла.

Рекомендация.

1) Вариант с буферным массивом.

Создать дополнительный массив Y , в который записать индексы четных чисел в массиве X . Пусть, например, задано

$X = (5, 8, 3, 7, 4, 6, 2, 1, 10, 15, 5, 8, 6, 9)$, $n = 14$

В этом случае

$Y = (2, 5, 6, 7, 9, 12, 13)$, $m = 7$.

Тогда обмену подвергаются элементы $x[y[1]]$ и $x[y[m]]$, $x[y[2]]$ и $x[y[m-1]]$ и т.д.

2) Вариант без буферного массива.

Разработать две функции: `FromLeft(k1:integer):integer` и `FromRight(k2:integer):integer`. Первая из них производит поиск ближайшего справа четного числа, начиная с позиции $k1$; вторая – ближайшего слева, начиная с позиции $k2$. Поиск прекращается, когда $k1 \geq k2$.

53. Элементы вещественного массива $X = (x_1, x_2, \dots, x_n)$ строго упорядочены по возрастанию, то есть $x_1 < x_2 < x_3 < \dots < x_n$, при этом $x_{i+1} - x_i > \varepsilon$ ($i = 1, \dots, n-1$; ε - малое число, например, 0.001).

Элементы вещественного массива $Y = (y_1, y_2, \dots, y_m)$, $m \leq n$ расположены в произвольном порядке. Включить в состав массива X те элементы y_j , $j = 1..m$, которые отличаются от элементов x_i не менее чем на ε , сохранив при этом упорядоченность массива X . Буферный массив не использовать.

Примечание.

Вполне возможен частный случай, когда все элементы массива Y будут добавлены в массив X . Поэтому в рассматриваемой задаче целесообразно записать объявления типов и переменных следующим образом:

```
Const Nmax = 200;  
Type Xar = array[1..2*Nmax] of real;  
      Yar = array[1..Nmax] of real;  
Var X : Xar; Y : Yar;  
     n, m : integer;
```

Рекомендация.

1) Для очередного элемента $y_i, i=1..m$ определить индекс k ближайшего большего или равного ему значения в массиве X .

2) При $1 < k < m$ сравнить элемент y_i с элементами x_{k-1} и x_k .

Если $|y_i - x_{k-1}| > \varepsilon$ и $|y_i - x_k| > \varepsilon$, то сдвинуть подмассив x_k, x_{k+1}, \dots, x_n на один элемент вправо, увеличив при этом значение n . Установить $x_k := y_i$.

3) Если $k = 1$, то проверяется лишь отношение $|y_i - x_1| > \varepsilon$.

4) Если $k = 0$, то проверяется лишь отношение $|y_i - x_n| > \varepsilon$ и, если оно выполняется, то устанавливается $x_{n+1} := y_i; n := n + 1$.

54. Целочисленный массив $X = (x_1, x_2, \dots, x_n)$ состоит из неотрицательных элементов. Найти индекс элемента, для которого сумма элементов, которые стоят до него, меньше всего отличается от суммы элементов, которые стоят после него, после чего удалить этот элемент из массива.

Рекомендация.

Для элемента $x_i, i=1..n$ определяются $S1 = \sum_{j=1}^{i-1} x_j$ и $S2 = \sum_{j=i+1}^n x_j$. Если $i = 1$, то

естественным образом получим $S1 = 0$; при $i = n$ будем иметь $S2 = 0$. В программе должен производиться поиск такого элемента x_i , для которого параметр $dS = |S1 - S2|$ имеет минимальное значение.

55. При однократном просмотре массива $X = (x_1, x_2, \dots, x_n)$ определить значения и положения (индексы) двух наибольших элементов, а затем удалить их из массива.

Указание. См. п.50 в методических указаниях к лабораторной работе №4.

