

Задания для семестровой работы (3 задачи)

1. Обработка компонентных файлов

Ниже приведен пример оформления лабораторной работы.

Л. Р. № __	Студент	Иванов И. И.
«Обработка компонентных файлов»	Группа	ХХ-999
	Дата	дд.мм.гг
	Допуск	
	Выполнение	
Вариант 88	Отчет	

Условие задачи

Ввести файл F действительных чисел. В файле элементы стоящие после второго нуля переставить в обратном порядке. Вывести файл до и после преобразования.

Тестовые примеры

1) Входные данные:

файл f до изменения:

5 74 2 8 0 2 5 0 6 5 4 1 56

Выходные данные:

файл f после изменения:

5 74 2 8 0 2 5 0 56 1 4 5 6

2) Входные данные:

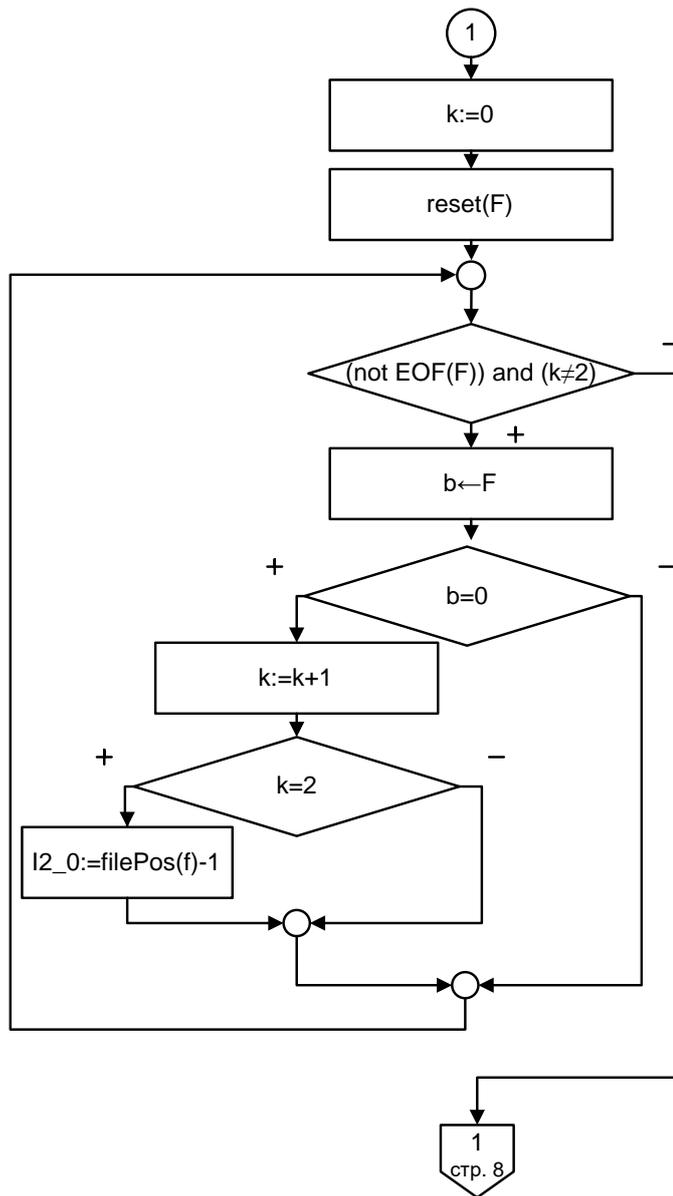
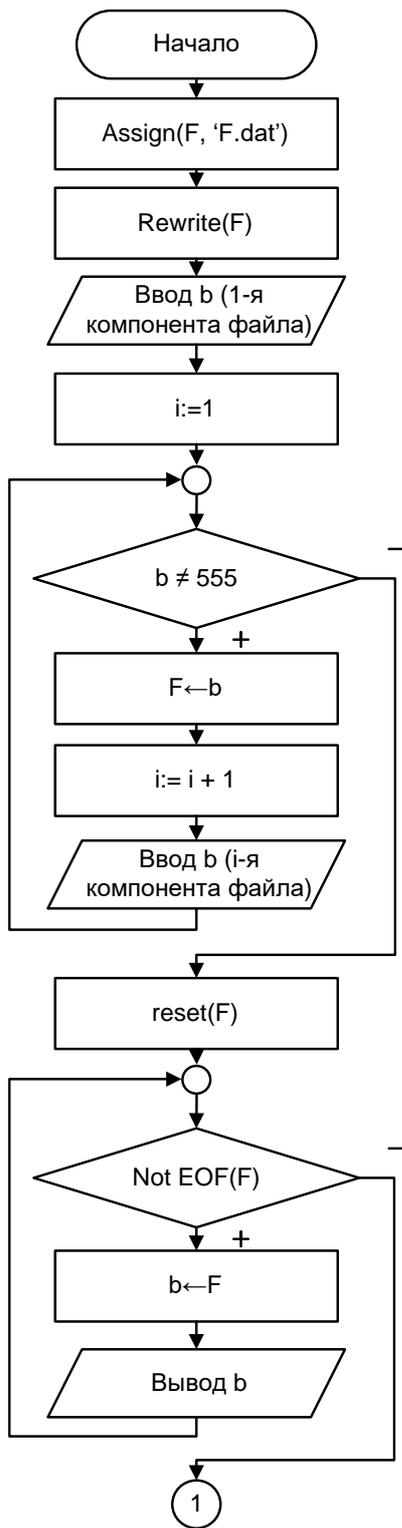
файл f до изменения:

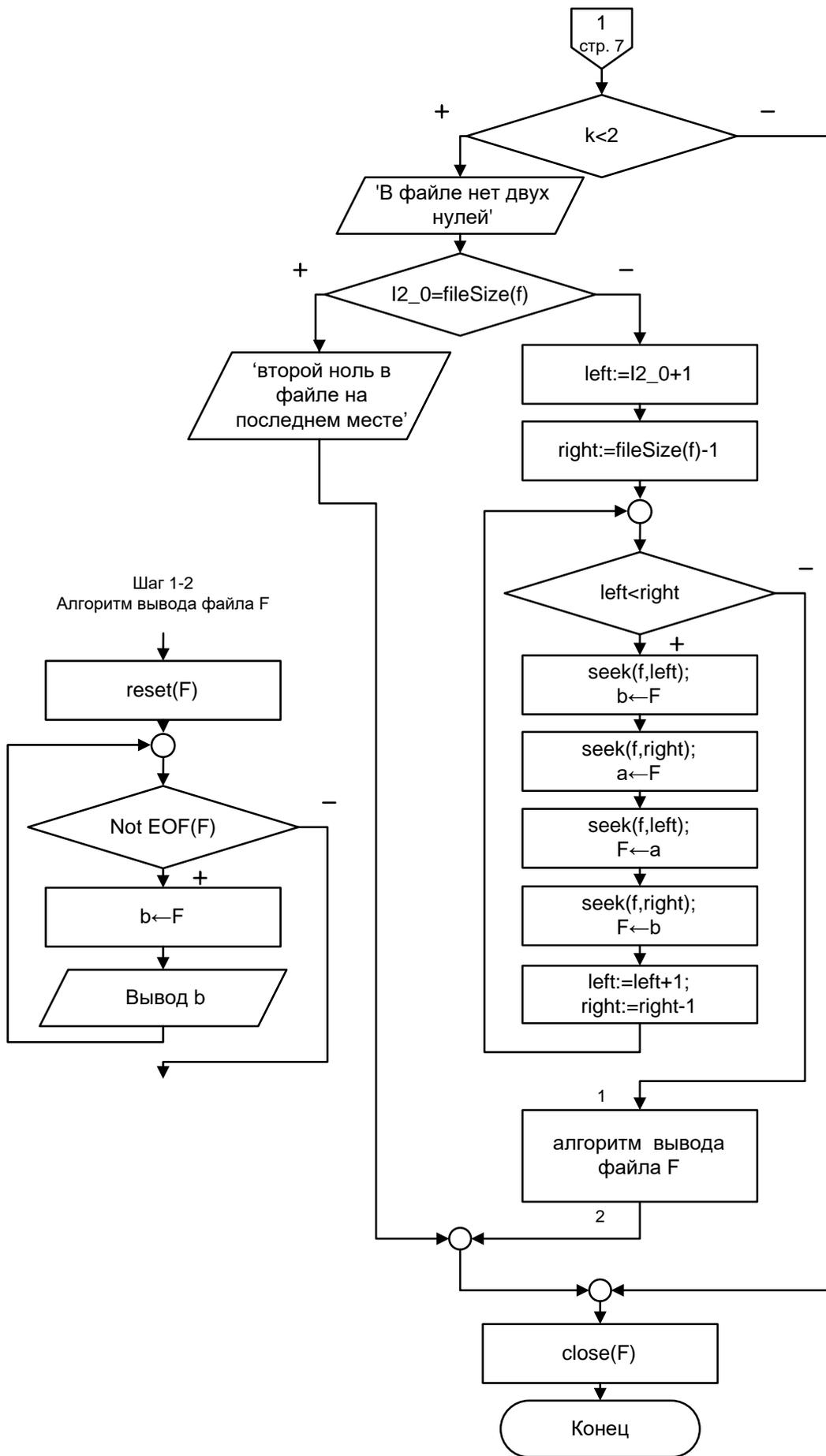
5 74 2 8 0 2 5 3 6 5 4 1 56

Выходные данные:

В файле меньше двух нулевых компонент

Блок-схема





Листинг программы на Pascal

```
program ABC;
var a,b,i,left,right,I2_0,k:integer;
    f:file of integer;
begin
    assign(f,'f.dat');
    writeln('введите первую компоненту файла f (признак конца ввода:
"555");');
    //ввод файла
    rewrite(f);
    readLn(b);
    i:=1; //номер компоненты файла
    while b<>555 do
        begin
            write(f,b);
            inc(i);
            writeln('введите ',i,'-ую компоненту файла f');
            readLn(b);
        end;
    //вывод файла до изменения
    reset(f);
    writeln('файл f до изменения:');
    while not EOF(f) do
        begin
            read(f,b);
            write(b:5);
        end;
    writeln;
    // поиск позиции второго нуля
    k:=0;
    reset(f);
    while (not EOF(f)) and (k<>2) do
        begin
            read(f,b);
            if b=0 then
                begin
                    k:=k+1;
```

```

    if k=2 then
        I2_0:=filePos(f)-1;
    end;
end;

// проверка возможности перестановки в файле
if k<2 then
    writeln('в файле нет двух нулей')
else
    if I2_0=fileSize(f) then
        writeln('второй ноль в файле на последнем месте')
    else
        begin
            // перестановка элементов файла
            left:=I2_0+1;
            right:=fileSize(f)-1;
            while left<right do
                begin
                    seek(f,left);
                    read(f,b);
                    seek(f,right);
                    read(f,a);
                    seek(f,left);
                    write(f,a);
                    seek(f,right);
                    write(f,b);
                    left:=left+1;
                    right:=right-1;
                end;
            //вывод файла после изменения
            reset(f);
            writeln('файл f после изменения:');
            while not EOF(f) do
                begin
                    read(f,b);
                    write(b:5);
                end;

```

```
writeLn;  
end;  
close(f);  
end.
```

Дополнительные сведения для выполнения работы «Обработка типизированных файлов».

Файл – именованная область на внешнем информационном носителе (диске, флэшнакопителе и т.п.), содержащая данные; это последовательность однотипных данных, расположенных в смежных ячейках памяти.

Язык *Pascal* содержит ряд достаточно продвинутых средств для работы с файлами. В *Pascal* файлы принято условно делить на *физические* и *логические*.

Физический файл – собственно файл, который содержится на носителе информации. Одной из важнейших его характеристик является наличие имени в файловой системе, т. е. адреса. Например, *C:\TP7\EXAMPLE.PAS* (имя в абсолютной адресации) или *FF\DDD.DAT* (имя в относительной адресации).

Логический файл – файловая переменная языка *Pascal*, смысл его в том, чтобы компилятор «знал» какого типа файл записан по определенному адресу и как с ним следует обращаться. Это, по сути, ссылка на физический файл. Кроме того, наличие файловой переменной облегчает запись программ, делая их более наглядными и выразительными.

Для связи логического и физического файлов существует процедура *assign*. Формат у нее следующий:

assign(*Файловая_переменная*, *путь к файлу*);

Путь к файлу – выражение строкового типа. Например, для связи файла *F* с именем *C:\Lab_1.dat* следует ввести следующую команду:

```
assign(F, 'C:\Lab_1.dat');
```

Файлы в *Pascal* бывают нескольких видов, а именно: *компонентные (типизированные), текстовые, бинарные(блоковые)*.

Бинарным называется файл состоящий из последовательности нулей и единиц без определенной структуры.

Текстовый файл – файл содержащий набор символов таблицы *ASCII*. Редактирование этого файла возможно с помощью простого текстового редактора типа *NotePad*, встроенного в ОС *Windows*.

