ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ

ОРЛОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Курсовая работа

Основы программирования и алгоритмизации

Вариант – 10

Работу выполнил студент

Группа Факультет ФНТиАП

Специальность Автоматизация технологических процессов и производств

Курсовая работа защищена с оценкой

Орел 2009

Задание на курсовую работу

Студент

Группа Факультет ФНТиАП

Специальность Автоматизация технологических процессов и производств

Вариант – 10

Задания

1. В ЭВМ поступают результаты соревнований по плаванию для трех спортсменов. Составьте программу, которая выбирает лучшей результат и выводит его на экран с сообщением, что это лучшей результат.

2. Дано натуральное число n. Составить программу, определяющую, есть ли среди чисел n, n+1,...,2n близнецы, т.е. простые числа, разность между которыми равна 2. (Использовать процедуру распознавания простых чисел).

3. Найдите сумму элементов массива ниже главной диагонали, произведение не равных нулю элементов выше главной диагонали и количество элементов в главной диагонали, попадающих в интервал [-1;1].

4. Опишите множества М1 (1, 2,3) и M2 (1, 4, 2, 5). Получите результирующие множество M3=M1\*M2. Определите, имеются ли в M3 элементы 1 и 2.

5. Создайте файл, компоненты которого являются целыми числами. Напишите программу, переписывающую компоненты файла в обратном порядке.(Новый файл не заводить).

Руководитель

Задание принял к исполнению «\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2009 г.

Подпись студента

Содержание

Задание № 1

Задание № 2

Задание № 3

Задание № 4

Задание № 5

Приложение 1

Приложение 2

Приложение 3

Приложение 4

Приложение 5

Задание №1

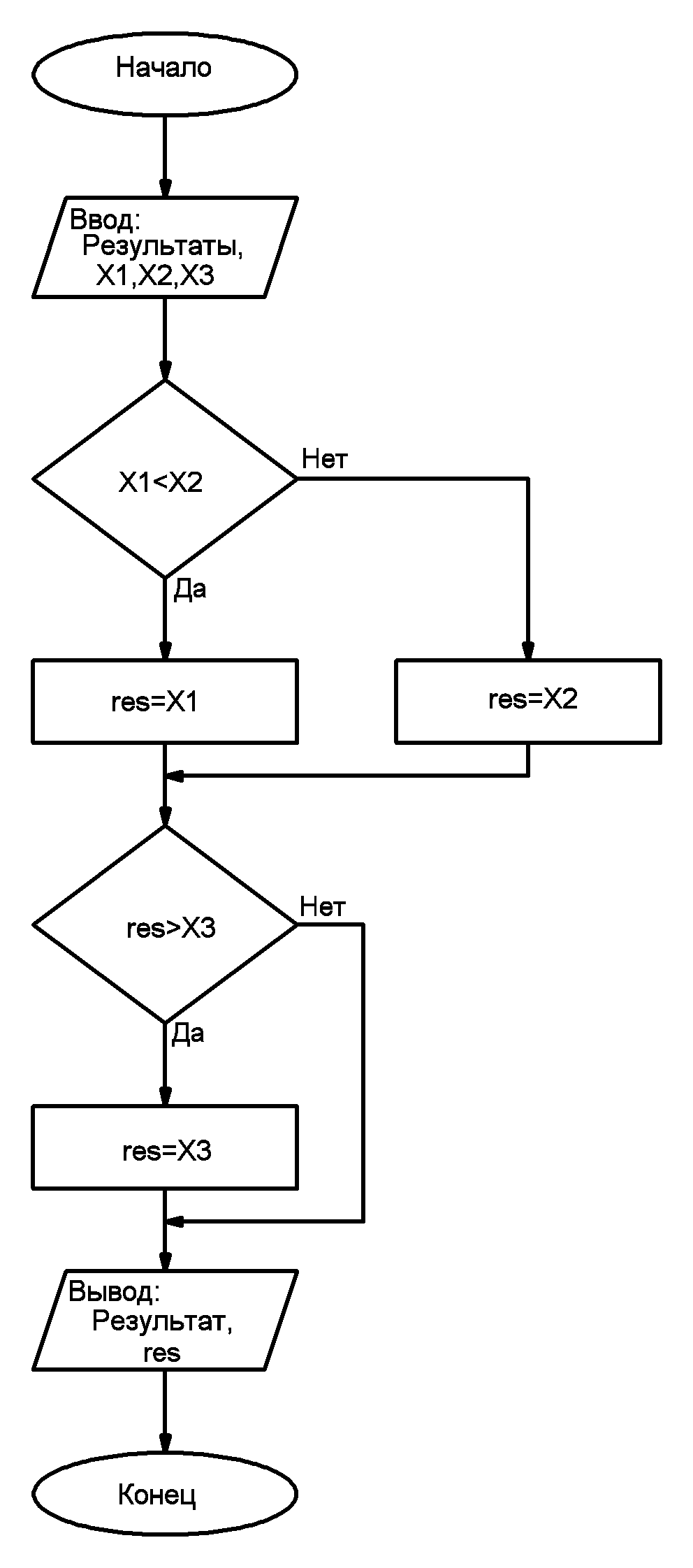
Название программы введено после program, в данном задание это – Zadanie1. Далее с помощью var выделены ячейки памяти под переменные X1, X2, X3, res вещественного типа данных. После begin написан ход работы программы из последовательности операторов.