56. Задан массив целых положительных чисел $X = (x_1, x_2, \dots, x_n)$. Найти в этом массиве значение и положение максимального x_{\max} и минимального x_{\min} элементов, являющихся степенью числа 2, после чего удалить из массива элементы, которые позиционно расположены между x_{\min} и x_{\max} . Учесть возможные частные случаи:

- в массиве нет элементов, являющихся степенью числа 2;
- имеется лишь один такой элемент;
- $x_{\max} = x_{\min}$.

Примечание.

Чтобы определить, является ли значение целочисленной переменной y степенью числа 2, можно использовать следующую программу:

```
While y mod 2 do  
  y := y div 2;
```

Если значение переменной y является степенью числа 2, то после окончания работы цикла получим $y = 1$, в противном случае $y > 1$.

57. В заданном массиве целых положительных чисел $X = (x_1, x_2, \dots, x_n)$ найти самый длинный подмассив, который является геометрической прогрессией, после чего удалить его из массива.

Примечание. Если подмассив является геометрической прогрессией, то для его элементов должно выполняться соотношение

$$x_i^2 = x_{i-1} \cdot x_{i+1}.$$

О методике поиска серии см.п.2 и п.15.

58. Задан массив $X = (x_1, x_2, \dots, x_n)$, элементами которого являются натуральные числа. В составе массива X могут быть повторяющиеся элементы. Сформировать массив Y , включив в его состав по одному экземпляру повторяющихся элементов из массива X .

Рекомендация.

1) Разработать функцию $\text{Repeat}(z:\text{integer}):\text{boolean}$, которая определяет, является ли значение z повторяющимся в массиве X .

2) Если $\text{Repeat}(x[i]) = \text{false}$, то проверить, не было ли ранее включено значение x_i в массив Y ; если это предположение не подтверждается, то добавить значение x_i в состав массива Y

59. В массиве $X = (x_1, x_2, \dots, x_n)$ наименьший элемент разместить на первом месте, наименьший из тех, что остались, - на последнем месте; очередной по убыванию - на втором месте, очередной после него - на предпоследнем месте и так далее до середины массива.

Пример. Для массива

$$X = (20, 10, 15, 5, 10, 8, 7, 1, 7, 14, 15)$$

получим

$$X = (1, 7, 8, 10, 15, 20, 15, 14, 10, 7, 5).$$

Рекомендация.

При использовании буферного массива решение рассматриваемой задачи может быть выполнено следующим образом.

- 1) Сгруппировать массив X по возрастанию.
- 2) Записать

$$y_1 := x_1; \quad y_n := x_2$$

$$y_2 := x_3; \quad y_{n-1} := x_4$$

$$y_3 := x_5; \quad y_{n-2} := x_6$$

.....

В общем случае в массив Y записываются элементы y_i и y_j , начальные значения их индексов $i = 1, j = n$. Цикл заполнения массива Y продолжается до тех пор, пока $i < j$.

- 3) Если количество элементов n нечетное, то дополнительно записывается

$$y_{m+1} := x_n, \text{ где } m = n \text{ div } 2.$$

- 4) Массив Y переписывается в массив X .

60. Выполнить циклический сдвиг массива $X = (x_1, x_2, \dots, x_n)$ на k элементов ($0 \leq k \leq n$) в направлении, которое определяется значением переменной s ($s = 0$ - влево, $s = 1$ - вправо). Например, при циклическом сдвиге массива $X = (4, -8, 6, 12, 1, 0, 7, 9)$ влево на 3 элемента получим $X = (12, 1, 0, 7, 9, 4, -8, 6)$. Значение переменных k и s ввести с клавиатуры. При $k = 0$ и $k = n$ сдвиг массива не производить.

Рекомендация.

Пусть $s = 1, 1 \leq k < n$.