Компонентный файл – файл состоящий из однотипных ячеек.

Комп о н е н т н ы й ф а й л п р и о б ъ я в л е н и и
в ы г л я д и т с л е д у ю щ и м о б р а з о м :

file of тип компонент;

Т и п к о м п о н е н т в ф а й л е м о ж е т б ы т ь
л ю б ы м к р о м е ф а й л о в о г о .

Если файл вновь создаваемый, то следует использовать процедуру *rewrite (Файловая_переменная)*; Если ранее в файле, на который ссылается файловая переменная, уже существовала какая-либо информация, то она автоматически стирается.

Для работы с существующим файлом следует использовать процедуру *reset (Файловая_переменная)*; При этом указатель автоматически сбрасывается на начало. После того как работа с файлом закончена, его следует закрыть при помощи процедуры *close (Файловая_переменная)*; Это команда операционной системе на освобождение системных ресурсов, а также предотвращение ошибок некорректного чтения/записи в то время, пока работа с ним не ведется.

Если обработка элементов файла ведётся от начала и до конца последовательно, без использования процедуры принудительного представления курсора, то такая обработка называется **последовательным доступом**, иначе – **произвольным доступом**.

Типовые процедуры и функции для работы с компонентными файлами:

seek(Файловая_переменная, номер позиции); - переместить указатель на произвольную файловую позицию;

filePos(Файловая_переменная); - эта функция через свое имя возвращает значение позиции, на которой в данный момент находится указатель;

fileSize(Файловая_переменная); - эта функция через свое имя возвращает значение длины файла;

read(Файловая_переменная, Читаемая_переменная_1, Читаемая_переменная_2, ...); - чтение компоненты из файла;

write(Файловая_переменная, Записываемая_переменная_1, Записываемая_переменная_2, ...); - запись компоненты в файл;

EOF(Файловая_переменная); - функция возвращает логическое значение (**TRUE** или **FALSE**), в зависимости от того, достигнут конец файла или нет;

truncate(Файловая_переменная); - усечение файла с текущей позиции.

Контрольные вопросы по теме «Обработка компонентных файлов».

1. Дайте определение файла.
2. Назовите виды файлов в *Pascal*.
3. Что такое физический и логические файлы?
4. Назовите роль указателя при работе с файлами.
5. Для чего используются процедуры *Assign, Reset, Rewrite, Close*?
6. Что такое произвольный и последовательный доступ к компонентам файла?
7. Назовите типовые процедуры и функции для работы с компонентными файлами.

Варианты заданий по теме «Обработка компонентных файлов»

1. Создать числовой файл F . Получить файл G , в который записать сначала положительные, затем отрицательные и, наконец, нулевые компоненты файла F . В файле G найти максимальный компонент. Файлы F и G вывести на печать.

2. Создать файл F целого типа. Получить два файла: $F1, F2$. В файл $F1$ последовательно писать четные числа, в файл $F2$ – все нечетные числа файла F . В файле $F2$ элементы кратные трём или пяти уменьшить на единицу. Файлы $F, F1, F2$ распечатать.

3. Создать числовой файл F . Найти сумму положительных компонент файла, расположенных до максимальной компоненты. Найденной суммой заменить отрицательные компоненты, строящие после максимума. Файл и сумму распечатать (как до, так и после преобразования).

4. Создать числовой файл F . Найти в файле сумму и произведение максимальной и минимальной компонент. Выяснить что больше. Найденным значением заменить первую и предпоследнюю компоненты файла, если сумма окажется больше. Заменить все отрицательные компоненты, если большим будет произведение. Сумму, произведение, исходный файл и файл после замены распечатать.

5. Создать файл F целого типа. Вычислить сумму четных компонент файла. Суммой заменить вторую отрицательную компоненту файла, или вывести сообщение о невозможности таковой замены. Исходный и файл после замены распечатать.

6. Создать файл F вещественных чисел. Найти максимальную компоненту и дописать её в конец файла исходную максимальную компоненту уменьшить в два раза. Далее сформировать файл H , записав в него каждую вторую компоненту файла F . Файлы распечатать.

7. Сформировать файл F целого типа. Найти и распечатать произведение максимальной и минимальной компонент файла. Произведением заменить все нулевые компоненты. Если нулевых компонент в файле нет, то поменять в нём местами максимум и минимум. Исходный и полученный файлы распечатать.

8. Создать файл F вещественных чисел. Найти SrA – среднее арифметическое элементов расположенных левее максимума. Все компоненты файла, начиная с максимальной и по последнюю включительно, увеличить на число SrA и просуммировать. Суммой заменить максимальную компоненту файла.

9. Ввести файл F , посчитать k – количество положительных компонент файла F , которые расположены на тех позициях, номера которых кратны трем. Далее все компоненты от k -й и до максимальной заменить единицами.

10. Сформировать файл F , компонентами которого являются действительные числа. Найти и распечатать сумму компонент файла, произведение отрицательных компонент файла и среднее арифметическое компонент файла. Эти три значения вписать в файл справа от его максимума. Файл вывести до и после преобразования.

11. Сформировать файл F . Определить сумму и произведение наибольшего и наименьшего из значений компонент. Заменить значением суммы первую и последнюю компоненты, а на значение произведения – увеличить каждую третью компоненту файла. Файл вывести до и после преобразования.

12. Ввести файл $F1$. Записать в файл $F2$ компоненты файла $F1$ в обратном порядке. В каждом из файлов элемент, стоящий справа от максимума продублировать значением этого максимума. Файлы вывести до и после преобразования. (При решении задачи вспомогательных массивов и файлов не использовать)

13. Ввести файлы $F1$ и $F2$. Переписать с сохранением порядка следования компоненты файла $F1$ в файл $F2$, а компоненты файла $F2$ в файл $F1$. При решении допускается использование вспомогательного файла H . Файлы вывести до и после преобразования с возведением в квадрат их максимальных компонент.

14. Сформировать два файла целых чисел одинаковой длины $F1$ и $F2$. Затем необходимо слить их в файл $F3$ таким образом, чтобы сначала шла компонента из $F1$, потом из $F2$ и т. д. В файле $F3$ поменять местами минимальную и предпоследние компоненты. Файлы вывести до и после преобразования.

15. Сформировать файл F целых чисел. Произвести «зеркалирование» файла, т.е. увеличение его длины в два раза методом добавления в конец собственных элементов в обратном порядке. В конец файла дописать произведение его нечётных и сумму чётных элементов. (При решении вспомогательных файлов и массивов не использовать)

16. Задан числовой файл F . Найти максимум среди компонент, расположенных на четных позициях файла и минимум среди компонент на нечётных позициях. Все нулевые компоненты заменить значением максимума, а значение минимума дописать в конце файла пять раз. Файл до и после преобразований распечатать.

17. Задан числовой файл F . Найти сумму компонент, стоящих между минимальной и максимальной. Заменить этой суммой все элементы из которых она сформирована (элементы между максимумом и минимумом). Файл вывести как до, так и после преобразования.

18. Дан целочисленный файл F . Все его отрицательные компоненты увеличить в два раза, положительные уменьшить в три раза, на место нулевых компонент записать их количество. Файл F распечатать как до, так и после преобразования.

19. Ввести файл F действительных чисел. Заменить в нем максимальный и минимальный компоненты на значение среднего арифметического всех положительных компонент файла. Выдать на печать файл до преобразования и после. В случае невозможности преобразования дать об этом сообщение.

20. Ввести файл целых положительных чисел $F1$. Компоненты файла $F1$, меньшие числа SrA (среднего арифметического его элементов), записать в файл $F2$. В файле $F1$ этим компонентам присвоить значение нуль. Выдать на печать файлы $F1$ и $F2$ как до, так и после преобразования.

21. Ввести с клавиатуры файл F действительных чисел. Компоненты файла, стоящие между наибольшим и наименьшим компонентом, переставить в обратном порядке. Выдать на печать исходный и преобразованный файл.

22. Ввести с клавиатуры файл F с числовыми компонентами и выдать его на печать. Определить среднее арифметическое компонентов файла и все компоненты, меньшие среднего арифметического, увеличить в два раза, а большие уменьшить в три раза. Преобразованный файл выдать на печать.

23. Ввести файл F целых чисел. Определить в нем первую компоненту, кратную пяти, и поменять её местами с минимальной по модулю компонентой в файле. Выдать на печать файл до преобразования и после. Если преобразование невозможно, то выдать об этом сообщение.

24. Ввести числовой файл F . Переставить в нём все компоненты, предшествующие минимальной в обратном порядке. Определить в изменённом файле на какой позиции находится максимум. Файл вывести как до, так и после преобразования.

25. Ввести числовой файл F , продублировать все компоненты файла, расположенные между максимумом и минимумом, дописав их в конец файла. Исходный файл, максимум, минимум и файл после преобразования вывести.

26. Ввести целочисленный файл F . Найти значение k – количество чётных элементов в файле, далее найти сумму последних k элементов. Полученным значением заменить каждую вторую компоненту файла. Вывести на экран файл как до, так и после преобразования.

27. Ввести с клавиатуры файл F действительных чисел и выдать его на печать. Определить среднее геометрическое положительных компонент файла, стоящих после третьей отрицательной компоненты, и записать его в конец файла. Выдать на печать файл после преобразования, значение вычисленного среднего геометрического или сообщение о невозможности преобразования.

28. Ввести числовой файл F . Если первая компонента файла положительна, то домножить все его компоненты на квадрат минимального элемента, если отрицательна, то домножить все компоненты на квадрат максимума. В случае нулевой первой компоненты максимум и минимум продублировать, записав их значения в конец файла. Файл до и после преобразования отпечатать.

29. В файле F заменить все нулевые компоненты значением суммы первой и последней компонент файла, все отрицательные компоненты уменьшить на значение суммы, а также найти среднее арифметическое положительных компонент, которым заменить сами положительные элементы. Файл вывести как до, так и после преобразования.

30. Ввести файл F действительных чисел и выдать его на печать. Поменять местами наибольший по модулю и первый компоненты файла. Преобразовать его, не создавая нового файла, разделив компоненты на абсолютное значение наибольшего по модулю компонента.

2. Обработка компонентных файлов и массивов

Ниже приведен пример оформления лабораторной работы.

Л. Р. № __ «Обработка компонентных файлов и массивов»	Студент	Иванов И. И.
	Группа	ХХ-999
	Дата	ДД.ММ.ГГ
	Допуск	
	Выполнение	
Вариант 88	Отчет	

Условие задачи

Ввести файл F , вывести его. Сформировать квадратную матрицу $A1_{N1 \times M1}$ максимально возможного размера из всех элементов левее максимума и матрицу $A2_{2 \times 3}$ из положительных элементов правее максимума (если формирование невозможно, то сообщить об этом). Для вывода матриц использовать подпрограммы.

Тестовый пример

Входные данные:

введенный файл F:

1 2 3 4 5 6 **100** 2 3 4 5 -3 -2 4 3 2

Выходные данные:

Матрица **A1** после формирования:

1 2

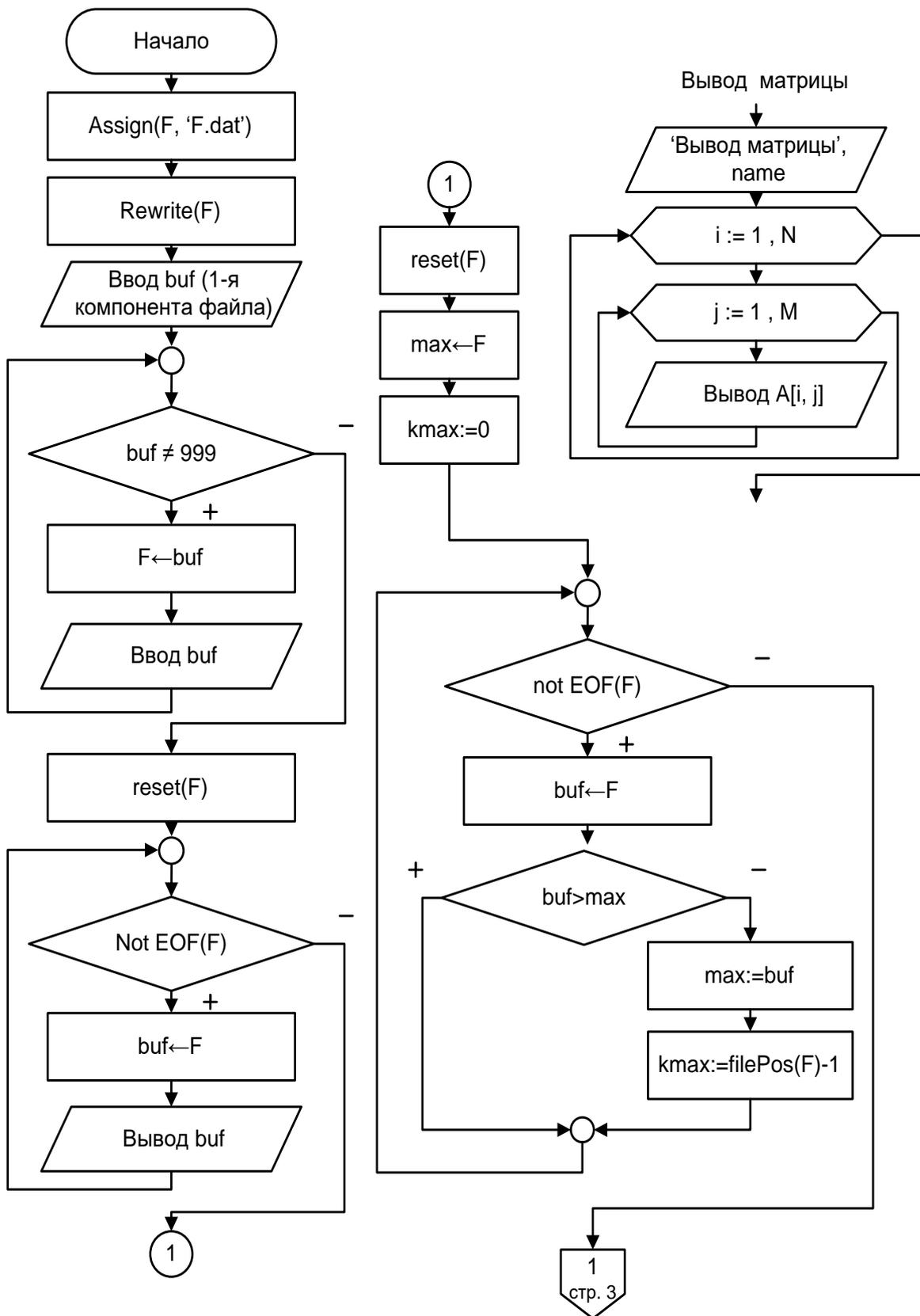
3 4

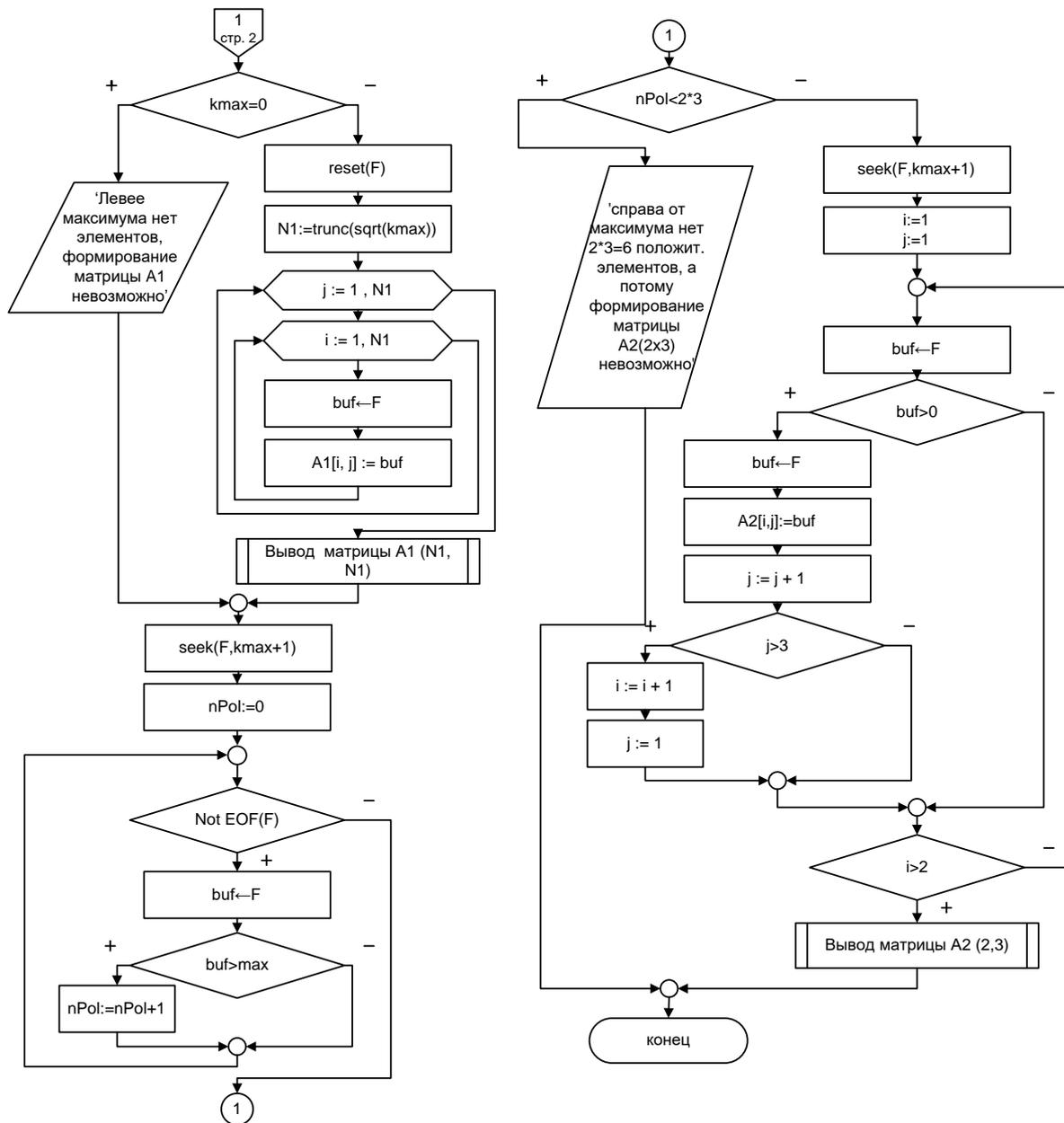
Матрица **A2** после формирования:

2 3 4

5 4 3

Блок-схема





Листинг программы на Pascal

```

program MaxMinExch;
const L=10;
type T2M = array[1.. L,1..L] of integer;
      Tname = string[4];
var A1,A2: T2M;
      i,j,N1,kmax,nPol:byte;
      buf,max:integer;
      F: file of integer;

```

```

begin
// инициализация файла
Assign(F,'F.dat');
rewrite(F);
writeln('введите первую компоненту файла');
readLn(buf);
while buf<>999 do
  begin
    write(F,buf);
    writeln('введите следующую компоненту:');
    readLn(buf);
  end;
// вывод файла
writeln('введенный файл F:');
reset(F);
while not EOF(F) do
  begin
    read(F,buf);
    write(buf:4);
  end;
writeln;
// поиск максимума
reset(F);
read(F,max);
kmax:=0;
while not EOF(F) do
  begin
    read(F,buf);
    if buf>max then
      begin
        max:=buf;
        kmax:=filePos(F)-1;
      end;
  end;
if kmax=0 then
  writeln('Левее максимума нет элементов, формирование матрицы A1
невозможно')

```

```

else
begin
  reset(F);
  N1:=trunc(sqrt(kmax));
  for i:=1 to N1 do
    for j:=1 to N1 do
      begin
        read(f,buf);
        A1[i,j]:=buf;
      end;
    writeln('Матрица A1 после формирования:');
  for i:=1 to N1 do
    begin
      for j:=1 to N1 do
        write(A1[i,j]:4);
      writeln;
    end;
  end;
  seek(F,kmax+1);
  nPol:=0;
  while not EOF(F) do
    begin
      read(F,buf);
      if buf>0 then
        inc(nPol);
    end;
    if nPol<2*3 then
      begin
        writeln('справа от максимума нет 2*3=6 пол. элементов,');
        writeln('а потому формирование матрицы A2(2x3) невозможно');
      end
    else
      begin
        seek(F,kmax+1);
        i:=1;
        j:=1;
        repeat

```

```

read(F,buf);
if buf>0 then // если элемент >0, то его в матрицу
  begin
    A2[i,j]:=buf;
    inc(j);
    if j>3 then
      begin
        inc(i);
        j:=1;
      end;
    end;
  until i>2;
  writeln('Матрица A2 после формирования:');
for i:=1 to 2 do
  begin
    for j:=1 to 3 do
      write(A2[i,j]:4);
    writeln;
  end;
end;
close(F);
end.

```

Контрольные вопросы по теме «Обработка компонентных файлов и массивов».

1. Напиши фрагмент программы создания файла.
2. Напиши фрагмент программы вывода файла на печать.
3. Как установить указатель в конец файла?
4. Как удалить в файле компоненты, начиная с номера N?
5. Какая процедура используется для прямого доступа к компонентам файла?
6. Напишите фрагменты программ копирования элементов файла в одномерный массив, и наоборот.

Варианты заданий по теме «Обработка компонентных файлов и массивов»

1. Ввести целую квадратную матрицу $A(N,N)$ ($N \leq 10$). Из четных ее элементов сформировать файл F. Из файла F сформировать файл G, в который сначала записать сумму всех элементов файла F, затем максимальное значение файла F, затем все его компоненты в обратном порядке.

2. Создать файл F вещественных чисел. Получить вектор B, записывая в него каждую третью компоненту файла, читая файл с начала, и вектор A, читая файл с конца. Файл и полученные векторы распечатать. Задан числовой файл F. Определить произведение первых N ненулевых компонент файла и сумму последних N компонент. Сформировать из файла вектор B(10), в конец которого дописать сумму, а затем произведение. Вектор распечатать.

Если формирование вектора невозможно, то сделать об этом сообщение.

3. Ввести с клавиатуры квадратную матрицу X (M,M) ($M \leq 10$). Определить в ней максимальные элементы каждой строке и записать вычисленные значения в файл F1. Определить минимальную компоненту в файле F1. Выдать на печать матрицу X, файл F1, минимальную компоненту и ее номер.

4. Из компонент файла F1, введенного с клавиатуры, создать квадратную матрицу X максимально возможного размера. Определить в матрице X сумму элементов каждого столбца и записать вычисленные суммы в конец файла. Выдать на печать файл F1 до преобразования и после, а также матрицу X в виде таблицы.

5. Сформировать файл F целого типа. Найти S — сумму первых трех положительных компонент файла. Если $S > 10$, то получить из файла одномерный массив A, записывая в него каждую вторую компоненту файла. Массив распечатать. Если $S \leq 10$, то каждую компоненту файла увеличить на величину S . Файл распечатать.

6. Сформировать матрицу $A(N, M)$ ($N \leq 10$, $M \leq 15$). Получить из матрицы файл, записывая в него сначала строку, затем столбец, содержащие максимальный элемент матрицы. Файл распечатать.

7. Сформировать файл F целого типа. Из файла получить матрицу $A(N, M)$. Начиная с 1-ой и по $(N-1)$ -ую строки формировать, читая компоненты файла с начала, N -ую строку заполнить компонентами, читая файл с конца. Матрицу распечатать. Если матрицу сформировать нельзя, то сделать сообщение. Значения N , M вводятся с клавиатуры.

8. Создать файл F вещественных чисел. Сформировать из файла матрицу $A(N, N)$, заполняя ее строками. Если для формирования матрицы не хватает компонент файла, то дописать числа $1, 2, 3, \dots, n$. Матрицу распечатать.

9. Создать числовой файл F. Найти сумму всех компонент и произведение отрицательных компонент файла. Получить из него два вектора: A и B. В вектор A последовательно записать положительные компоненты файла и в конец — полученную сумму, в вектор B записать сначала вычисленное произведение, а далее — отрицательные компоненты файла. Вектора A, B распечатать

10. В заданной целочисленной матрице $A(N, M)$ ($N \leq 10$, $M \leq 15$) строки, не содержащие ни одного нуля, переписать последовательно в файл F. Если файл не сформирован, сделать текстовое сообщение.

11. Дана матрица $A(N,N)$ ($N \leq 10$, $M \leq 15$), получить файл F, записывая в него минимальные элементы каждого столбца, в конец файла записать последний элемент 2^{ой} строки и первый элемент последней строки. В файле вычислить сумму каждой второй компоненты и записать ее на место 2^{ой} компоненты. Файл распечатать.

12. Элементы файла F поместить на главную и побочную диагональ матрицы $D(N,N)$ ($N \leq 10$). Отрицательные элементы полученного массива дописать в файл F. Файл и матрицу вывести на экран.

13. Создать числовой файл F1. Из положительных компонент файла F1 получить квадратную матрицу $A(N,N)$ возможного размера. В конец файла записать сумму элементов главной и побочной диагонали. Файл и матрицу напечатать.

14. Сформировать матрицу $C(N,N)$ ($N \leq 10$). Строки с отрицательным элементом на главной диагонали последовательно записать в файл F1. В конец файла F1 дописать сумму первой и последней компонент, а затем 1^{ой} и последний элемент матрицы C. Файл F1 распечатать.

