*Федеральное государственное бюджетное образовательное учреждение* *высшего образования*

|  |  |
| --- | --- |
| Gerb-BMSTU_01 | ***«Московский государственный технический университет имени Н.Э. Баумана***  ***(национальный исследовательский университет)»***  ***(МГТУ им. Н.Э. Баумана)*** |

ФАКУЛЬТЕТ: ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА: КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

**Отчет**

**по домашнему заданию № 2**

**Название лабораторной работы: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Дисциплина:** Основы программирования

Студент гр. ИУ6-12Б  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ С.В.Астахов**

(Подпись, дата) (И.О. Фамилия)

Преподаватель  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(Подпись, дата) (И.О. Фамилия)

Москва, 2019

I вариант

Задание 1

Сортировать одномерный массив вещественных чисел по возрастанию абсолютной величины числа. Использовать метод вставок.

Текст программы:

program Project1;

{$APPTYPE CONSOLE}

uses

System.SysUtils;

type

arr = array [0 .. 50] of real;

var

a: arr;

n, i, g: byte;

target: real;

begin

writeln('Enter array size in (0;50]');

readln(n);

if ((n <= 0) or (n >= 50)) then

writeln('n not in (0;50]')

else

begin

writeln('Enter array, each element in new string');

for i := 1 to n do

readln(a[i]);

writeln;

writeln('Not sorted array: ');

for i := 1 to n do

write(a[i]:6:2, ' ');

for i := 1 to n do

begin

target := a[i];

a[0] := target;

g := i - 1;

while ((abs(target)) < (abs(a[g]))) do

begin

a[g + 1] := a[g];

g := g - 1;

end;

a[g + 1] := target;

end;

writeln;

writeln('Sorted array: ');

for i := 1 to n do

write(a[i]:6:2, ' ');

end;

readln;

end.

Тесты:

|  |  |  |
| --- | --- | --- |
| Входные данные | Ожидаемые выходные данные | Выходные данные |
| 5  10  1  -9  8  -3 | Not sorted array:  10.00 1.00 -9.00 8.00 -3.00  Sorted array:  1.00 -3.00 8.00 -9.00 10.00 | Not sorted array:  10.00 1.00 -9.00 8.00 -3.00  Sorted array:  1.00 -3.00 8.00 -9.00 10.00 |
| -3 | 'n not in (0;50]' | 'n not in (0;50]' |
| 7  1  4  3  -2  8  -6  7 | Not sorted array:  1.00 4.00 3.00 -2.00 8.00 -6.00 7.00  Sorted array:  1.00 -2.00 3.00 4.00 -6.00 7.00 8.00 | Not sorted array:  1.00 4.00 3.00 -2.00 8.00 -6.00 7.00  Sorted array:  1.00 -2.00 3.00 4.00 -6.00 7.00 8.00 |

Схема алгоритма:



Задание 2

Составить программу, которая в матрице A(n, n), (n<11) меняет местами строку, содержащую максимальный элемент, со строкой, содержащей минимальный элемент. Предполагается, что искомые элементы единственные. Вывести на экран исходную и преобразованную матрицы, минимальный и максимальный элементы, а также номера строк, в которых они расположены.

Текст программы:

program Project1;

{$APPTYPE CONSOLE}

uses

SysUtils;

type

arr = array [1 .. 11, 1 .. 11] of real;

var

a: arr;

n, i, g, imin, imax: byte;

min, max, dub: real;

begin

writeln('Enter size of matrix in range (0;11]');

readln(n);

if ((n <= 0) or (n > 11)) then

writeln('N not in (0;11]')

else

begin

writeln('Enter elements(each with enter)');

for i := 1 to n do

for g := 1 to n do

readln(a[i, g]);

min := a[1, 1];

max := a[1, 1];

imin := 1;

imax := 1;

writeln;

writeln('Inputed matrix: ');

for i := 1 to n do

begin

writeln;