- 1) Временно последние k элементов из массива X перенести в буферный массив Y :

$$y_1 := x_{n-k+1}$$

$$y_2 := x_{n-k+2}$$

.....

$$y_k := x_n$$

2) Сдвинуть массив X на k элементов вправо:

$$x_n := x_{n-k}$$

$$x_{n-1} := x_{n-k-1}$$

$$x_{n-2} := x_{n-k-2}$$

.....

$$x_{k+1} := x_1$$

3) В начальную часть массива X перенести элементы из массива Y :

$$x_1 := y_1$$

$$x_2 := y_2$$

.....

$$x_k := y_k$$

При $s = 0$ обработка массива X выполняется в основном аналогично описанному выше.

61. В каждой серии отрицательных элементов, содержащейся в вещественном массиве $X = (x_1, x_2, \dots, x_n)$ и имеющей в своем составе не менее трех чисел, удалить первый и последний элементы серии.

Примечание. О методике поиска серии см.п.2.

62. Массив целых чисел $X = (x_1, x_2, \dots, x_n)$ содержит некоторое количество совпадающих элементов. Определить количество пар совпадающих элементов и удалить в каждой из них второй такой элемент. Определить, как изменилось среднее арифметическое значение элементов массива X после его преобразования (в процентах).

Указание. Если $x_i = x_j$, $j = i + 1 \dots n$, то после удаления элемента x_j элемент x_i уже не рассматривается как принадлежащий следующей паре одинаковых элементов.

Пример. В массиве

$$X = (8, 3, 5, 8, 4, 8, 7, 5, 3, 8, 5, 8, 6, 5, 8, 9, 8)$$

три пары чисел 8, две пары чисел 5 и одна пара чисел 3. Преобразованный массив будет иметь следующий вид:

$$X = (8, 3, 5, 4, 8, 7, 5, 8, 6, 9, 8)$$

63. Задан целочисленный массив $X = (x_1, x_2, \dots, x_n)$. Найти в этом массиве такой отрезок максимальной длины, в котором первое число равно последнему, второе - предпоследнему и т.д. Определить длину этого отрезка, индекс его первого элемента и среднее арифметическое значение элементов, входящих в состав данного отрезка.

Примечание. О методике поиска серии см.п.2.

64. Элементы массива $A = (a_0, a_1, \dots, a_n)$ - это коэффициенты полинома $P(x)$ степени n . Определить коэффициенты полинома $P(x) + P(nx)$.

Примечание. Коэффициенты нового полинома вычислять по формуле

$$a'_m = a_m(n^m + 1), \quad m = 0..n$$

Для вычисления n^m использовать рекуррентное отношение

$$n^m = n \cdot n^{m-1}$$

Последовательность выполнения задания – см. п.10.

65. Задан целочисленный массив $X = (x_1, x_2, \dots, x_n)$, в состав которого входят только положительные элементы. Если в этом массиве имеются пары элементов $x_i = b$ и $x_j = b + 1$, где b - четное значение и $j > i$, то в случае, когда в массиве больше нет равных им элементов, удалить их из состава массива. В любом случае после удаления в массиве X должен оставаться хотя бы один элемент.

Пример. В соответствии с условием задачи в массиве

$$X = (1, 8, 14, 12, 8, 10, 6, 8, 15, 7, 13, 4, 31, 13)$$

должны быть удалены элементы 14 ($i = 3$) и 15 ($j = 9$), 6 ($i = 7$) и 7 ($j = 10$). При этом получим

$$X = (1, 8, 12, 8, 10, 8, 13, 4, 31, 13).$$

Пусть мы имеем массив

$$X = (6, 7, 8, 9, 10, 11, 0, 1, 4, 5)$$

Здесь в принципе подлежат удалению каждая пара x_1 и x_2 , x_3 и x_4 , x_5 и x_6 , x_7 и x_8 , x_9 и x_{10} . Однако по условию задачи после удаления элементов массив не должен быть пустым. Это приведет к тому, что в данном случае конечное содержимое данного массива будет иметь вид $X = (4, 5)$.