15. Ввести целую квадратную матрицу $A(N,N)$ ($N \leq 10$). Из положительных ее элементов сформировать файл f, причем формирование файла вести, рассматривая матрицу по столбцам. После записи столбца матрицы дописывать в файл количество положительных элементов в столбце и среднее арифметическое их значение. Файл F распечатать.

16. Сформировать матрицу $A(N,N)$ ($N \leq 10$) вещественного типа. Найти суммы элементов главной и побочной диагоналей. Записать в файл F1 сначала элементы диагонали, имеющей большее значение суммы, а затем элементы последней строки матрицы. В файле F1 каждую вторую компоненту, начиная с середины файла, уменьшить вдвое. Файл F1 распечатать.

17. Дана матрица $X(N,M)$ ($N \leq 10$, $M \leq 15$). Строки матрицы, которые начинаются с положительного элемента, последовательно писать в файл F1. В конец файла дописать сумму максимальной и минимальной компоненты файла, а также первый и последний элементы матрицы. Файл F1 распечатать.

18. Ввести две матрицы $A(N,N)$, $B(K,K)$ ($N \leq 10$, $K \leq 15$). Матрицу с наименьшим минимальным элементом записать в файл F1. Запись выполнять по строкам. В конец файла записать минимальный элемент оставшейся матрицы. В файле вычислить среднее арифметическое всех компонент. Файл, среднее арифметическое напечатать.

19. Сформировать квадратную матрицу $A(M,M)$ ($M \leq 15$). Создать файл F, записывая в него последовательно строки с отрицательным элементом на главной диагонали. Если файл не сформирован, то сделать сообщение. Если файл сформирован, то удалить из файла 1-ую компоненту и распечатать.

20. Сформировать файл F целого типа, вывести его. Определить в нем k_1 – количество чётных и k_2 – количество нечётных компонент. Если $k_1 > k_2$, то сформировать квадратную матрицу $A_{N \times N}$ максимально возможного размера из первых k_1 компонент, а в противном случае из первых k_2 компонент файла.

21. Сформировать файл F целого типа, вывести его. Найти P – произведение первых k компонент (k – количество кратных трём компонент файла). Из компонент файла F сформировать матрицу $A_{3 \times 4}$, читая файл с начала, если k чётно и матрицу $A_{4 \times 3}$, если k нечётно. Если матрицу создать невозможно, сделать сообщение.

22. Сформировать файл F целого типа, вывести его. Найти сумму последних k компонент файла (k – количество элементов между максимумом и минимумом файла). Компоненты первой половины файла переписать в вектор $B1$, а отрицательные компоненты второй половины в вектор $B2$.

23. Сформировать матрицу $A1_{N1 \times M1}$, вывести её. Получить файл F , записывая в него элементы нечетных строк матрицы. В файле F вычислить S – сумму отрицательных компонент. Если сумма окажется четной, то из элементов файла сформировать квадратную матрицу $A2_{N2 \times N2}$ максимально возможного размера.

24. Сформировать матрицу $A1_{N1 \times M1}$, вывести её. Получить файл $F1$, записывая в него элементы нечетных столбцов матрицы и файл $F2$, записывая элементы четных. В матрице угловые элементы заменить произведением максимума первого файла и минимума второго. Ввести целую квадратную матрицу $A_{N \times N}$, вывести ее. Из положительных элементов матрицы сформировать файл F , который вывести. Далее из файла F сформировать файл G , в который сначала записать сумму всех элементов файла F , затем максимальное значение файла F , затем все его компоненты в обратном порядке.

25. Создать файл F вещественных чисел. Получить вектор B , записывая в него каждую третью компоненту файла, читая файл с начала, и вектор A , читая файл с конца. Файл и полученные векторы распечатать.

26. Ввести с клавиатуры квадратную матрицу $A_{N \times N}$, вывести её. Создать файлы F и G следующим образом: в файл G сначала записать элементы главной диагонали, затем побочной, в файл F по строкам вписать элементы матрицы, исключая диагональные. В конец файлов приписать удвоенную сумму всех элементов матрицы.

27. Ввести с клавиатуры квадратную матрицу $A_{N \times N}$, вывести её. Определить в матрице максимальные элементы каждой строки и записать их значения в файл F . Определить минимальную компоненту в файле F , которую утроить.

28. Из компонент файла F , введенного с клавиатуры, создать квадратную матрицу $A_{N \times N}$ максимально возможного размера. Определить в матрице сумму элементов каждого столбца и записать вычисленные суммы в конец файла. Выдать файл F до преобразования и после, а также матрицу в виде таблицы.

29. Сформировать матрицу B (M, M) ($M \leq 15$). Создать файл F , записывая в него последовательно столбцы с четным элементом на побочной диагонали. Если файл не сформирован, то сделать сообщение. Если файл сформирован, то удалить из файла максимальную компоненту и распечатать.

30. Ввести с клавиатуры матрицу $A_{N \times N}$, вывести её. Определить в матрице сумму отрицательных элементов каждой строки и записать их значения в файл F . Определить минимальную компоненту в файле F , которую записать вместо максимальной компоненты файла и матрицы.

3. Построение подпрограмм

Ниже приведен пример оформления лабораторной работы.

Л. Р. № __	Студент	Иванов И. И.
«Построение подпрограмм»	Группа	ХХ-999
	Дата	дд.мм.гг
	Допуск	
	Выполнение	
Вариант 88	Отчет	

Условие задачи

Ввести двумерные массивы $X1(N1, N1)$, $X2(N2, N2)$, $X3(N3, N3)$. Найти наименьший из максимумов этих массивов. Значением найденного минимального максимума заменить элементы побочных диагоналей всех матриц. Далее переписать в одномерный массив Y главную диагональ той матрицы, в которой максимум окажется наибольшим (*при решении реализовать процедуры ввода и вывода массивов: $Vvod2m(A, N, Name)$, $Vivod2m(A, N, Name)$, $Vivod1m(A, N, Name)$; $Form(A, B, N)$ – процедуру формирования из двумерного массива A одномерного массива B ; $Zamena(A, N, Chislo)$ – замена побочной диагонали значением $Chislo$; и функцию $Maximum(A, N, M)$: $\langle \text{тип матрицы} \rangle$ – максимум в матрице; а также функцию $min(x, y)$: $\langle \text{тип матрицы} \rangle$ – минимум из пары чисел x, y)*

Тестовые примеры

Входные данные:

$N1 = 2$

	1	2
X1=	3	4

N2=3

	-2	-4	5
X2=	6	-7	9
	10	1	-6

N3=3

	-9	4	-9
X3=	1	16	5
	1	13	34

Максимум в X1 = 4; Максимум в X2 = 10; Максимум в X3 = 34.

Наименьший из этих максимумов в массиве X1, он равен 4.

Выходные данные № 1

Результаты преобразования массивов путем замены значением найденного минимального максимума элементов побочных диагоналей всех матриц:

	1	4
X1=	4	4

	-2	-4	4
X2=	6	4	9
	4	1	-6

	-9	4	4
X3=	1	4	5
	4	13	34

Следующий шаг решения:

Наибольший из этих максимумов в массиве X3, он равен 34.

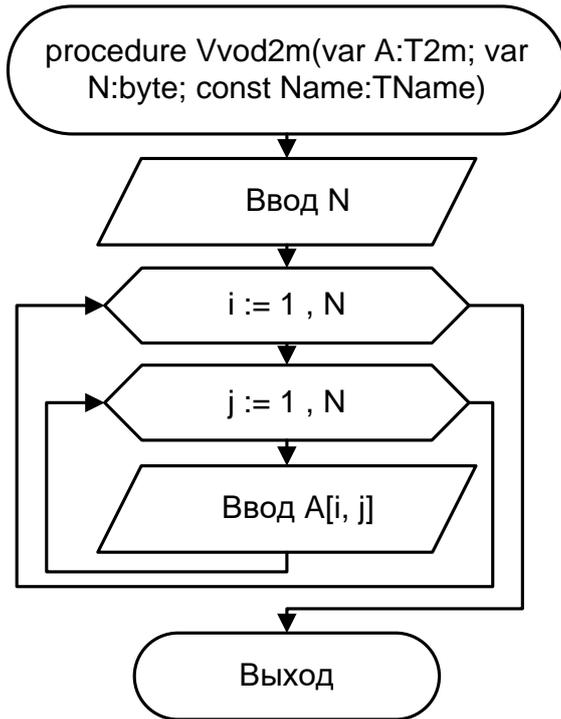
Выходные данные № 2

Формирование одномерного массива Y путем записи главной диагонали матрицы $X3$.

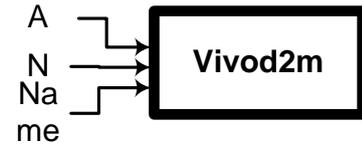
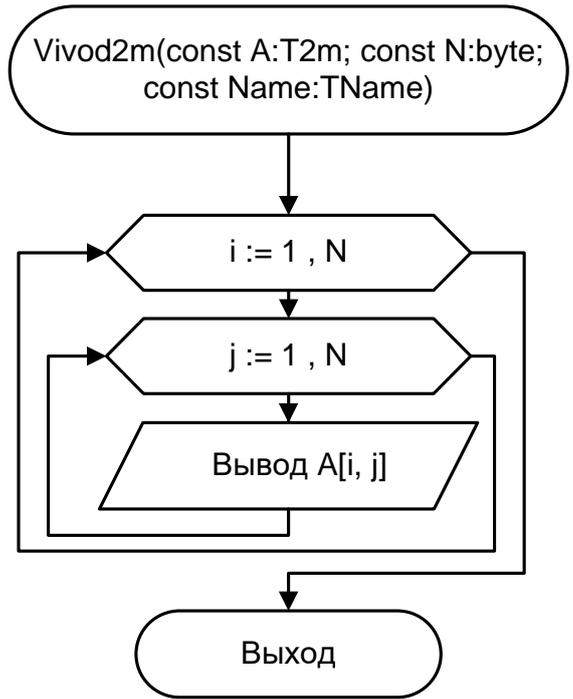
$$Y = -9 \ 4 \ 34$$

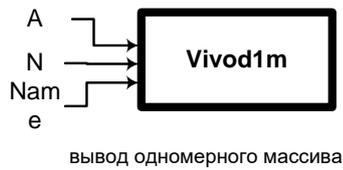
Блок-схема

ввод двумерного массива

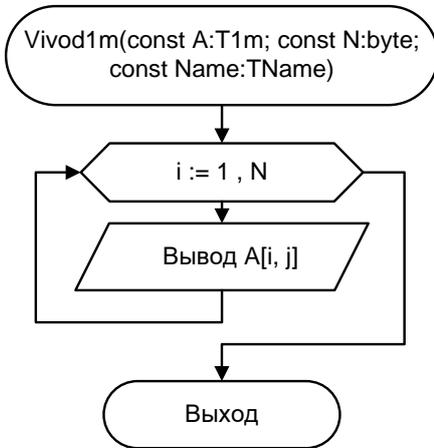


вывод двумерного массива

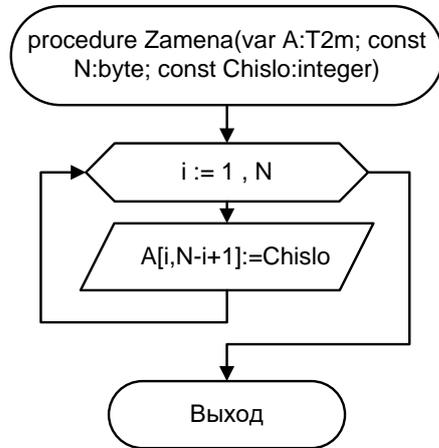




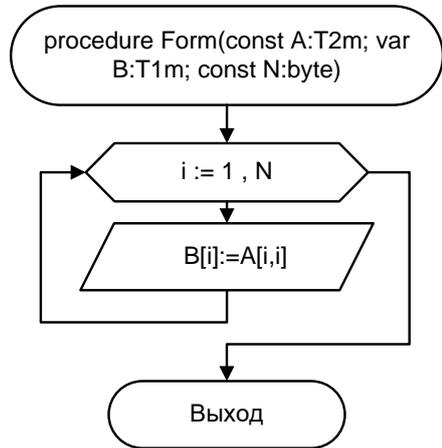
вывод одномерного массива



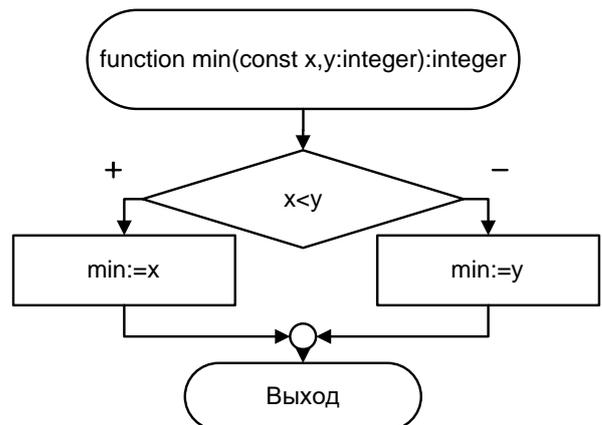
замена элементов побочной диагонали матрицы