for g := 1 to n do

write(a[i, g]:6:2, ' ');

end;

for i := 1 to n do

for g := 1 to n do

begin

if (a[i, g] > max) then

begin

max := a[i, g];

imax := i;

end;

if (a[i, g] < min) then

begin

min := a[i, g];

imin := i;

end;

end;

for g := 1 to n do

begin

dub := a[imax, g];

a[imax, g] := a[imin, g];

a[imin, g] := dub;

end;

writeln;

writeln('Modified matrix: ');

for i := 1 to n do

begin

writeln;

for g := 1 to n do

write(a[i, g]:6:2, ' ');

end;

writeln;

writeln('max(imax), min(imin): ', max:6:2, '(', imax, ') ', min:6:2, '(',

imin, ')');

end;

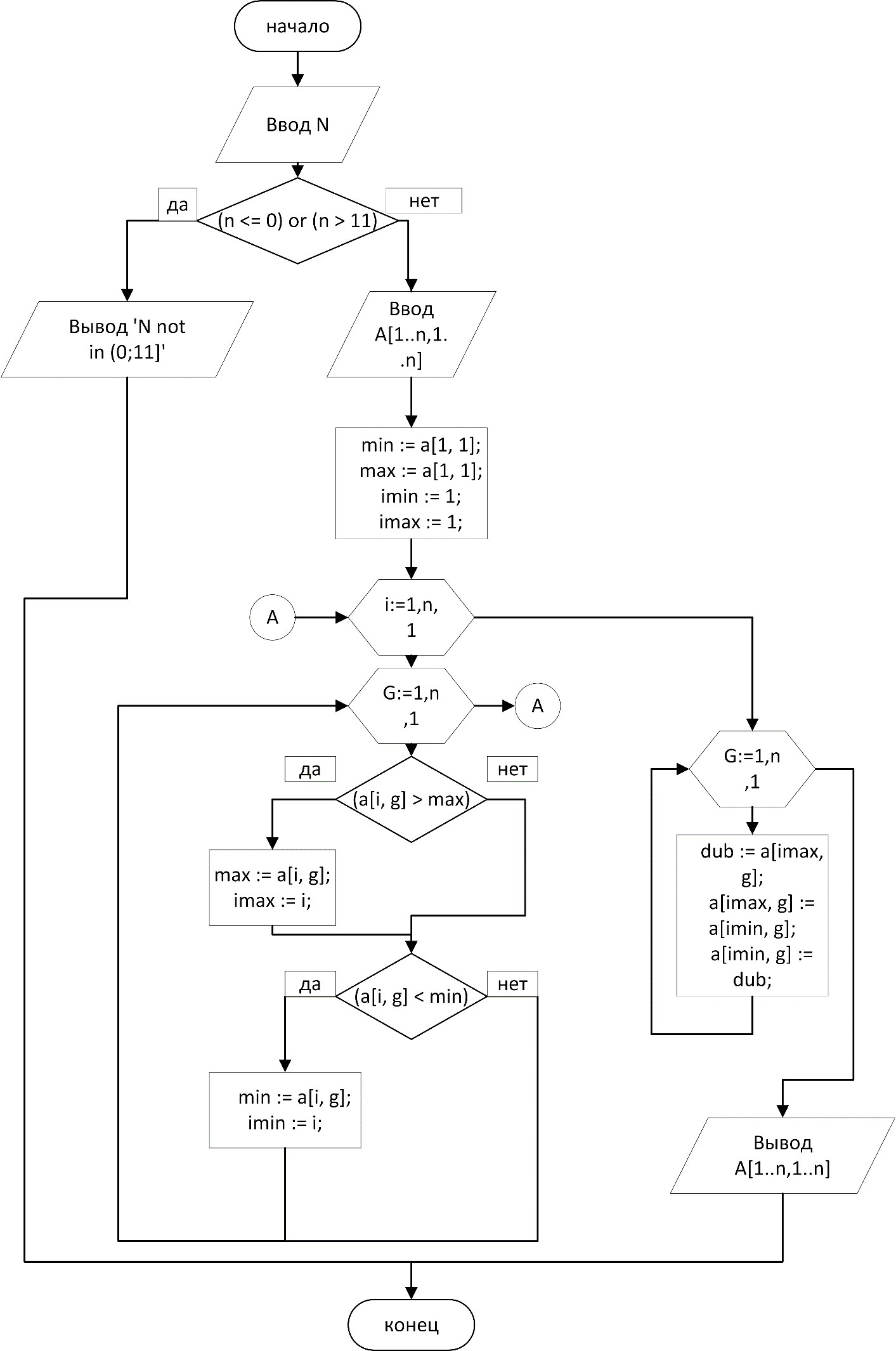
readln;