С помощью операторов writeln и readln описывается, что нужно ввести и записывается введенный результат в память, соответственно. Таким образом, вводим результаты спортсменов X1, X2, X3.

Далее с помощью условных операторов if определяем наименьший результат, который присваивается переменной res: if X1<X2 then res:=X1 else res:=X2; if res>X3 then res:=X3;. С помощью оператора writeln на экран выдается сообщение с результатом победителя заплыва.

Закрытие программы – end.

Алгоритм к Заданию №1:



Задание №2

Название программы введено после program, в данном задание это – Zadanie2. Далее с помощью var выделены ячейки памяти под переменные j, B, N типа данных word и массива а логического типа данных. После begin написан ход работы программы из последовательности операторов.

С помощью процедуры Pch определяем все простые числа, используемые в программе. Далее оператором writeln выводим сообщение о том, что ниже будут выдаваться числа близнецы.

Создаем цикл: for j:=N to 2\*N do, где будет проверяться условие, что разность между числами равна 2. Если условие верно, то выдаются на экран сравнивающиеся числа, которые и будут близнецы.

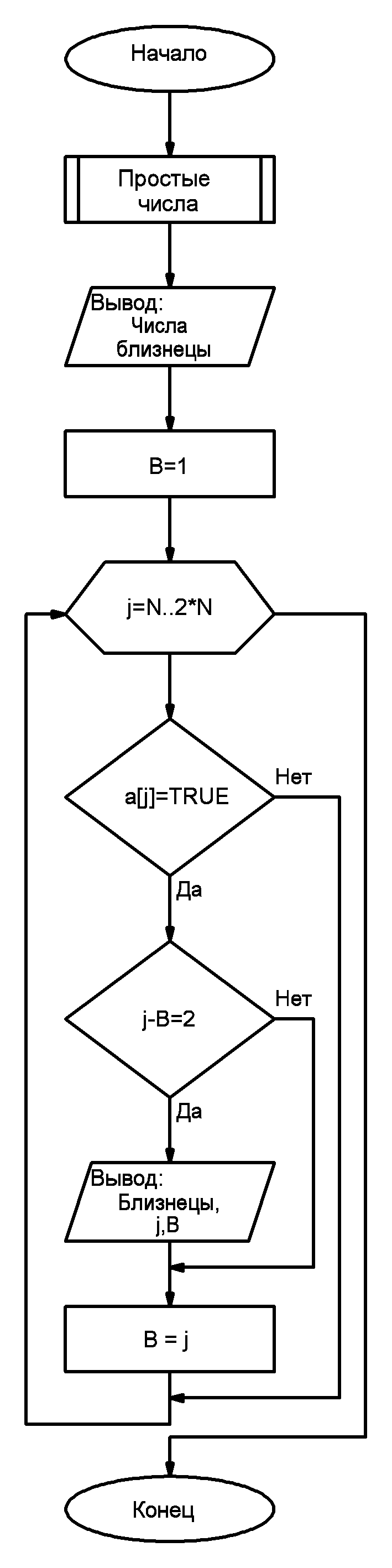
Закрытие программы – end.

Процедура Pch.