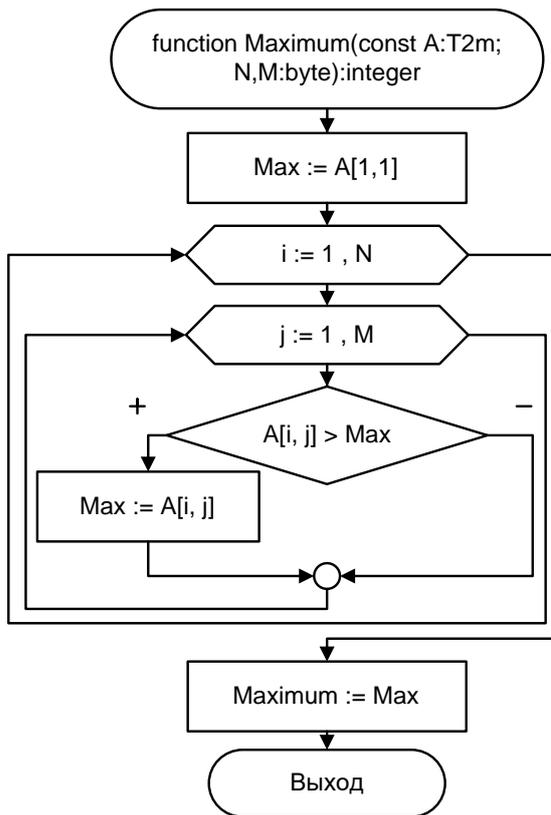
формирование одномерного массива B из двумерного A



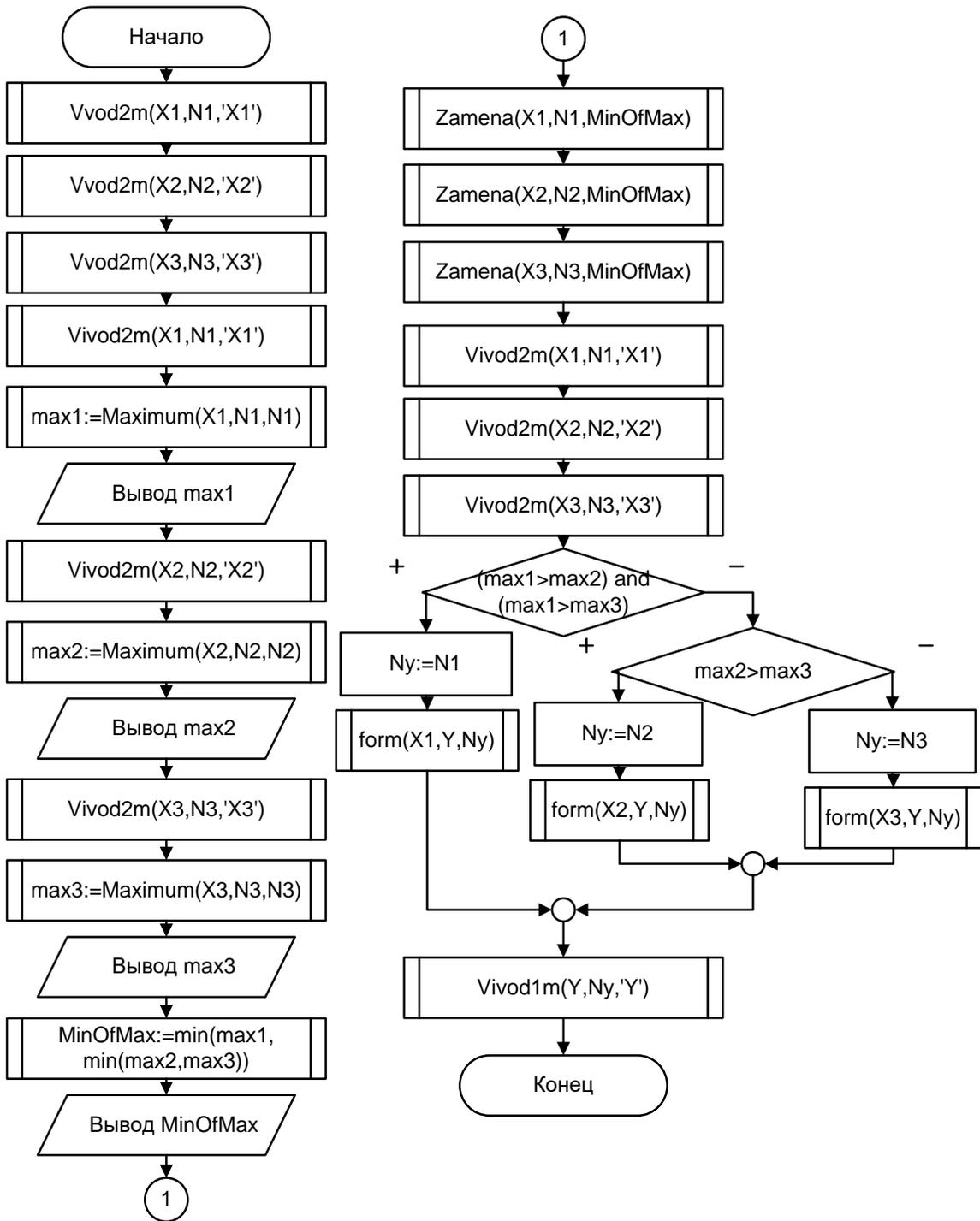
поиск наименьшего из пары чисел



поиск максимума в двумерном массиве



Основная программа



Листинг программы на Pascal

```
program Z19;
const MaxRazm = 10;
type T1m=array[1..sqr(MaxRazm)] of integer;
    T2m=array[1..MaxRazm,1..MaxRazm] of integer;
    TName=string[5];
var X1,X2,X3:T2m;
    Y:T1m;
    N1,N2,N3,Ny:byte;
    max1,max2,max3,MinOfMax,MaxOfMax:integer;
// ввод двумерного массива
procedure Vvod2m(var A:T2m; var N:byte; const Name:TName);
var i,j:byte;
begin
    writeln('вводите массив ', Name, ':');
    write('размерность массива N=');
    readLn(N);
    for i:=1 to N do
        for j:=1 to N do
            begin
                write(Name,['i,',j,']=');
                readLn(A[i,j]);
            end;
    end;
// вывод двумерного массива
procedure Vivod2m(const A:T2m; const N:byte; const Name:TName);
var i,j:byte;
begin
    writeln('выводим массив ', Name, ':');
    for i:=1 to N do
        begin
            for j:=1 to N do
                write(A[i,j]:5);
            writeln;
        end;
    end;
// поиск максимума в двумерном массиве
```

```

function Maximum(const A:T2m; N,M:byte):integer;
var max:integer;
    i,j:byte;
begin
    max:=A[1,1];
    for i:=1 to N do
        for j:=1 to M do
            if A[i,j]>max then
                max:=A[i,j];
        Maximum:=max;
    end;
// поиск наименьшего из пары чисел
function min(const x,y:integer):integer;
begin
    if x<y then
        min:=x
    else
        min:=y;
    end;
// замена элементов побочной диагонали матрицы
procedure Zamena(var A:T2m; const N:byte;
const Chislo:integer);
var i:byte;
begin
    for i:=1 to N do
        A[i,N-i+1]:=Chislo;
    end;
// формирование одномерного массива B из матрицы A
procedure Form(const A:T2m; var B:T1m; const N:byte);
var i:byte;
begin
    for i:=1 to N do
        B[i]:=A[i,i];
    end;
// вывод одномерного массива
procedure Vivod1m(const A:T1m; const N:byte; const Name:TName);
var i:byte;

```

```

begin
  writeln('Выводим одномерный массив ', Name, ':');
  for i:=1 to N do
    write(A[i]:5);
  writeln;
end;
begin
  // вводим массивы
  Vvod2m(X1,N1,'X1');
  Vvod2m(X2,N2,'X2');
  Vvod2m(X3,N3,'X3');
  // выводим массивы до преобразования
  // ищем максимумы в массивах
  Vivod2m(X1,N1,'X1');
  max1:=Maximum(X1,N1,N1);
  writeln('max1=',max1);
  Vivod2m(X2,N2,'X2');
  max2:=Maximum(X2,N2,N2);
  writeln('max2=',max2);
  Vivod2m(X3,N3,'X3');
  max3:=Maximum(X3,N3,N3);
  writeln('max3=',max3);
  // ищем минимальный максимум
  MinOfMax:=min(max1,min(max2,max3));
  writeln('минимальный максимум равен ',MinOfMax);
  // заменяем побочные диагонали в матрицах
  Zamena(X1,N1,MinOfMax);
  Zamena(X2,N2,MinOfMax);
  Zamena(X3,N3,MinOfMax);
  // выводим матрицы после замены
  Vivod2m(X1,N1,'X1');
  Vivod2m(X2,N2,'X2');
  Vivod2m(X3,N3,'X3');
  // формирование одномерного массива
  if (max1>max2) and (max1>max3) then
    begin
      Ny:=N1;

```

```

    form(X1,Y,Ny);
end
else
if max2>max3 then
begin
    Ny:=N2;
    form(X2,Y,Ny);
end
else
begin
    Ny:=N3;
    form(X3,Y,Ny);
end;
// выводим одномерный массив
Vivod1m(Y,Ny,'Y');
end.

```

Дополнительные сведения для выполнения работы «Построение подпрограмм».

Подпрограмма – это снабженный заголовком внутренний программный блок, расположенный в разделе описаний внешнего программного блока или программы.

Подпрограммы делятся на **процедуры** и **функции**.

Функция – подпрограмма языка *Pascal*, реализующая некоторый алгоритм,

результатом которого является формирование некоторого единственного значения. Обращение к функции происходит через ее имя. Функция всегда возвращает, как минимум, один параметр, через свое имя, например, `function sin(x:real):real`.

Процедура – подпрограмма языка *Pascal*, предназначенная для формирования нескольких значений и/или выполнения некоторых действий не связанных напрямую с изменением значений параметров. Заголовок выглядит следующим образом: ***procedure*** имя_процедуры (список формальных параметров).

Подпрограммы описываются в разделе описаний программы до исполняемой части с любым порядке, а вызываются в исполняемой части программы (в разделе операторов) в строго определенном порядке.

Идентификаторы, описанные в подпрограмме, являются **локальными** для нее, т.е. работа с ними возможна только внутри этой подпрограммы и внутри вложенных в нее блоков.

Имена, описанные в модулях более высокого уровня, являются **глобальными** для всех своих подчиненных. Эти имена могут быть использованы в любом модуле стоящем ниже на иерархической лестнице, а также в исполнительной части самого модуля.

Для того чтобы подпрограмма могла быть использована многократно для разных наборов входных и выходных параметров, используют наборы формальных и фактических параметров.

Формальные параметры – это локальные переменные, необходимые для описания алгоритма подпрограммы, они описываются в ее заголовке и используются в собственном разделе операторов.

Формальные параметры указываются в скобках справа от названия подпрограммы *при ее описании*. Потому описание формальных параметров происходит только один раз.

Фактические параметры – это набор данных, в обработке которых и заключается предназначение алгоритма. В момент вызова формальные параметры заменяются фактическими во всей подпрограмме. Фактические параметры – все то, что указывается в скобках справа от названия процедуры или функции *при ее вызове*. Фактические параметры у подпрограммы могут меняться при каждом вызове, а формальные нет.

Ф о р м а л ь н ы е и ф а к т и ч е с к и е
п а р а м е т р ы д о л ж н ы с о в п а д а т ь :

- *по количеству*, т.е. количество заявленных и реально используемых переменных должно совпадать;
- *по типу*, т.е. тип заявленных и реально используемых переменных должен совпадать;
- *по порядку следования*, т.е. переменные в описании подпрограммы и при ее вызове должны быть перечислены в одинаковом порядке.

- *по способу передачи*, т.е. статус параметров в главной программе должен быть совместимым с заявленным статусом параметров подпрограммы.

Формальные параметры делятся на:

- **параметры-константы** – параметры, переданные по ссылке без права их изменения. Параметры-константы используются, когда передаются входные данные, являющиеся сложными структурированными переменными (например, массивы). При таком способе передачи изменение формального параметра запрещено, если переданный параметр будет изменяться, компилятор выдаст ошибку;
- **параметры-переменные** – параметры, переданные по ссылке с правом их изменения. Параметры-переменные используются для передачи выходных значений процедур. При изменении параметров-

переменных изменяется соответствующий фактический параметр, таким образом, изменения сохраняются и после завершения работы подпрограммы. Для использования этого способа передачи, в списке формальных параметров перед параметром- переменной ставится префикс *var*. Следует отметить, что переменная типа *file* всегда описывается как параметр-переменная.