end.

Тесты

|  |  |  |
| --- | --- | --- |
| Входные данные | Ожидаемые выходные данные | Выходные данные |
| 5  1  2  3  4  5  -5  1  2  3  4  1  2  3  4  5  56  1  2  3  4  7  8  9  10  11 | Inputed matrix:  1.00 2.00 3.00 4.00 5.00  -5.00 1.00 2.00 3.00 4.00  1.00 2.00 3.00 4.00 5.00  56.00 1.00 2.00 3.00 4.00  7.00 8.00 9.00 10.00 11.00  Modified matrix:  1.00 2.00 3.00 4.00 5.00  56.00 1.00 2.00 3.00 4.00  1.00 2.00 3.00 4.00 5.00  -5.00 1.00 2.00 3.00 4.00  7.00 8.00 9.00 10.00 11.00  max(imax), min(imin): 56.00(4) -5.00(2) | Inputed matrix:  1.00 2.00 3.00 4.00 5.00  -5.00 1.00 2.00 3.00 4.00  1.00 2.00 3.00 4.00 5.00  56.00 1.00 2.00 3.00 4.00  7.00 8.00 9.00 10.00 11.00  Modified matrix:  1.00 2.00 3.00 4.00 5.00  56.00 1.00 2.00 3.00 4.00  1.00 2.00 3.00 4.00 5.00  -5.00 1.00 2.00 3.00 4.00  7.00 8.00 9.00 10.00 11.00  max(imax), min(imin): 56.00(4) -5.00(2) |
| 6  1  2  3  4  5  6  7  8  9  10  11  12  21  22  -40  24  25  60  32  33  34  35  36  4  4  4  4  4  4  9  8  7  6  5  4  1 | Inputed matrix:  1.00 2.00 3.00 4.00 5.00 6.00  7.00 8.00 9.00 10.00 11.00 12.00  21.00 22.00 -40.00 24.00 25.00 60.00  32.00 33.00 34.00 35.00 36.00 4.00  4.00 4.00 4.00 4.00 4.00 9.00  8.00 7.00 6.00 5.00 4.00 1.00  Modified matrix:  1.00 2.00 3.00 4.00 5.00 6.00  7.00 8.00 9.00 10.00 11.00 12.00  21.00 22.00 -40.00 24.00 25.00 60.00  32.00 33.00 34.00 35.00 36.00 4.00  4.00 4.00 4.00 4.00 4.00 9.00  8.00 7.00 6.00 5.00 4.00 1.00  max(imax), min(imin): 60.00(3) -40.00(3) | Inputed matrix:  1.00 2.00 3.00 4.00 5.00 6.00  7.00 8.00 9.00 10.00 11.00 12.00  21.00 22.00 -40.00 24.00 25.00 60.00  32.00 33.00 34.00 35.00 36.00 4.00  4.00 4.00 4.00 4.00 4.00 9.00  8.00 7.00 6.00 5.00 4.00 1.00  Modified matrix:  1.00 2.00 3.00 4.00 5.00 6.00  7.00 8.00 9.00 10.00 11.00 12.00  21.00 22.00 -40.00 24.00 25.00 60.00  32.00 33.00 34.00 35.00 36.00 4.00  4.00 4.00 4.00 4.00 4.00 9.00  8.00 7.00 6.00 5.00 4.00 1.00  max(imax), min(imin): 60.00(3) -40.00(3) |
| -2 | N not in (0;11] | N not in (0;11] |

Схема алгоритма



Задание 3

(Составить программу, используя множественный тип.)