66. Элементы массива $X = (x_1, x_2, \dots, x_n)$ - это последовательность цифр дробной части числа, записанного в системе счисления с основанием q , $2 \leq q \leq 10$, $0 \leq x_i < q$. Переставить элементы массива X в обратном порядке, удалить незначащие нули (если они имеются) и напечатать десятичное значение числа до и после перестановки его цифр. Значение q ввести с клавиатуры.

Указание. Десятичное значение числа вычислять по схеме Горнера.

Рекомендация.

Запишем дробную часть числа в виде

$$X = (b_1, b_2, b_3, \dots, b_m)$$

Это означает

$$z = b_1 q^{-1} + b_2 q^{-2} + \dots + b_m q^{-m}$$

По схеме Горнера

$$z = (\dots((b_1 / q + b_2) / q + b_3) / q + \dots + b_m) / q$$

Последовательность вычислений:

$$z := 0$$

$$z := (z + b_1) / q$$

$$z := (z + b_2) / q$$

$$z := (z + b_3) / q$$

.....

$$z := (z + b_m) / q$$

67. Заданы два целочисленных массива $X = (x_1, x_2, \dots, x_n)$ и $Y = (y_1, y_2, \dots, y_m)$, причем в каждом из них элементы могут повторяться. Удалить из массивов X и Y все элементы, которые одновременно содержатся в обоих массивах. Определить, как при этом изменились максимальные и минимальные значения элементов массива X и массива Y .

Пример. Массивы

$$X = (8, 5, 4, 5, 8, 6, 10, 8, 5, 10, 5) \text{ и } Y = (12, 15, 8, 5, 8, 5, 6, 10, 4, 11)$$

после обработки примут вид

$$X = (8, 5, 10, 5) \text{ и } Y = (12, 15, 11).$$

Примечание. Если в массиве X содержится k_1 одинаковых элементов, а в массиве Y k_2 таких же элементов, то удаляются $k = \min(k_1, k_2)$ элементов.

68. По заданному целочисленному массиву $X = (x_1, x_2, \dots, x_n)$ сформировать массив $Y = (y_1, y_2, \dots, y_m)$ такой, что y_i - это количество элементов в массиве X , которые превосходят значение x_i на начальном отрезке массива X длиной $i-1$.

Пример. Для массива

$$X = (7, 10, 5, 3, 12, 4, 6, 3, 1, 8, 10)$$

получим

$$Y = (0, 0, 2, 0, 0, 4, 3, 6, 8, 2, 1).$$

69. Два целых неотрицательных числа A и B заданы массивами двоичных цифр $A = (a_n, a_{n-1}, \dots, a_1, a_0)$ и $B = (b_m, b_{m-1}, \dots, b_1, b_0)$. Найти сумму $C = A+B$ в двоичном изображении и напечатать десятичное значение чисел A , B и C .

Указание. Десятичное значение числа вычислять по схеме Горнера.

Примечание. О сложении и вычитании двоичных чисел см. приложение «Числа и системы счисления» в учебном пособии «Основы программирования на языке Turbo Паскаль».

70. Задан целочисленный массив $X = (x_1, x_2, \dots, x_n)$, в составе которого могут быть повторяющиеся элементы. Сформировать массивы $V = (v_1, v_2, \dots, v_m)$ и $K = (k_1, k_2, \dots, k_m)$, $1 \leq m \leq n$, где v_j определяет значение элемента массива X , а k_j - количество его повторений.

Пример. Для массива

$$X = (7, 8, 3, 5, 8, 7, 4, 6, 12, 10, 11, 8, 3, 8, 7, 5, 6, 4, 10)$$

получим

$$V = (7, 8, 3, 5, 4, 6, 12, 10, 11); \quad K = (3, 4, 2, 2, 2, 2, 1, 2, 1).$$

71. Заданы вещественные массивы $X = (x_1, x_2, \dots, x_n)$ и $Y = (y_1, y_2, \dots, y_n)$. Считая элементы (x_i, y_i) координатами точек на плоскости, определить, есть ли в массивах X , Y три смежные точки $(1,2,3)$, $(2,3,4)$, $(3,4,5)$, ..., которые лежат на одной прямой. Напечатать номер первой и последней групп таких точек (если они имеются).