При передаче структурированных типов (файлов, массивов, записей и т.д.) необходимо создавать новый тип в разделе описаний типов *type*.

Контрольные вопросы по теме «Потроение подпрограмм».

1. Назовите преимущества использования подпрограмм.
2. Чем подпрограмм-процедура отличается от подпрограммы-функции?
3. Назовите назначение формальных и фактических параметров. Где они располагаются?
4. Назовите критерии соответствия формальных и фактических параметров. Должны ли они совпадать по идентификаторам?
5. Что такое и где описываются локальные и глобальные параметры?
6. Чем параметры-константы отличаются от параметров-переменных?

Варианты заданий по теме «Построение подпрограмм»

Обработка массивов с использованием подпрограмм

1. Ввести одномерные массивы $X1_{N_1}$, $X2_{N_2}$ и $X3_{N_3}$. Заменить в каждом из них максимальный элемент средним арифметическим положительных элементов (если оно существует) соответствующего массива. Вывести массивы до преобразования и после. (При решении реализовать процедуры ввода и вывода массивов **Vvod1m(A,N,Name)** и **Vivod1m(A,N,Name)**, а также функции **IndMax(A,N):byte** – поиск индекса максимума и **SrAPol(A,N):real** – поиск среднего арифметического положительных элементов (равно нулю, если положительных нет))

2. Ввести матрицы X_{NxMx} и Y_{NyMy} . В матрицах найти максимумы, которые вывести. Преобразовать каждую матрицу, уменьшив её положительные элементы на значение максимума другой матрицы. Исходные и преобразованные матрицы вывести на экран. (При решении реализовать процедуры ввода и вывода массивов: **Vvod2m(A,N,M,Name)**, **Vivod2m(A,N,M,Name)** и **Zamena(A,N,M,Vichet)** – уменьшение положительных элементов матрицы на величину **Vichet**, а также функцию **Maximum(A,N,M):<тип массива>** – поиск максимума)

3. Ввести одномерные массивы $X1_{N_1}$, $X2_{N_2}$ и $X3_{N_3}$. Определить в каждом из них среднее арифметическое отрицательных элементов. Если среднее арифметическое отрицательных существует, то поместить его в чётные ячейки соответствующего массива. Вывести массивы до преобразования и после. (При решении реализовать процедуры ввода и вывода массивов **Vvod1m(A,N,Name)** и **Vivod1m(A,N,Name)**, функцию **SrArOtr(A,N):real** – поиск среднего арифметического отрицательных, которая равна нулю если отрицательных нет и процедуру **ZamenaChet(A,N,Chislo)** – замена чётных ячеек значением **Chislo**)

4. Ввести квадратные матрицы $X1_{N_1 \times N_1}$, $X2_{N_2 \times N_2}$ и $X3_{N_3 \times N_3}$. Найти минимальное из 3-х чисел: $Tr1$, $Tr2$, $Tr3$, каждое из которых является следом соответствующей матрицы (суммой диагональных

элементов). В матрицах поменять местами первую строку и побочную диагональ. Матрицы вывести до и после преобразования. (При решении реализовать процедуры ввода и вывода массивов: $Vvod2m(A,N,Name)$, $Vivod2m(A,N,Name)$, а также функции $Trace(A,N):<min\ массива>$ – поиск следа и $Min(p, q):<min\ массива>$ – минимум среди двух чисел; $Zamena(A,N)$ – замена первой строки и побочной диагонали в матрице)

5. Ввести двумерные массивы $X1_{N1 \times M1}$ и $X2_{N2 \times M2}$. Найти наибольший элемент в каждом из массивов. В каждой матрице записать в строку, содержащую этот элемент нули, исключая сам максимальный элемент. Вывести на экран матрицы до преобразования и после. (При решении реализовать процедуры ввода и вывода массивов: $Vvod2m(A,N,M,Name)$, $Vivod2m(A,N,M,Name)$ и $ZamenaStr(A,Iz,M,Jz)$ – замена нулями элементов строки Iz , кроме элемента расположенного в столбце Jz ; а также процедуру $Maximum(A,N,M,Imax,Jmax)$ – поиск координат максимума)

6. Ввести одномерные массивы $X1_{N1}$, $X2_{N2}$ и $X3_{N3}$. Разделить каждый из элементов на максимальный элемент среди отрицательных элементов соответствующего массива. Массивы вывести на экран до и после преобразования (или сообщение о невозможности преобразования). (При решении реализовать процедуры ввода и вывода массивов: $Vvod1m(A,N,Name)$ и $Vivod1m(A,N,Name)$; функцию $MaxOtr(A,N):<min\ массива>$ – максимум среди отрицательных, которая равна нулю если отрицательных нет; и процедуру $Delenie(A,N,Chislo)$ – деление элементов массива на $Chislo$)

7. Ввести двумерные массивы $X1_{N1 \times N1}$, $X2_{N2 \times N2}$. Заменить в каждом массиве элементы главной диагонали на произведение положительных элементов противоположной матрицы. Матрицы вывести на экран до и после преобразования (если преобразование невозможно, то

вывести соответствующее сообщение). (При решении реализовать процедуры ввода и вывода массивов: $Vvod2m(A,N,Name)$, $Vivod2m(A,N,Name)$, а также функцию $PrPol(A,N):<mun\ массива>$ – поиск произведения положительных (равна нулю при их отсутствии) и процедуру $ZamenaDiag(A,N,Chislo)$ – замена диагонали значением $Chislo$)

8. Ввести двумерные массивы $X1_{N1 \times M1}$, $X2_{N2 \times M2}$. Переписать элементы строки матрицы, содержащей минимальный элемент, соответственно, в одномерные массивы $Y1_{M1}$ и $Y2_{M2}$. Обе матрицы и массивы вывести на экран. (При решении реализовать процедуры ввода и вывода массивов: $Vvod2m(A,N,M,Name)$, $Vivod2m(A,N,M,Name)$, $Vivod1m(A,N,Name)$, а также функцию $IndMin(A,N,M):byte$ – индекс строки, содержащей минимум и процедуру $FormFromStr(A,B,M,Str)$ – перепись из A в B строки Str)

9. Ввести одномерные массивы $X1_N$, $X2_N$. Определить новый вектор-массив $X3_N$ как: $X3 = X1^* + X2^*$, где знак «*» означает, что все компоненты этого вектора получены из исходного путём его деления на максимальную компоненту (производится нормировка). (При решении реализовать процедуры ввода и вывода массивов: $Vvod1m(A,N,Name)$ и $Vivod1m(A,N,Name)$; функцию $MaxOtr(A,N):<mun\ массива>$ – максимум массива; процедуру $Delenie(A,N,Chislo)$ – деление элементов массива на значение $Chislo$; процедуру $Form(A1,A2,A3,N)$ – формирование третьего массива из пары исходных)

10. Ввести двумерные массивы $X1_{N1 \times M1}$, $X2_{N2 \times M2}$. Если максимальный элемент массива $X1$ больше максимального элемента массива $X2$, то заменить им – все отрицательные элементы обеих матриц. В противном случае – все положительные элементы. Массивы до и после преобразования вывести на экран. (При решении реализовать процедуры ввода и вывода массивов: $Vvod2m(A,N,M,Name)$, $Vivod2m(A,N,M,Name)$, а

также функцию $Maximum(A,N,M):<min\ массива>$ – максимум массива и процедуры $ZamenaPol(A,N,M,Chislo)$, $ZamenaOtr(A,N,M,Chislo)$ – замена, соответственно, положительных и отрицательных элементов значением $Chislo$)

11. Ввести двумерные массивы $X1_{N1 \times M1}$, $X2_{N2 \times M2}$. Переписать элементы столбцов обеих матриц, содержащих максимальные элементы в своих матрицах в одномерный массив Y_{N1+N2} . Обе матрицы и массив Y вывести на экран. (При решении реализовать процедуры ввода и вывода массивов: $Vvod2m(A,N,M,Name)$, $Vivod2m(A,N,M,Name)$, $Vivod1m(A,N,Name)$, а также функцию $IndMin(A,N,M):byte$ – индекс столбца, содержащего минимум и процедуру $Form(A1,A2,B,N1,N2,Stlb1,Str2)$ – перепись из $A1$, $A2$ в одномерный массив B столбцов $Stlb1$, $Stlb2$)

12. Ввести двумерные массивы $X1_{N1 \times N1}$, $X2_{N2 \times N2}$, $X3_{N3 \times N3}$. Заменить в них элементы побочной диагонали на произведение положительных и кратных 3 элементов соответствующей матрицы. Исходные и преобразованные матрицы вывести на экран (или сообщение о невозможности преобразования). (При решении реализовать процедуры ввода и вывода массивов: $Vvod2m(A,N,Name)$, $Vivod2m(A,N,Name)$; функцию $PrKr3Pol(A,N):integer$ – поиск произведения положительных и кратных трём (если произведение не существует, то функция равна нулю); процедуру $ZamenaDiag(A,N,Chislo)$ – замена побочной диагонали значением $Chislo$)

13. Ввести двумерные массивы $X1_{N1 \times M1}$, $X2_{N2 \times M2}$. В этих массивах найти минимальные элементы. Преобразовать каждую матрицу, увеличив её отрицательные элементы на значение минимального по модулю элемента другой матрицы. Исходные и преобразованные матрицы вывести на экран. (При решении реализовать процедуры ввода и вывода массивов:

Vvod2m(A,N,M,Name), Vivod2m(A,N,M,Name); а также функцию MinimumMod(A,N,M):<min матрицы> – поиск минимального по модулю элемента; процедуру ZamenaOtr(A,N,M,Chislo) – увеличение элементов матрицы на значение Chislo)

14. Ввести двумерные массивы $X1_{N1 \times N1}$, $X2_{N2 \times N2}$, $X3_{N3 \times N3}$. В каждом массиве найти среднее арифметическое положительных элементов. Преобразовать каждый из массивов по правилу: деление элементов 1-го столбца на сумму положительных элементов 1-ой строки, элементы 2-го столбца на сумму положительных элементов 2-ой строки и т.д. (При решении реализовать процедуры ввода и вывода массивов: *Vvod2m(A,N,Name), Vivod2m(A,N,Name); а также процедуру Modification(A,N) – деление столбцов на суммы строк; и функцию SrAPol(A,N):real – поиск среднего арифметического положительных элементов (равно нулю, если положительных нет)*)

15. Ввести двумерные массивы $X1_{N \times N}$, $X2_{N \times N}$. Получить массивы $X3_{N \times N}$, $X4_{N \times N}$, элементы которых вычисляются по формулам $X3[i, j] := X1[i, j] + X2[i, j]$, $X4[i, j] := X2[i, j] + X3[i, j]$. Вывести все вышеупомянутые массивы вместе со значением соответствующего максимума среди элементов главной диагонали. (При решении реализовать процедуры ввода и вывода массивов: *Vvod2m(A,N,Name), Vivod2m(A,N,Name); а также процедуру Summ(A,Ax,Ay,N) – суммирование матриц Ax, Ay с сохранением результата в A; и функцию MaxDiag(A,N):<min массива> – максимум среди элементов главной диагонали)*

16. Ввести двумерные массивы $X1_{N \times M}$, $X2_{N \times M}$, $X3_{N \times M}$. Получить массив $X4_{N \times M}$ по формуле $X4[i, j] := \max(X1[i, j], X2[i, j], X3[i, j])$ и $X5_{N \times M}$ по формуле $X5[i, j] := \max(X1[i, j], X2[i, j], X4[i, j])$. Все вышеупомянутые массивы вывести. (При решении реализовать процедуры

ввода и вывода массивов: $Vvod2m(A,N,M,Name)$, $Vivod2m(A,N,M,Name)$; а также процедуру $Sozd(A,Ax,Ay,Az,N,M)$ – создание матрицы A , на основе Ax , Ay , Az ; и функцию $Max(p,q):<min массива>$ – максимум среди пары элементов)

17. Ввести двумерные массивы $X1_{N1 \times M1}$, $X2_{N2 \times M2}$, $X3_{N3 \times M3}$. Для этих матриц найти $MinOfMin$ – минимум среди минимальных элементов и $MinOfMax$ – минимум среди максимальных элементов, которым заменить нулевые элементы в каждом массиве. Вывести массивы до и после замены. (При решении реализовать процедуры ввода и вывода массивов: $Vvod2m(A,N,M,Name)$, $Vivod2m(A,N,M,Name)$; а также функции $Min2mass(A,N,M):<min массива>$ и $Max2mass(A,N,M):<min массива>$ – поиск минимума и максимума в матрицах; и функцию $Min(p,q):<min массива>$ – минимум среди пары элементов; $ZamenaZero(A,N,M,Chislo)$ – замена нулевых элементов значением $Chislo$)