Кроме общих переменных заданных в программе в процедуре используются внутренние переменные b, k типа данных word. Задаем диапазон, на котором будем выделять простые числа. Далее создаем цикл: for j:=2 to 2\*N do, в котором всем значениям массива а присваиваем значение TRUE. С помощью нескольких циклов в цикле: while k<=b do, всем не простым числам в массиве а присваиваются значения FALSE.

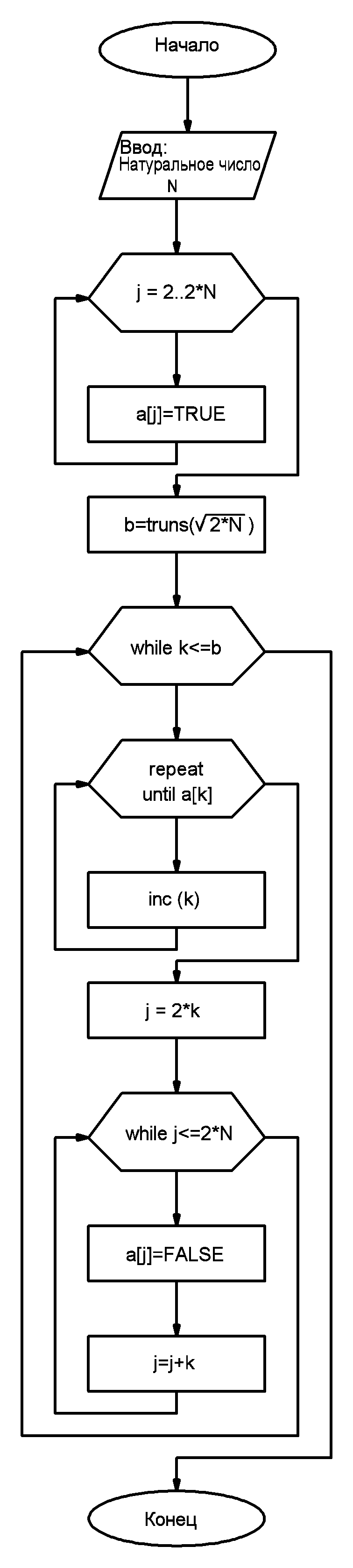
Таким образом получаем массив а, где простые числа отмечены TRUE. Закрытие процедуры – end.

Алгоритм к Заданию №2:



Процедура

Простые числа:



Задание №3

Название программы введено после program, в данном задание это – Zadanie3. Далее с помощью var выделены ячейки памяти под переменные i, j, K, n, m типа данных byte, переменные S, Pr и массив a[1..255,1..255] вещественного типа данных. После begin написан ход работы программы из последовательности операторов.