Примечание. Использовать следующее свойство: если три точки лежат на одной прямой, то площадь треугольника с вершинами в этих точках равна нулю. Удвоенная площадь треугольника

$$S = x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2).$$

Площадь треугольника считать нулевой, если $|S| < \varepsilon$, где ε - малое число (например, 0.001).

Примечание. В условии задачи предполагается, что на одной прямой может быть расположено три и более точек.

72. Значение целого неотрицательного числа задано в виде массива его двоичных цифр $A = (a_n, a_{n-1}, \dots, a_1, a_0)$. Вычесть из двоичного изображения числа единицу и напечатать десятичное значение исходного и преобразованного чисел.

Указание. Десятичное значение числа вычислять по схеме Горнера.

Примечание. О сложении и вычитании двоичных чисел см. приложение «Числа и системы счисления» в учебном пособии «Основы программирования на языке Turbo Паскаль».

73. Удалить в заданном целочисленном массиве $X = (x_1, x_2, \dots, x_n)$ “лишние” (кроме первого) элементы таким образом, чтобы те элементы, которые остались, образовали возрастающую последовательность

$$x_1 \leq x_2 \leq x_3 \leq \dots \leq x_n.$$

Преобразование массива выполнить за один его просмотр.

74. В целочисленном массиве $X = (x_1, x_2, \dots, x_n)$ удалить все нечетные элементы, кроме последнего такого элемента. Определить, как при этом изменились среднее арифметическое значение S и среднее квадратичное отклонение G элементов массива X (в процентах). Буферный массив не использовать.

Примечание. См. п.19.

75. Элементы каждой серии положительных чисел массива $X = (x_1, x_2, \dots, x_n)$ сгруппировать в порядке уменьшения. В состав серии должно входить не меньше трех элементов.

Примечание. О методике поиска серии см.п.2.

76. Значение целой части неотрицательного вещественного числа задано в системе счисления с основанием q ($2 \leq q \leq 10$) в виде массива цифр $A = (a_n, a_{n-1}, \dots, a_1, a_0)$, значение его дробной части - в виде массива цифр $B = (b_1, b_2, \dots, b_m)$. Выполнить округление числа, оставив в нем k дробных цифр ($0 \leq k < m$), и напечатать полученное десятичное значение. Параметр k ввести с клавиатуры.

Рекомендация.

Для округления числа до k разрядов требуется к его $(k+1)$ -му разряду добавить значение p :

$$p = \begin{cases} q \operatorname{div} 2, & \text{если } q - \text{четное} \\ q \operatorname{div} 2 + 1, & \text{если } q - \text{нечетное} \end{cases}$$

Если в $(k+1)$ -ом разряде образуется значение $x_{k+1} \geq q$, то

- 1) отбросить разряды $k+1, k+2, \dots, m$;
- 2) число $a_n a_{n-1} \dots a_0 . b_1 b_2 \dots b_k$ сложить с единицей в k -ом разряде.

Если $x_{k+1} < q$, то сложение не производить.

77. В заданном массиве целых положительных чисел $X = (x_1, x_2, \dots, x_n)$ найти наибольший подмассив, который является перестановкой некоторого отрезка ряда натуральных чисел. Например, таким отрезком является последовательность (18, 21, 16, 19, 14, 22, 15, 20, 17).

Примечание.

Подмассив является перестановкой некоторого отрезка натурального ряда чисел, если

- 1) количество элементов подмассива

$$k = x_{\max} - x_{\min} + 1,$$

где x_{\max} , x_{\min} — максимальный и минимальный элементы подмассива;

- 2) в подмассиве нет одинаковых чисел.

О методике поиска серий см.п.2.