18. Ввести квадратные двумерные массивы $X1_{N1 \times N1}$, $X2_{N2 \times N2}$ и $X3_{N3 \times N3}$. Вывести эти матрицы. Найти $S = S1 \cdot S2 \cdot S3$, где $S1, S2, S3$ – произведение диагональных элементов (главной и побочной) и $m = m1 + m2 + m3$, где $m1, m2, m3$ – максимумы вышеупомянутых матриц. Значением S заменить нулевые элементы в каждой из матриц. (При решении реализовать процедуры ввода и вывода массивов: $Vvod2m(A,N,Name)$, $Vivod2m(A,N,Name)$; а также функции $SumDiag(A,N):<min массива>$ – сумма диагональных элементов и $Maximum(A,N):<min массива>$ – поиск максимума в матрице; а также процедуру замены нулевых элементов значением $Chislo$: $ZamenaZero(A,N,M,Chislo)$)

19. Ввести квадратные двумерные массивы $X1_{N1 \times N1}$ и $X2_{N2 \times N2}$. Вывести эти матрицы. В матрицах вычислить и отпечатать средние арифметические элементов расположенных под главной диагональю,

которыми заменить элементы побочной диагонали противоположной матрицы. (При решении реализовать процедуры ввода и вывода массивов: **Vvod2m(A,N,Name)**, **Vivod2m(A,N,Name)**; а также функция **SrArDiag(A,N):real** – среднее арифметическое поддиагональных элементов; и **ZamenaPoboch(A,N,Chislo)** – замена побочных элементов на значение **Chislo**)

20. Для заданных четырёх матриц $X1_{N1 \times N1}$, $X2_{N2 \times N2}$, $X3_{N3 \times N3}$ и $X4_{N4 \times N4}$ найти и напечатать длины векторов $Ya_3 = \{s1, s2, s3\}$ и $Yb_3 = \{s2, s3, s4\}$ где $s1, s2, s3, s4$ – суммы положительных элементов соответствующих матриц. Следует напомнить, что длина вектора вычисляется как квадратный корень из суммы квадратов координат. (При решении реализовать процедуры ввода и вывода массивов: **Vvod2m(A,N,Name)**, **Vivod2m(A,N,Name)**; а также функции **SummPol(A,N):<min массива>** – сумма положительных элементов; и **Dlina(p,q,r):real** – длина вектора с координатами **p,q,r**)

21. Ввести три матрицы $X1_{N1 \times N1}$, $X2_{N2 \times N2}$, $X3_{N3 \times N3}$. Вычислить и отпечатать значение функции $z = b_1 y^2 + b_2 y + b_3$, если y вводится с клавиатуры, b_1, b_2, b_3 – следы матриц $X1$, $X2$, $X3$ (следом матрицы называется сумма элементов главной диагонали). Значением z заменить нулевые элементы выше главной диагонали в каждой из матриц. Матрицы вывести до и после преобразования. (При решении реализовать процедуры ввода и вывода массивов: **Vvod2m(A,N,Name)**, **Vivod2m(A,N,Name)**; а также функцию **Sled(A,N):<min массива>** – след; и процедуру **Zamena(A,N,Chislo)** – замена нулевых элементов выше главной диагонали на **Chislo**)

22. Ввести три массива $X1_{N1}$, $X2_{N2}$, $X3_{N3}$. Найти и вывести на экран r – минимальное из трёх чисел: $r1, r2, r3$ (количества положительных чисел для каждого из массивов). Значением r заменить

нулевые элементы в каждом из массивов. Массивы вывести до и после преобразования. (При решении реализовать процедуры ввода и вывода массивов: **VvodIm(A,N,Name)**, **VivodIm(A,N,Name)**; а также функцию **KolPol(A,N):byte** – количество положительных; и процедуру **Zamena(A,N,Chislo)** – замена нулевых элементов значением **Chislo**)

23. Ввести одномерные массивы целых чисел $X1_{N1}$, $X2_{N2}$, $X3_{N3}$. Определить в каждом массиве среднее арифметическое положительных элементов кратных 5 или 3, и записать его целую часть (встроенная функция целой части: **trunc(x)**) на место 1-го, последнего и предпоследнего элемента соответствующего массива (при условии существования среднего арифметического). Вывести массивы до преобразования и после (при невозможности вычислить среднее арифметическое, вывести соответствующее сообщение). (При решении реализовать процедуры ввода и вывода массивов: **VvodIm(A,N,Name)**, **VivodIm(A,N,Name)**; а также функцию **SrAkr53(A,N):real** – среднее арифметическое положительных кратных пяти или трём (функция равна «-1», если оно не существует); и процедуру **Zamena(A,N,Chislo)** – замена первого, последнего и предпоследнего значением **Chislo**)

24. Ввести одномерные массивы $X1_{N1}$, $X2_{N2}$, $X3_{N3}$. Найти и отпечатать максимальные по модулю отрицательные элементы в каждом из массивов. Найденными значениями заменить положительные элементы соответствующих массивов. (При решении реализовать процедуры ввода и вывода массивов: **VvodIm(A,N,Name)**, **VivodIm(A,N,Name)**; а также функцию **MaxModOtr(A,N):<min массива>** – максимальное по модулю отрицательное (функция равна нулю при отсутствии отрицательных); и процедуру **Zamena(A,N,Chislo)** – замена положительных элементов значением **Chislo**)

25. Ввести квадратные матрицы $X1_{N1 \times N1}$, $X2_{N2 \times N2}$. Обменять главную и побочные диагонали у той матрицы, у которой значение суммы её элементов окажется больше. Матрицы вывести до и после преобразования. (При решении реализовать процедуры ввода и вывода массивов: $Vvod2m(A,N,Name)$, $Vivod2m(A,N,Name)$; а также функцию $Summa(A,N):<min массива>$ – сумма элементов; и процедуру $Obmen(A,N)$ – обмен диагоналей в матрице)

26. Ввести квадратные матрицы $X1_{N \times M}$, $X2_{N \times M}$. Заменить первой строкой одной матрицы первую строку другой матрицы. Матрица, строка которой будет дублирована в другой матрице, определяется минимальным значением (у какой матрицы минимальный элемент меньше – та и является донором, т.е. источником). Матрицы вывести до и после преобразования вместе со значением их минимумов. (При решении реализовать процедуры ввода и вывода массивов: $Vvod2m(A,N,M,Name)$, $Vivod2m(A,N,M,Name)$; а также функцию $Minimum(A,N,M):<min массива>$ – минимум матрицы; и процедуру $Zamena(A1,A2,M,Str)$ – замена строки номер Str в матрице $A1$ строкой Str в матрице $A2$)

27. Ввести квадратные матрицы $X1_{N1 \times N1}$, $X2_{N2 \times N2}$. Матрицу у которой над побочной диагональю окажется больше чётных элементов транспонировать. Матрицы вывести до и после преобразования. (При решении реализовать процедуры ввода и вывода массивов: $Vvod2m(A,N,Name)$, $Vivod2m(A,N,Name)$; а также функцию $ChetNadPob(A,N):byte$ – количество чётных над побочной диагональю; и процедуру $Transp(A,N)$ – транспонирование матрицы)

28. Ввести матрицы $X1_{N1 \times M1}$, $X2_{N2 \times M2}$. Матрицу у которой сумма элементов последнего столбца окажется меньше преобразовать. Преобразование заключается в замене отрицательных элементов каждой строки суммой положительных элементов этой строки. Матрицы вывести

до и после преобразования. (При решении реализовать процедуры ввода и вывода массивов: $Vvod2m(A,N,M,Name)$, $Vivod2m(A,N,M,Name)$; а также функцию $SumPoslStlb(A,N):<mun\ матрицы>$ – сумма элементов последнего столбца; и процедуру $Modif(A,N,M)$ – замена элементов в строках)

29. Ввести матрицы $X1_{N1 \times M1}$, $X2_{N2 \times M2}$. Сформировать из матриц пару одномерных массивов $Y1_{Ny1}$ и $Y2_{Ny2}$, в которые поместить положительные элементы соответствующих матриц. В одномерных массивах найти количество четных элементов и вывести их вместе с самими массивами. (При решении реализовать процедуры ввода и вывода массивов: $Vvod2m(A,N,M,Name)$, $Vivod2m(A,N,M,Name)$, $Vivod1m(A,N,Name)$; процедуру $FormPol(A,Na,Ma,B,Nb)$ – формирование из положительных элементов двумерного массива A одномерного массива B , а также функцию $KolChet(A,N):byte$ – количество чётных элементов в одномерном массиве)

30. Ввести матрицы $X1_{N1 \times M1}$, $X2_{N2 \times M2}$. Сформировать из матриц пару одномерных массивов $Y1_{Ny1}$ и $Y2_{Ny2}$, в которые поместить элементы, превышающие среднее арифметическое соответствующих матриц. (При решении реализовать процедуры ввода и вывода массивов: $Vvod2m(A,N,M,Name)$, $Vivod2m(A,N,M,Name)$, $Vivod1m(A,N,Name)$; процедуру $Form(A,Na,Ma,B,Nb,Chislo)$ – формирование из элементов двумерного массива A , превышающих значение $Chislo$, одномерного массива B ; функцию $SrA(A,N,M):real$ – среднее арифметическое элементов двумерного массива)

Обработка компонентных файлов и массивов с использованием подпрограмм

1. Ввести 2 квадратные матрицы $A1_{N1 \times N1}$ и $A2_{N2 \times N2}$, вывести их. Далее переписать в файл F построчно ту из матриц, у которой модуль суммы отрицательных элементов под побочной диагональю окажется больше. В конец файла дописать полученные суммы. (в случае равенства предпочтение отдать матрице $A1$). Полученный файл вывести. (*прим.: при решении использовать подпрограммы*)

2. Ввести 3 файла Fx , Fy и Fz одинаковой длины M . Вывести их друг под другом. Сформировать из этих файлов двумерный массив $A_{3 \times M}$ в котором верхняя строка – последовательно записанные элементы файла Fx , средняя – элементы Fy , а нижняя, соответственно, элементы Fz . Матрицу отсортировать по первой строке и вывести ее как до, так и после преобразования. (*прим.: при решении использовать подпрограммы*)

3. Ввести 3 файла $F1$, $F2$, $F3$, вывести их. Упорядочить их по убыванию методом пузырька. Переписать в одномерный массив V_k элементы того файла, чья сортировка потребует наибольшее число перестановок (в случае равенства переписать первый среди равных). Массив V вывести. (*прим.: при решении использовать подпрограммы*)

4. Ввести 2 файла Fx и Fy одинаковой длины $M \geq 2$. Вывести их друг под другом. Далее сократить размер каждого из файлов на 1 заменяя в нем компоненты разностью соседних ячеек (от правой отнять левую). Файлы вывести. После этого сформировать одномерный массив V_{M-1} элементами которого являются частные от деления соответствующих элементов Fy на Fx . Полученный массив вывести. (*прим.: при решении использовать подпрограммы*)

5. Ввести файл $F1$, из положительных элементов которого сформировать другой файл $F2$ читая $F1$ от конца к началу. Из этих файлов сформировать, соответственно, квадратные матрицы $A1_{N1 \times N1}$ и $A2_{N2 \times N2}$ максимально возможного размера записывая в них элементы построчно. В

полученных матрицах поменять местами первую строку и главную диагональ. В процессе решения задачи выводить файлы и матрицы как до, так и после преобразований. (*прим.: при решении использовать подпрограммы*)

6. Ввести 2 числа $N \leq 7$ и $M \leq 7$. Сформировать 2 квадратные матрицы $A_{N \times N}$ и $B_{M \times M}$ по правилу $a_{i,j} = (i + j - 1)!$ (факториал числа), вывести их. Переписать в файл F четные строки матрицы A и нечетные столбцы матрицы B , файл вывести. После этого каждую третью компоненту файла заменить значением суммы $N + M$ и вновь его вывести. (*прим.: при решении использовать подпрограммы*)

7. Ввести файлы $Fx1$ и $Fx2$ вещественных чисел, вывести их. Дважды протабулировать функцию $y = 2x^2 + 4$ сначала читая значения x из файла $Fx1$, а затем из файла $Fx2$ (для корректной табуляции файл нужно предварительно отсортировать по возрастанию). Во время табуляции сформировать двумерные массивы $A1_{2 \times N1}$ и $A2_{2 \times N2}$ (на экран вывести именно эти 2 матрицы минуя вывод при вычислении $y(x)$). В данных матрицах ($A1$ и $A2$) первые строки – копии соответствующих файлов ($Fx1$ и $Fx2$), а вторые – значения функций в соответствующих точках. (*прим.: при решении использовать подпрограммы*)

8. Ввести 2 целочисленных файла $F1$ и $F2$ в которых первая компонента – количество строк, а вторая – количество столбцов в хранящихся в этих файлах матрицах. Третья компонента – сумма всех элементов в этих матрицах. Восстановить исходные матрицы $A1$, $A2$ и вывести их. В случае невозможности проведения операции восстановления (элементов в файле меньше заявленного или размерность меньше 1) вывести об этом сообщение. Кроме того, вывести истинную и заявленные суммы элементов в матрицах. (*прим.: при решении использовать подпрограммы*)