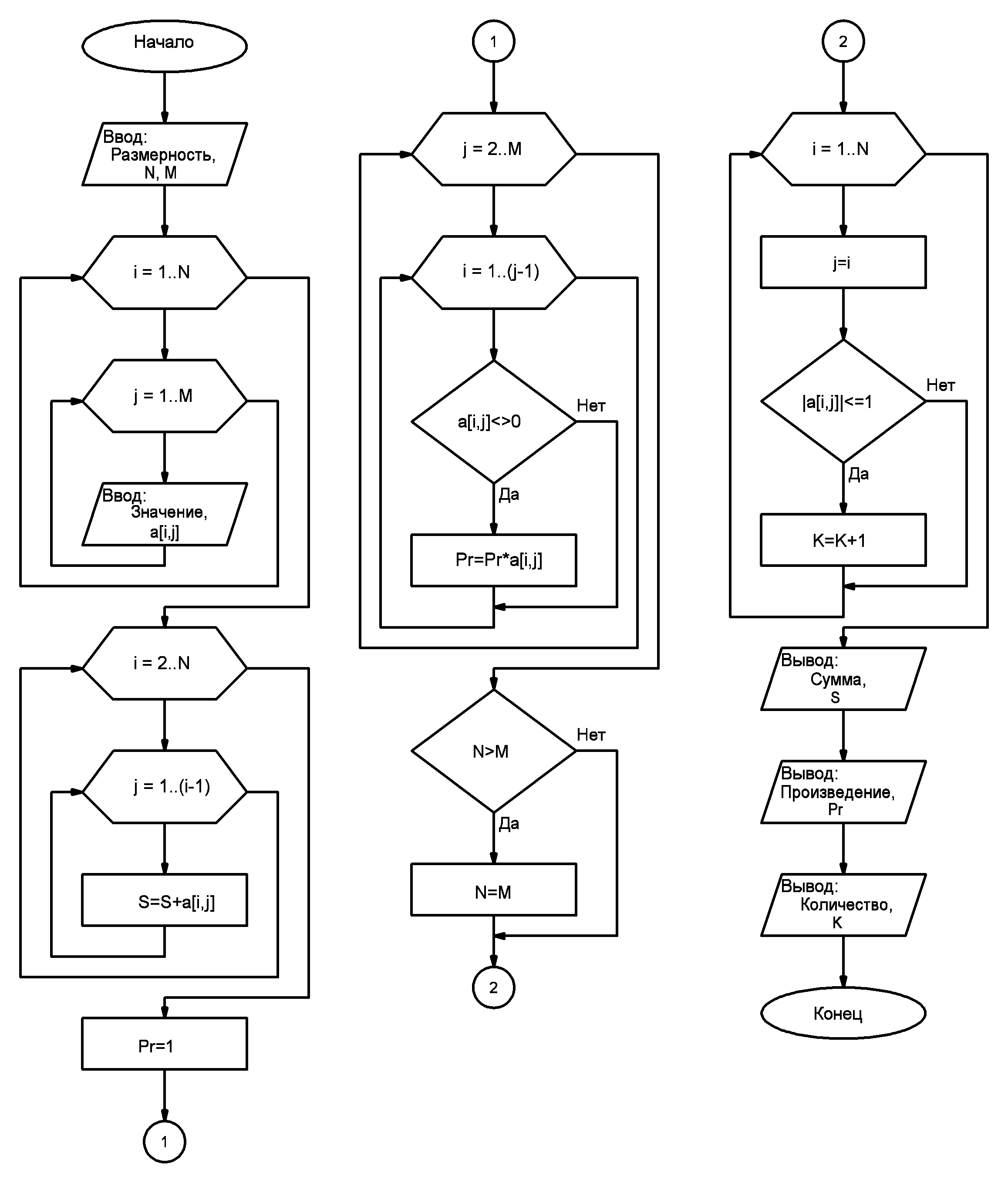
Вводим размерность матрицы NxM, далее с помощью цикла: for j:=1 to m dо в цикле: for i:=1 to n do задаем все значения массива а.

Следующий этап – создаем цикл: for j:=1 to (i-1) do в цикле: for i:=2 to n do, где будут перебираться только элементы ниже главной диагонали и находим их сумму. Далее создаем цикл: for i:=1 to (j-1) do в цикле: for j:=2 to m do, где будут перебираться только элементы выше главной диагонали. Проверяем условие не равности нулю каждого элементы и находим их произведение: if a[i,j]<>0 then Pr:=Pr\*a[i,j].

Далее в цикле: for i:=1 to n do, j присваиваем значение i, то есть перебираем только элементы главной диагонали, где проверяем принадлежность этих элементов диапазону [-1;1] и считаем это количество.

С помощью операторов writeln на экран выдаем результат. Закрытие программы – end.

Алгоритм к Заданию №3:



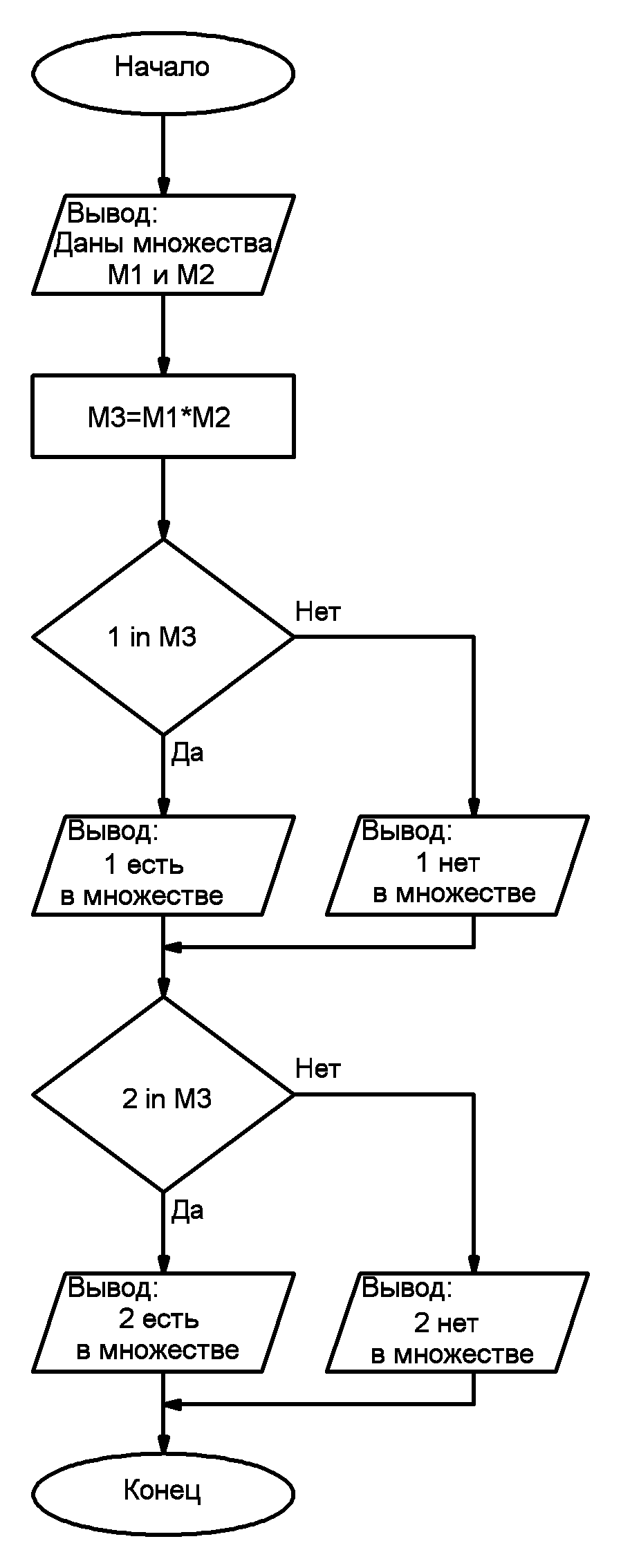
Задание №4

Название программы введено после program, в данном задание это – Zadanie4. Далее с помощью const заданы константы, множества M1=[1,2,3] и M2=[1,4,2,5], а с помощью var выделены ячейки памяти под множество M3 типа данных byte. После begin написан ход работы программы из последовательности операторов.

С помощью оператора writeln выводим сообщение о том, что даны множества М1и M2. Далее находим множество М3: М3:=M1\*M2. С помощью условных операторов if проверяем наличие в множестве М3 элементов 1 и 2, после чего выдается соответствующее сообщение.

С помощью строки readln (); выполнение программы автоматически не закроется, что позволит посмотреть результат, а закрытие осуществляется после нажатия кнопки Enter. Закрытие программы – end.

Алгоритм к Заданию №4:



Задание №5

Название программы введено после program, в данном задание это – Zadanie5. Далее с помощью var выделены ячейки памяти под переменные i целого типа данных, k, j типа данных byte, массив а и f файл с целым типом данных. После begin написан ход работы программы из последовательности операторов.