78. Для заданного значения n сформировать массив $X = (x_1, x_2, \dots, x_n)$ как последовательность чисел Фибоначчи. Оценить, насколько отличается положение элемента x_i , наиболее близкого к среднему арифметическому значению S массива X , от положения элемента x_j , наиболее близкого к среднему геометрическому значению P этого же массива.

Примечание.

$$S = \frac{1}{n} \sum_{i=1}^n x_i; \quad P = \sqrt[n]{\prod_{i=1}^n x_i}$$

О степени отличия элемента x_i от значений S и P см. примечания к варианту 18 в заданиях к лабораторной работе № 4.

79. Элементы массива $A = (a_0, a_1, \dots, a_n)$ - это коэффициенты полинома $P(x)$ степени n . Определить значения коэффициентов полинома $P_1(x) = P(x) + 2P(x) + 3P(x) + \dots + nP(x)$.

Примечание. Пользуясь формулой суммы членов арифметической прогрессии, получим

$$P_1(x) = \frac{n(n+1)}{2} P(x)$$

Тогда коэффициенты нового полинома

$$a'_m = \frac{n(n+1)}{2} a_m, \quad m = 0..n$$

Последовательность выполнения задания – см. п.10.

80. Элементы вещественных массивов $X = (x_1, x_2, \dots, x_n)$ и $Y = (y_1, y_2, \dots, y_n)$ определяют координаты точек ломаной линии. Удалить из состава ломаной отрезок минимальной длины и отрезок максимальной длины. Определить, как при этом изменилась общая длина ломаной линии и средняя длина ее отрезков.

Указание. Для удаления первого или последнего отрезков достаточно удалить соответственно первую или последнюю точку ломаной линии. Для удаления других отрезков нужно выполнить параллельный перенос части точек.

Предположим, что из состава ломаной нужно удалить отрезок $(x_k, y_k) - (x_{k+1}, y_{k+1})$. В этом случае надо выполнить следующие действия:

1) вычислить $dx = x_{k+1} - x_k$; $dy = y_{k+1} - y_k$;

2) скорректировать координаты точек

$$x_{k+2} := x_{k+2} - dx; \quad x_{k+3} := x_{k+3} - dx; \quad \dots; \quad x_n := x_n - dx;$$

$$y_{k+2} := y_{k+2} - dy; \quad y_{k+3} := y_{k+3} - dy; \quad \dots; \quad y_n := y_n - dy;$$

3) удалить из массивов X и Y точку (x_{k+1}, y_{k+1}) ;

4) значение n уменьшить на 1.

81. Преобразовать массив $X = (x_1, x_2, \dots, x_n)$, расположив сначала его отрицательные, а потом неотрицательные элементы, сохранив при этом в группе неотрицательных элементов их первоначальный относительный порядок, а в группе отрицательных элементов изменив его на обратный. Определить, как при этом изменилось положение минимального элемента массива X . Буферный массив не использовать.

82. Удалить из массива $X = (x_1, x_2, \dots, x_n)$ его максимальный и минимальный элементы, после чего переставить элементы, которые остались, в обратном порядке. Определить, как изменилось среднее арифметическое значение элементов массива X после его преобразования (в процентах). Буферный массив не использовать.

83. В вещественном массиве $X = (x_1, x_2, \dots, x_n)$ могут быть положительные, отрицательные и нулевые элементы. Найти такой фрагмент массива, ограниченный индексами p и q , что $y = \sum_{i=p}^q x_i$ имеет максимальное значение.

Примечание. В задаче фактически идет речь о серии с заданным свойством. О методике поиска серий см.п.2.

84. При однократном просмотре массива $X = (x_1, x_2, \dots, x_n)$ определить среднее арифметическое его трех последних положительных элементов, после чего удалить их из массива. Учесть частный случай, когда массив содержит меньше трех таких элементов.