9. Ввести 2 равноразмерные матрицы $A1_{N \times M}$ и $A2_{N \times M}$. Вывести их. Из элементов этих матриц сформировать файлы $F1$ и $F2$ записывая построчно в первый файл $\max(a1_{i,j}, a2_{i,j})$, а во второй, соответственно, $\min(a1_{i,j}, a2_{i,j})$, нулевые элементы пропускать. Полученные файлы вывести. Далее их отсортировать по убыванию и снова вывести. (*прим.: при решении использовать подпрограммы*)

10. Ввести файлы $F1$ и $F2$, вывести их. Поменять в них местами перекрестно максимальные и минимальные компоненты (т.е. \max из первого файла идет на место \max из второго и наоборот, для \min аналогично). Далее элементы стоящие после первого встреченного в файле экстремального значения (максимума или минимума) переписать в одномерные массивы $V1$ и $V2$ соответственно. Массивы вывести. (*прим.: при решении использовать подпрограммы*)

11. Ввести файлы $F1$ и $F2$, вывести их. Посчитать в файлах среднее арифметическое. Все элементы, значения которых отличаются от среднего арифметического соответствующего файла более чем в полтора раза по модулю (как в большую так и в меньшую сторону), переписать в одномерные массивы $V1$ и $V2$, которые отсортировать по убыванию. Полученные массивы и файлы вывести. (*прим.: при решении использовать подпрограммы*)

12. Ввести 2 действительных файла $F1$ и $F2$ одинаковой длины N . Вывести их. Далее сформировать массив V_N по формуле $b_i = (K1^2 + K2^2) \cdot (L1 \cdot f2_i + L2 \cdot f1_i)$, где $K1, K2$ – соответственно максимальные по модулю компоненты файлов $F1$ и $F2$. $L1, L2$ – количество неотрицательных компонент в этих файлах, соответственно, меньших чем $|K1|/2$ и $|K2|/2$. Массив V вывести. (*прим.: при решении использовать подпрограммы*)

13. Ввести матрицу $A_{N \times M}$, вывести ее. Далее из четных строк матрицы сформировать файл $F1$, а из нечетных столбцов – файл $F2$, вывести их. В файле, где собственный наименьший элемент окажется наибольшим поменять местами максимум и минимум (в случае равенства минимумов в файлах $F1$ и $F2$ замену провести в обоих). (прим.: при решении использовать подпрограммы)

14. Ввести файлы $F1$ и $F2$ одинаковой длины N , вывести их. Вставить все элементы файла $F1$ в середину $F2$ между его элементами, а элементы $F2$, соответственно, в середину между элементами $F1$ (можно воспользоваться вспомогательным массивом). Показать полученные файлы. Каждый из файлов проверить на отсортированность по убыванию составляющих его элементов (вывести сообщение «отсортирован» или «не отсортирован»). (прим.: при решении использовать подпрограммы)

15. Ввести три одномерных массива $B1$, $B2$ и $B3$. Вывести эти массивы. Переписать элементы из массивов в файл F . Сначала переписать массив с наибольшим средним арифметическим, далее – с наименьшим, а потом оставшийся. Файл вывести найдя в нем значение и позицию максимального по модулю элемента. (прим.: при решении использовать подпрограммы)

16. Ввести файлы $F1$ и $F2$, найти в них количество элементов $K1$ и $K2$, стоящих после максимального четного (считать что в файлах есть хоть один четный элемент). Файлы вывести. Из этих двух файлов сформировать построчно квадратную матрицу $A_{N \times N}$ максимально возможного размера читая в нее сначала $F1$, а затем $F2$. Первый и последний элементы матрицы домножить на сумму $K1+K2$. Матрицу вывести. (прим.: при решении использовать подпрограммы)

17. Ввести 2 квадратные матрицы $A1_{N1 \times N1}$ и $A2_{N2 \times N2}$, вывести их. В каждой из них упорядочить по возрастанию элементы главной диагонали.

Матрицы вывести. Далее по столбцам переписать в файл F ту из матриц, у которой меньше положительных элементов в нижнем треугольнике (в случае равенства переписать обе). Файл вывести. (*прим.: при решении использовать подпрограммы*)

18. Ввести матрицы $A1_{N1 \times M1}$ и $A2_{N2 \times M2}$. Вывести их. Сформировать файлы $F1$ и $F2$ записывая в них периметр матрицы (начиная с первого элемента и далее против часовой стрелки). Полученные файлы вывести. В файле, где наименьший элемент встречается позже, все компоненты после наименьшего заменить нулями (в случае равенства позиций минимумов в файлах $F1$ и $F2$ замену провести в $F1$). Файлы вывести. (*прим.: при решении использовать подпрограммы*)

19. Ввести 2 матрицы $A1_{N1 \times M1}$ и $A2_{N2 \times M2}$, вывести их. Далее из суммы квадратов столбцов каждой из матриц сформировать файлы $F1$ и $F2$. Из каждого элемента файлов извлечь квадратный корень. Файлы вывести. Элементы файла $F2$ в обратном порядке дописать к файлу $F1$. $F1$ снова отпечатать. (*прим.: при решении использовать подпрограммы*)

20. Ввести 2 матрицы $A1_{N1 \times M1}$ и $A2_{N2 \times M2}$, вывести их. Максимальные элементы из каждого столбца поместить, соответственно, в файлы $F1$ и $F2$. Вывести их вместе с сообщением о том в каком из файлов раньше встречается минимальный среди положительных элемент. (*прим.: при решении использовать подпрограммы*)

21. Ввести матрицы $A1_{N1 \times M1}$ и $A2_{N2 \times M2}$, вывести их. В матрицах найти предмаксимальные элементы (по значению) и разделить все элементы этих матриц на соответствующие полученные значения. Далее переписать в файл F элементы из матриц по столбцам. Файл вывести, поменять в нем максимальный и первый элементы местами и снова вывести. (*прим.: при решении использовать подпрограммы*)

22. Ввести матрицы $A1_{N1 \times M}$ и $A2_{N2 \times M}$, вывести их. Далее из этих матриц сформировать файл F записывая в него сначала элементы первой строки матрицы $A1$, далее первую строку $A2$, потом вторую строку из $A1$, вторую строку из $A2$ и т.д. до M -ой строки. Файл вывести. В полученном файле провести усреднение элементов по следующему правилу: $f_i = (f_{i-1} + f_i) / 2$, первый элемент вычисляется как $f_1 = (f_1 + f_2) / 2$. Файл вывести. (прим.: при решении использовать подпрограммы)

23. Ввести 3 квадратные матрицы $A1_{N1 \times N1}$, $A_{N2 \times N2}$ и $A3_{N3 \times N3}$. Вывести их. Найти в них максимальные по модулю элементы на побочных диагоналях $M1$, $M2$ и $M3$. Далее переписать в файл Fy отрицательные значения функции $y(x) = M1 \cdot ((M3 + M2) \cos(x))$ если $x \in [-\pi; 2\pi]$, $\Delta x = \pi / 4$, а в файл Fx , соответствующие значения x . Полученные файлы вывести. (прим.: при решении использовать подпрограммы)

24. Ввести 3 файла $F1$, $F2$, $F3$, вывести их. Далее из них составить матрицу $A_{3 \times M}$, где M – это размерность длиннейшего файла. В матрице A первая строка – файл $F1$, вторая – $F2$, третья $F3$ (если элементов в файле не хватает, то конец строки заполнить нулями). Матрицу вывести. Далее в матрице поменять местами первый и последний столбцы. (прим.: при решении использовать подпрограммы)

25. Сформировать файлы $F1$ и $F2$ длины $N1$ и $N2$ по правилу последовательности Фибоначчи: $a_i = a_{i-1} + a_{i-2}$, причем $a_1 = a_2 = 1$, $i = 1, 2, 3, \dots, N$. Полученные файлы вывести. Далее составить из файлов 2 квадратные матрицы $A1$ и $A2$ максимально возможного размера. Матрицы вывести. Транспонировать эти матрицы и снова вывести. (прим.: при решении использовать подпрограммы)

26. Ввести матрицу $A_{N \times M}$, вывести ее. Далее из четных строк сформировать файл $F1$, а из нечетных файл $F2$. Файлы вывести. Далее отсортировать каждый из файлов. Получить новый файл F записывая в

него сначала $F2$ в прямом, а потом $F1$ в обратном порядке пропуская нулевые элементы. Файлы вывести. (прим.: при решении использовать подпрограммы)

27. Ввести матрицы $A1_{N2 \times M2}$ и $A2_{N2 \times M2}$, вывести их. Отсортировать в них четные столбцы по возрастанию, а нечетные по убыванию. Матрицы вывести. Переписать построчно в файл F , сначала элементы матрицы $A1$ превышающие величину $S1 = \max A1/2$, а потом $A2$, превышающие величину $S2 = \max A2/2$. Значения $S1$, $S2$ и полученный файл вывести. (прим.: при решении использовать подпрограммы)

28. Ввести 2 матрицы $A1_{N1 \times M1}$ и $A2_{N2 \times M2}$ ($M1, M2 \geq 2$), вывести их. В каждой из матриц найти все суммы четных элементов в соседних столбцах. Полученные суммы сохранить, соответственно, в файлах $F1$ и $F2$. Файлы вывести, указав в каком из файлов больше положительных компонент. (прим.: при решении использовать подпрограммы)

29. Ввести матрицу $A_{N \times M}$, вывести ее. Найти в ней координаты максимума. Далее в матрице определить в какой из четвертей относительно максимума больше всего элементов (слева и выше; справа и выше; слева и ниже; или справа и ниже). Из этих элементов сформировать матрицу $A1_{N1 \times M1}$, а из оставшихся элементов файл $F1$, в конец файла приписать сумму его элементов, далее файл вывести. Для матрицы $A1_{N1 \times M1}$ проделать аналогичные действия, сформировав при этом матрицу $A2_{N2 \times M2}$ и файл $F2$. (прим.: при решении использовать подпрограммы)

30. Ввести матрицу $A_{N \times M}$, вывести ее. Найти в ней координаты минимума. Далее из элементов, расположенных правее и ниже найденного минимума сформировать матрицу $A1_{N1 \times M1}$, а из оставшихся элементов файл $F1$. Для матрицы $A1_{N1 \times M1}$ проделать аналогичные действия, сформировав при этом матрицу $A2_{N2 \times M2}$ и файл $F2$. Файлы вывести. Далее вывести еще раз тот файл, в котором больше положительных элементов совместно с найденным числом (*прим.: при решении использовать подпрограммы*)

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Авдеюк О.А. Сложные структурированные типы данных в языке ТР: массивы : учеб. пособие / О.А. Авдеюк, О.В. Гостевская, С.Р. Калмыкова, Е.С. Павлова / ВолгГТУ. – Волгоград, 2006. – 64 с.
2. Двумерные массивы : метод.указания по выполнению лабораторных и практических работ / О.А. Авдеюк, Л.Г. Акулов, Р.С. Богатырев, О. В. Гостевская, В.Ю. Наумов, Е.С. Павлова / ВолгГТУ. – Волгоград, 2008. – 24 с.
3. Епанешников А. М. Программирование в среде TurboPascal7.0 / А. М. Епанешников, В. А. Епанешников - М. : ДИАЛОГ-МИФИ, 2004.
4. Наумов, В.Ю. Сборник заданий для лабораторных работ по информатике : учеб.пособие / В.Ю. Наумов, О.В. Гостевская, И.Г. Лемешкина, Е.С. Павлова, Р.С. Богатырев, Л.Г. Акулов, Р.В. Литовкин, О.А. Авдеюк. – Волгоград : ИУНЛ ВолгГТУ, 2010. – 79 с.
5. Фаронов В. В. Турбо Паскаль 7.0. Практика программирования : учеб.пособие / В. В. Фаронов - М. : ОМД Групп, 2003.

Учебное издание

Оксана Алексеевна **Авдеюк**
Леонид Геннадьевич **Акулов**
Ольга Владиславовна **Гостевская**
Ирина Геннадьевна **Лемешкина**
Елена Станиславна **Павлова**
Вадим Юрьевич **Наумов**

ИНФОРМАТИКА И ПРОГРАММИРОВАНИЕ: РУКОВОДСТВО К ЛАБОРАТОРНЫМ И ПРАКТИЧЕСКИМ ЗАНЯТИЯМ. ЧАСТЬ 2.

Учебное пособие

Редактор Е. А. Пичугина

Темплан 2014 г. Позиция № ____.
Лицензия ИД № 04790 от 18.05.2007 года
Подписано в печать _____. Формат 60 x 84 1/16.
Бумага газетная. Печать офсетная. Усл. печ. л. _____. Уч.-изд. л. _____.
Тираж экз. Заказ _____.

Волгоградский государственный технический университет.
400005 Волгоград, просп. Ленина, 28.

РПК “Политехник”
Волгоградского государственного технического университета.
400005 Волгоград, ул. Советская, 35.