Дана последовательность слов, разделенных пробелами, в конце точка. Определить количество слов, в записи которых есть специальные символы #,%,&,@,\*,^. Вывести на экран найденные символы в порядке возрастания их кодов, а также количество слов, удовлетворяющих условию. При отсутствии таких слов выдать соответствующее сообщение.

program Project1;

{$APPTYPE CONSOLE}

uses

System.SysUtils;

type

st1 = string[50];

chrs = set of char;

var

new: boolean;

included, symbs: chrs;

i, k: byte;

a: st1;

begin

symbs := ['#', '%', '&', '@', '\*', '^'];

included := [];

writeln('Enter string');

readln(a);

i := 0;

new := true;

while ((a[i] <> '.') and (i < length(a))) do

begin

i := i + 1;

if (a[i] = ',') then

new := true;

if ((new) and (a[i] in symbs)) then

begin

k := k + 1;

included := included + [a[i]];

new := false;

end;

end;

if (k = 0) then

writeln('No searching words')

else

begin

writeln('Number of found words: ', k);

write('Founded symbs: ');

for i := 0 to 255 do

if (chr(i) in included) then

write(chr(i), ' ');

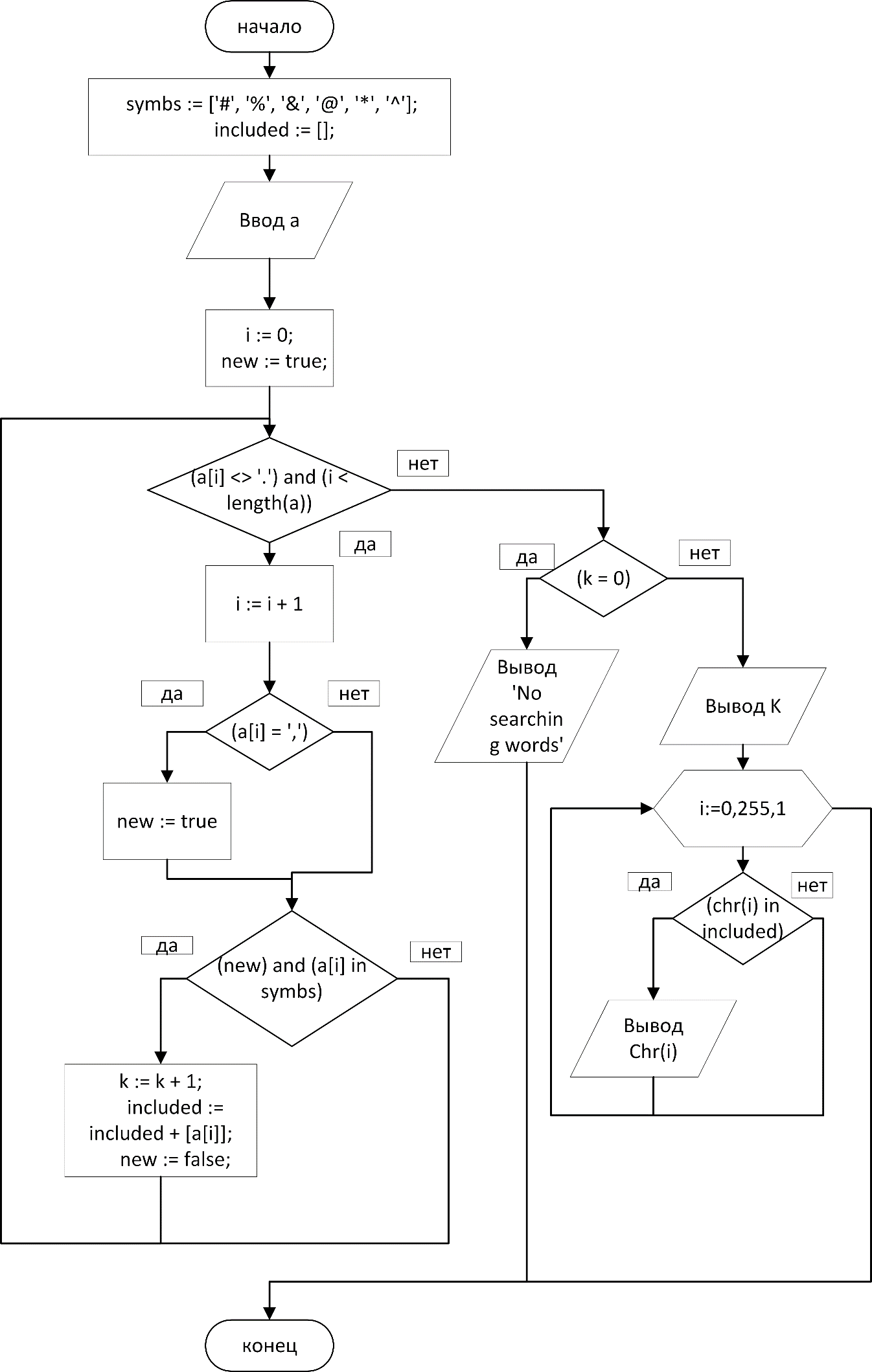
end;

readln;

end.

Тесты

|  |  |  |
| --- | --- | --- |
| Входные данные | Ожидаемые выходные данные | Выходные данные |
| df@, hgf, fvb#. | Number of found words: 2  Founded symbs: # @ | Number of found words: 2  Founded symbs: # @ |
| hj, mku,lk,bc. | No searching words | No searching words |
| hj&, mn@,###, ggg. | Number of found words: 3  Founded symbs: # & @ | Number of found words: 3  Founded symbs: # & @ |

Схема алгоритма:

Задание 4

Разработать модуль, содержащий указанную процедуру. Написать тестирующую программу.

Составить подпрограмму-процедуру MINMAX, отыскивающую x[a, b], для которых функция y=f(x) принимает максимальное и минимальное значение с точностью 0,01.

Функцию f(x) передать в процедуру через параметр. В основной программе использовать процедуру для функций y=(x-1)/(x+2); x[0, 2] и y=sin(x/2-1), x[-1, 1].

Текст программы:

program Project1;

{$APPTYPE CONSOLE}

uses

System.SysUtils,

LabLib1;

function fx1(x: real): real;

begin

result := (x - 1) / (x + 2);

end;

function fx2(x: real): real;

begin

result := sin((x / 2) - 1); {abs(x-0.3) to check}

end;

var

minx1, maxx1, minx2, maxx2: real;

begin

minmax(fx1, 0, 2, 0.01, minx1, maxx1);

minmax(fx2, -1, 1, 0.01, minx2, maxx2);

writeln('x of min (x - 1) / (x + 2): ', minx1:6:2, ' x of max: ', maxx1:6:2);

writeln('x of min sin((x / 2) - 1): ', minx2:6:2, ' x of max: ', maxx2:6:2);

readln;

end.

Текст модуля:

unit LabLib1;

interface

type

funcOf1 = function(arg: real): real;

procedure minmax(f: funcOf1; a, b, eps: real; var minarg, maxarg: real);

implementation

procedure minmax;

var

tempArg: real;

begin

tempArg := a;

minarg := tempArg;

maxarg := tempArg;

while (tempArg <= b) do

begin

if (f(tempArg) < f(minarg)) then

minarg := tempArg;

if (f(tempArg) > f(maxarg)) then

maxarg := tempArg;

tempArg := tempArg + eps;

end;

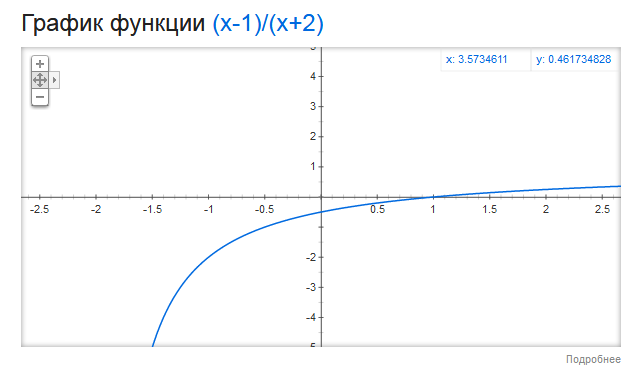
end;

end.

Тесты:

|  |  |  |
| --- | --- | --- |
| Входные данные | Ожидаемые выходные данные | Выходные данные |
| - | x of min (x - 1) / (x + 2): 0.00 x of max: 1.99  x of min sin((x / 2) - 1): -1.00 x of max: 0.99 | x of min (x - 1) / (x + 2): 0.00 x of max: 1.99  x of min sin((x / 2) - 1): -1.00 x of max: 0.99 |

Графики для проверки:



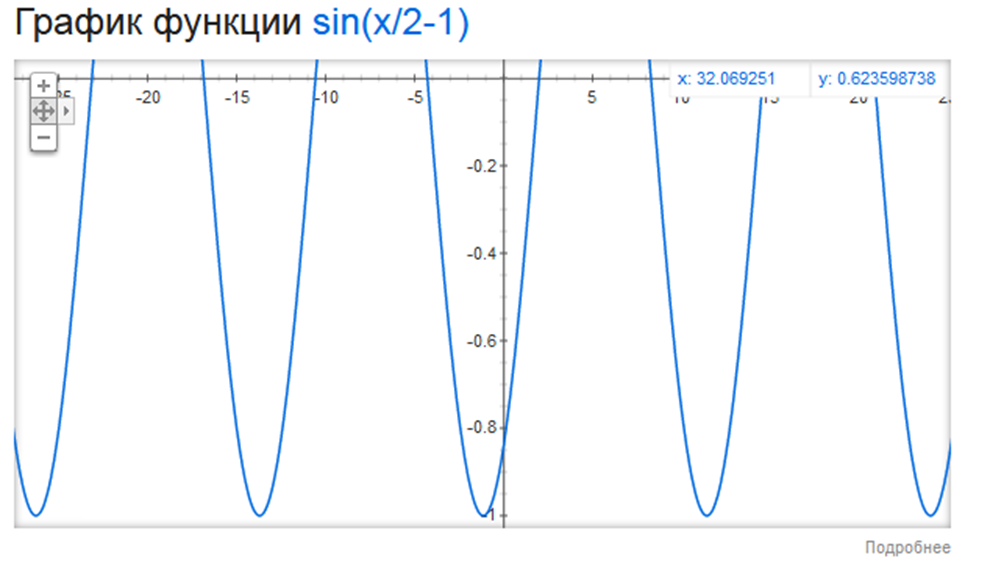
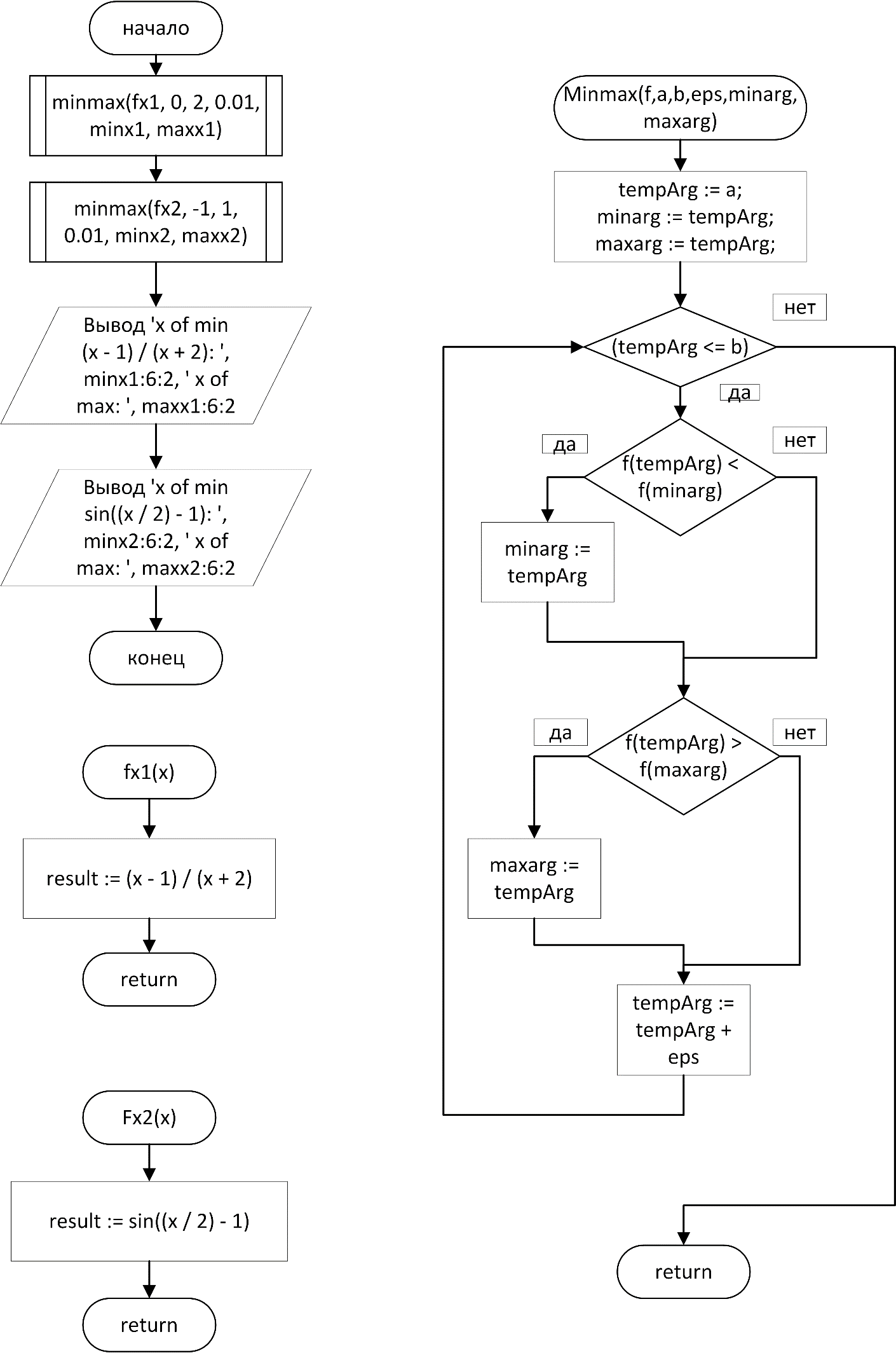


Схема алгоритма:



Вывод:

* сортировку массива можно осуществлять по различным критериям и различными алгоритмами.
* многие операции в матрицах проще осуществлять по индексам.
* множества облегчают решения многих типов задач, например, проверку вхождения элемента в последовательность(т.е. символа в строку).
* модули позволяют более удобно хранить часто используемые подпрограммы, чтобы их можно было вызывать из модулей, а не переписывать в каждой программе.