85. Целочисленный массив $X = (x_1, x_2, \dots, x_n)$ разделяется нулевыми элементами на несколько подмассивов. Первый и последний элементы массива ненулевые. Сгруппировать элементы каждого подмассива в порядке возрастания. Учесть частные случаи (в массиве нет нулевых элементов; подмассив пустой или содержит лишь один элемент).

86. В массиве $X = (x_1, x_2, \dots, x_n)$ записаны значения разрядов двоичного числа, причем первой цифрой является 1. Инвертировать двоичные цифры (заменить 0 на 1 и наоборот), удалить незначащие нули (если они имеются) и напечатать десятичное значение полученного числа.

Указание. Десятичное значение числа вычислять по схеме Корнера (см.п.11).

87. В вещественном массиве $X = (x_1, x_2, \dots, x_n)$ может быть несколько серий положительных элементов, содержащих не менее двух чисел каждая. Переставить местами первую и последнюю такие серии (если они имеются). При этом учесть, что длины этих серий могут быть разными.

Примечание. О методике поиска серии см.п.2.

88. В целочисленном массиве $X = (x_1, x_2, \dots, x_n)$ расположить в порядке возрастания положительные элементы, которые входят в его состав, а потом в порядке уменьшения - отрицательные элементы, не изменяя их относительного положения в массиве. После этого удалить нулевые элементы, если они имеются в массиве X . Определить, как при этом изменилось положение максимального и минимального элементов массива X . Буферный массив не использовать.

Пример. Для массива

$$X = (4, 6, 8, -3, 10, -14, 20, -8, 0, 6, 3, -10, 0, -8, 16, 10, -12, 0, 6, 3, 7, 8)$$

получим

$$X = (3, 3, 4, 6, 6, 6, 7, 8, 8, 10, 10, 16, 20, -3, -8, -8, -10, -12, -14).$$

89. В массиве $X = (x_1, x_2, \dots, x_n)$ переставить местами последний и предпоследний положительные элементы, третий и четвертый от конца массива положительные элементы и т.д. Если количество положительных элементов в массиве X меньше двух,

преобразование массива не производить. Определить, как изменилось положение минимального и максимального по модулю элементов массива X при его преобразовании.

90. Найти минимальное из чисел, которые встречаются в целочисленном массиве $X = (x_1, x_2, \dots, x_n)$ ровно два раза (если такое число имеется), после чего удалить эти два числа из массива.

91. В массиве $X = (x_1, x_2, \dots, x_n)$ обменять местами максимальный элемент x_{\max} с первым нулевым, а минимальный элемент x_{\min} - с последним нулевым элементом массива. Если в массиве только один нулевой элемент, то обмену подлежит тот из элементов x_{\max} и x_{\min} , который ближе расположен к нулевому элементу (если эти расстояния одинаковые, то преимущество отдать элементу x_{\max}). Учесть частные случаи (в массиве нет нулевых элементов, максимальный или минимальный элемент равен нулю).

92. Числовая последовательность $X = (x_1, x_2, \dots, x_n)$ пилообразная, если

$$x_1 < x_2 > x_3 < x_4 > \dots > x_n$$

или

$$x_1 > x_2 < x_3 > x_4 < \dots < x_n .$$

Найти в массиве X наиболее длинную пилообразную последовательность, после чего удалить ее из состава массива.

Примечание. О методике поиска серий см.п.2.

93. Нулевые элементы разделяют целочисленный массив $X = (x_1, x_2, \dots, x_n)$ на несколько подмассивов разной длины. Первый и последний элементы массива ненулевые. Обменять местами подмассивы максимальной и минимальной длины.. Учесть частные случаи (в массиве нет нулевых элементов, все элементы массива равны нулю, все подмассивы имеют одинаковую длину).

94. Для заданного вещественного массива $X = (x_1, x_2, \dots, x_n)$ вычислить среднее арифметическое значение S и среднее квадратичное отклонение G , после чего преобразовать массив X , записав в его начале элементы, расположенные в интервале $[S-G, S+G]$, потом элементы, которые расположены в интервале $[S-2G, S+2G]$ и не вошли в первую группу элементов, после этого - все другие элементы массива X . В преобразованном массиве в каждой из трех групп элементов должен быть сохранен их первоначальный относительный порядок. Определить, как в этом случае изменилось положение максимального и минимального элементов массива X .