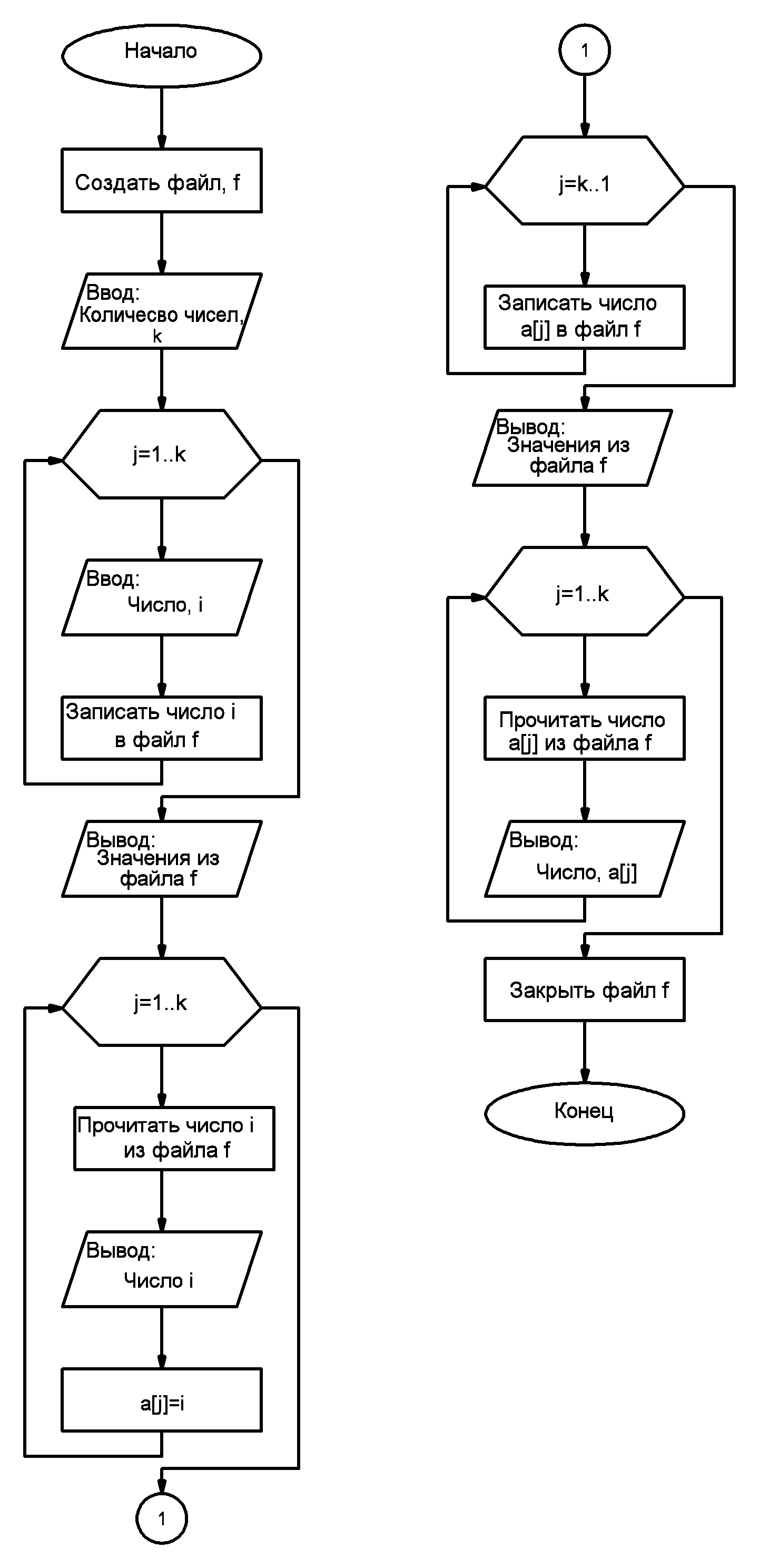
Оператором assign устанавливаем связь между программой и файлом zadanie5.txt. Далее командой rewrite (f); открываем (создаем) файл для записи. Вводим количество вводимых чисел в файл. С помощью цикла: for j:=1 to k do, вводим поочередно k целых чисел, а с помощью команды write (f,i); записываем эти числа в файл.

Далее открываем файл, только в этот раз для чтения reset (f). И прочитываем все значения i и записываем их в массив a. Далее командой rewrite (f) опять открываем файл для записи и с помощью цикла: for j:=k downto 1 do, записываем значения массива а в файл f в обратном порядке.

Далее открываем файл для чтения reset (f). И прочитываем все значения. Затем с помощью цикла: for j:=1 to k do, командой read (f,a[j]) читаем числа и выводим их на экран через пробел: write (a[j],' ').

Далее закрываем файл и программу.

Алгоритм к Заданию №5:



Приложение 1

Текст программы к Заданию №1

program Zadanie1;

var

X1,X2,X3,res:real;

begin

writeln ('Введите результат 1 спортсмена: ');

readln (X1);

writeln ('Введите результат 2 спортсмена: ');

readln (X2);

writeln ('Введите результат 3 спортсмена: ');

readln (X3);

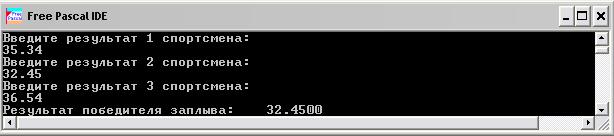
if X1<X2 then res:=X1 else res:=X2;

if res>X3 then res:=X3;

writeln ('Результат победителя заплыва: ', res:10:4);

end.