Примечание. См. п.19.

95. Элементы массива $A = (a_0, a_1, \dots, a_n)$ - это коэффициенты полинома $P(x)$ степени n . Определить коэффициенты полинома $P(bx) - P(cx)$, где b и c - заданные вещественные числа, введенные с клавиатуры.

Примечание. Вычисление коэффициентов нового полинома выполнять по формуле

$$a'_m = a_m(b^m - c^m), \quad m = 0..n$$

При этом для определения b^m и c^m использовать рекуррентные отношения

$$b^m = b \cdot b^{m-1}; \quad c^m = c \cdot c^{m-1}$$

96. Заданы целочисленные массивы $X = (x_1, x_2, \dots, x_n)$ и $Y = (y_1, y_2, \dots, y_m)$. Найти наибольший элемент y_i , который не имеет себе равных в массиве X , и, если обнаружен такой элемент, то добавить его в массив X и одновременно удалить из массива Y . Учесть частный случай, когда все элементы y_i имеются в массиве X .

97. Значение целой части неотрицательного вещественного числа задано в виде массива двоичных цифр $A = (a_n, a_{n-1}, \dots, a_1, a_0)$, значение его дробной части - в виде массива двоичных цифр $B = (b_1, b_2, \dots, b_m)$. По отношению к данному числу выполнить следующее:

- переставить отдельно цифры целой и дробной части числа в обратном порядке;
- удалить незначащие нули в целой и дробной частях, если имеются такие нули.

Напечатать полученное десятичное значение, используя для его определения схему Горнера.

Примечание. О схеме Горнера см. п. 11 и 66.

98. В целочисленном массиве $X = (x_1, x_2, \dots, x_n)$ между каждыми двумя смежными отрицательными элементами вставить нулевой элемент. Определить, как при этом изменилось среднее арифметическое значение элементов массива (в процентах). Буферный массив не использовать.

Примечание. В частном случае все элементы исходного массива могут быть отрицательными. Тогда размер массива увеличится на $(n-1)$ элемент, что должно быть учтено при объявлении типа массива:

```
Const Nmax = 200;  
Type Xar = array[1..2*Nmax-1] of real;  
Var X : Xar;
```

99. Задан массив $X = (x_1, x_2, \dots, x_n)$. Найти длину k самой длинной "пилообразной (зубьями вверх)" последовательности чисел, которые идут подряд: $x_{p+1} < x_{p+2} > x_{p+3} < \dots > x_{p+k}$, после чего удалить ее из состава массива. Определить, как при том изменилось значение и положение максимального элемента массива.

Примечание. О методике поиска серии см. п. 2.

100. Все отрицательные числа в целочисленном массиве $X = (x_1, x_2, \dots, x_n)$ переставить в обратном порядке, не изменяя положения других чисел. После этого удалить нулевые числа, если они имеются в составе массива. Определить, как при этом изменилось среднее арифметическое значение элементов массива (в процентах).

Рекомендация.

Разработать две функции: `Neg1(k:integer):integer` и `Neg2(k:integer):integer`. Функция `Neg1` производит поиск слева направо ближайшего отрицательного элемента, функция `Neg2` выполняет то же, но справа налево. Тогда начальная часть программы обработки массива X может иметь следующий вид:

```
Cond:=true; k1:=0; k2:=n+1;  
While Cond do  
  Begin  
    k1:=Neg1(k1+1);
```

```
If k1=0 then
  Cond:=false
Else
  Begin
    k2:=Neg2(k2-1);
    If (k2=0) or (k1>=k2) then
      Cond:=false
    Else
      Begin
        Обмен элементов x[k1] и x[k2]
      End
    End
  End
.....
```