Решение:



Приложение 2

Текст программы к Заданию №2

program Zadanie2;

var

j,B,N: word;

a: array [2..1000] of boolean;

procedure Pch;

var

b,k:word;

begin

writeln ('Введите натуральное число N: ');

readln (N);

for j:=2 to 2\*N do

a[j]:=true;

b:=trunc(sqrt(2\*N));

while k<=b do

begin

repeat

inc(k);

until a[k];

j:=2\*k;

while j<=2\*N do

begin

a[j]:=false;

j:=j+k;

end;

end;

end;

begin

Pch;

writeln ('Числа близнецы на диапозоне ',N,'..',2\*N,' :');

B:=1;

for j:=N to 2\*N do

if a[j] then

begin

if j-B=2 then writeln (j,' | ',B);

B:=j;

end;

end.

Решение:



Приложение 3

Текст программы к Заданию №3

program Zadanie3;

var

i,j,K,n,m: byte;

S,Pr: real;

a: array [1..255,1..255] of real;

begin

writeln ('Введите размерность массива N x M:');

readln (n,m);

for i:=1 to n do

for j:=1 to m do

begin

writeln ('Введите a[',i,',',j,'] : ');

readln (a[i,j]);

end;

for i:=2 to n do

for j:=1 to (i-1) do

S:=S+a[i,j];

Pr:=1;

for j:=2 to m do

for i:=1 to (j-1) do

if a[i,j]<>0 then Pr:=Pr\*a[i,j];

if n>m then n:=m;

for i:=1 to n do

begin

j:=i;

if abs(a[i,j])<=1 then K:=K+1;

end;

writeln ('Сумма элементов ниже главной диагонали:',S:7:2);

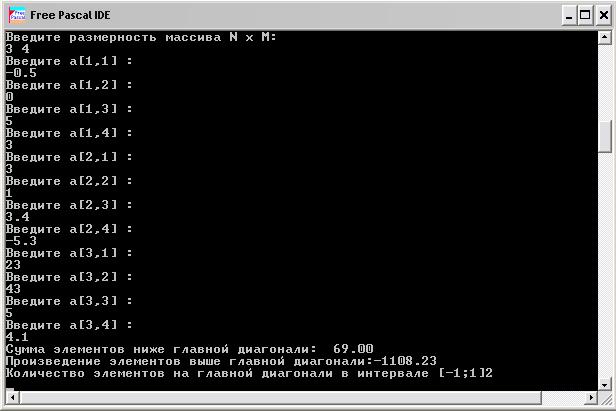
writeln ('Произведение элементов выше главной диагонали:',Pr:7:2);

writeln ('Количество элементов на главной диагонали в интервале [-1;1]',K);

readln();

end.

Решение:



Приложение 4

Текст программы к Заданию №4

program Zadanie4;

const

M1=[1,2,3];

M2=[1,4,2,5];

var

M3: set of byte;

begin

writeln ('Даны множества М1(1,2,3), M2(1,4,2,5) и М3=М1\*М2');

M3:=M1\*M2;

if 1 in M3 then writeln ('1 есть в множестве М3')

else writeln ('1 нет в множестве М3');

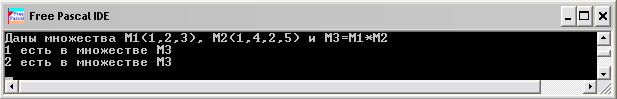
if 2 in M3 then writeln ('2 есть в множестве М3')

else writeln ('2 нет в множестве М3');

readln;

end.

Решение:



Приложение 5

Текст программы к Заданию №5

program Zadanie5;

var

f: file of integer;

i: integer;

k,j: byte;

a: array [1..255] of integer;

begin

assign (f,'zadanie5.txt');

rewrite (f);

writeln ('Введите количество вводимых чисел: ');

readln (k);

for j:=1 to k do

begin

write ('Введите ',j,' число: ');

read (i);

write (f,i);

end;

reset (f);

writeln ('Значения из созданного файла:');

for j:=1 to k do

begin

read (f,i);

write (i,' ');

a[j]:=i;

end;

rewrite (f);

for j:=k downto 1 do

write (f,a[j]);

writeln;

reset (f);

writeln ('Значения из файла:');

for j:=1 to k do

begin

read (f,a[j]);

write (a[j],' ');

end;

close (f);

writeln;

end.

Решение:

