Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра программного обеспечения информационных технологий

Дисциплина: Основы алгоритмизации и программирования (ОАиП)

ОТЧЕТ

по лабораторной работе №4

Тема работы:

Работа с многомерными массивами

Выполнил

студент: гр. 151004 Данилов Ф.А.

Проверил: Фадеева Е.П.

Минск 2021

Содержание

[1 Постановка задачи 3](#_Toc87565823)

[2 Ход решения 4](#_Toc87565824)

[3 Текстовый алгоритм решения задачи 5](#_Toc87565825)

[4 Структура данных 7](#_Toc87565826)

[5 Схема алгоритма решения задачи по ГОСТ 19.701-90 8](#_Toc87565827)

[6 Результаты расчетов 12](#_Toc87565828)

[Приложение А 13](#_Toc87565829)

[Приложение Б 18](#_Toc87565830)

[Приложение В 21](#_Toc87565831)

[Приложение Г 24](#_Toc87565832)

# Постановка задачи

Дан целочисленный массив  размерности . Каждая строка массива упорядочена по неубыванию. Найти числа, встречающиеся во всех строках.

# Ход решения

Для того, чтобы решить задачу необходимо проверить каждый элемент первой строки и сравнить его с элементами других строк. Так как по условию задачи строки первоначального массива отсортированы по неубыванию, можно оптимизировать задачу, не сравнивая элементы первой строки со всеми элементами других строк.

Необходимо сравнить каждый элемент первой строки с элементами других строк. В начале цикла значение Count = 1. Если элемент первой строки меньше или равен элементу другой строки цикл сравнивания останавливается.

Если элемент первой строки присутствует в другой строке, значение *Count* инкрементируется*.* Иначе процесс сравнивания приостанавливается, так как элемент первой строки не присутствует во всех строках массива.

После проверки всех строк, если *Count = N*, элемент первой строки присутствует в каждой строке и мы добавляем его в массив *Num*. После проверки всех элементов первой строки, мы выводим уникальные элементы массива *Num*.

# Текстовый алгоритм решения задачи

Таблица – Алгоритм решения

|  |  |
| --- | --- |
| Номер  шага | Назначение шага |
|  | N := 4 |
|  | Ввод X[1..N, 1..N] |
|  | M := 0 |
|  | j := 1 |
|  | Начало цикла А1. Проверка выполнения условия (j ≤ N). Если условие истинно, идти к шагу 6, иначе – к шагу 27 |
|  | Count := 1 |
|  | IsAbsent := false |
|  | i := 2 |
|  | Начало цикла А2. Проверка выполнения условия (i ≤ N and not IsAbsent). Если условие истинно, идти к шагу 10, иначе – к шагу 22 |
|  | k := 1 |
|  | IsFound := false |
|  | Начало цикла А3. Проверка выполнения условия (X[1,j] ≥ X[i,k] and k ≤ N and not IsFound). Если условие истинно, идти к шагу 13, иначе – к шагу 18 |
|  | Проверка выполнения условия (X[1,j] = X[i,k). Если условие истинно, идти к шагу 14, иначе – к шагу 16 |
|  | IsFound := true |
|  | Count := Count + 1 |
|  | k := k + 1 |
|  | Конец цикла А3. Идти к шагу 12 |
|  | Проверка выполнения условия (IsFound = false). Если условие истинно, идти к шагу 19, иначе – к шагу 20 |
|  | IsAbsent := true |
|  | i := i + 1 |
|  | Конец цикла А2. Идти к шагу 9 |
|  | Проверка выполнения условия (Count = N). Если условие истинно, идти к шагу 23, иначе – к шагу 25 |
|  | M := M + 1 |
|  | Num[M] := X[1,j] |
|  | j := j + 1 |
|  | Конец цикла А1. Идти к шагу 5 |
|  | Проверка выполнения условия (M > 0). Если условие истинно, идти к шагу 32, иначе – к шагу 39 |
|  | Вывод B[1] |
|  | i := 2 |

Продолжение таблицы 1

|  |  |
| --- | --- |
| 30 | Начало цикла А6. Проверка выполнения условия (i ≤ M). Если условие истинно, идти к шагу 35, иначе – к шагу 39 |
| 31 | Проверка выполнения условия (Num[i] ≠ Num[i-1]). Если условие истинно, идти к шагу 36, иначе – к шагу 37 |
| 32 | Вывод Num[i] |
| 33 | i := i + 1 |
| 34 | Конец цикла А6. Идти к шагу 34 |
| 35 | Останов. |

# Структура данных

Таблица – Данные

|  |  |  |
| --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение |
| N | Integer | Размер массива *X* |
| M | Integer | Размер массива *Num* |
| X | Array [1..N, 1..N] of Integer | Первоначальный массив |
| Num | Array[1..M] of Integer | Массив чисел, которые соответствуют условиям постановки |
| i, j, k | Integer | Счетчики циклов |
| IsFound | Boolean | Переменная для проверки нахождения одинакового числа в строке |
| IsAbsent | Boolean | Переменная для проверки отсутствия элемента во всех строках |
| Count | Integer | Счетчик найденных чисел в строках |

# Схема алгоритма решения задачи по ГОСТ 19.701-90

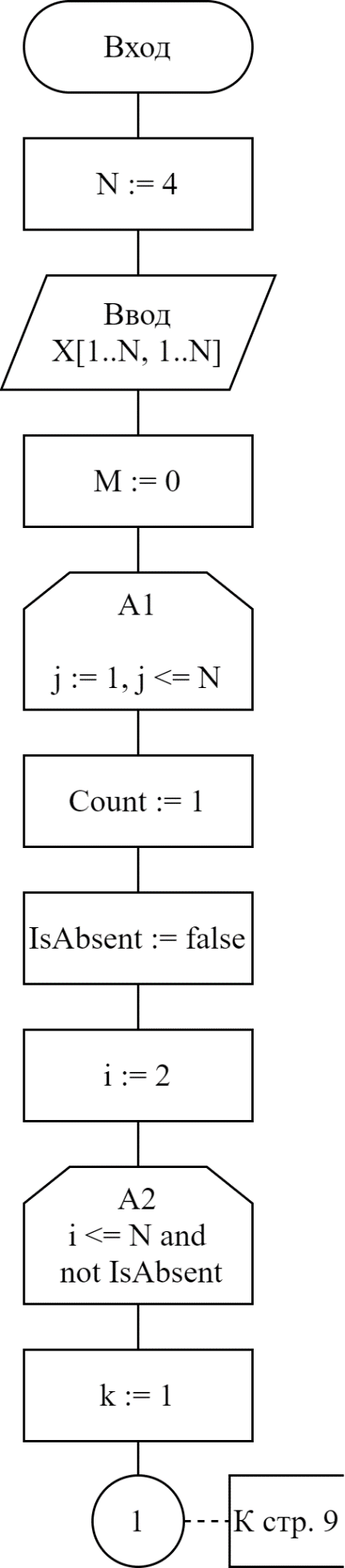


Рисунок – Схема алгоритма решения задачи по ГОСТ 19.701-90 (1 часть)

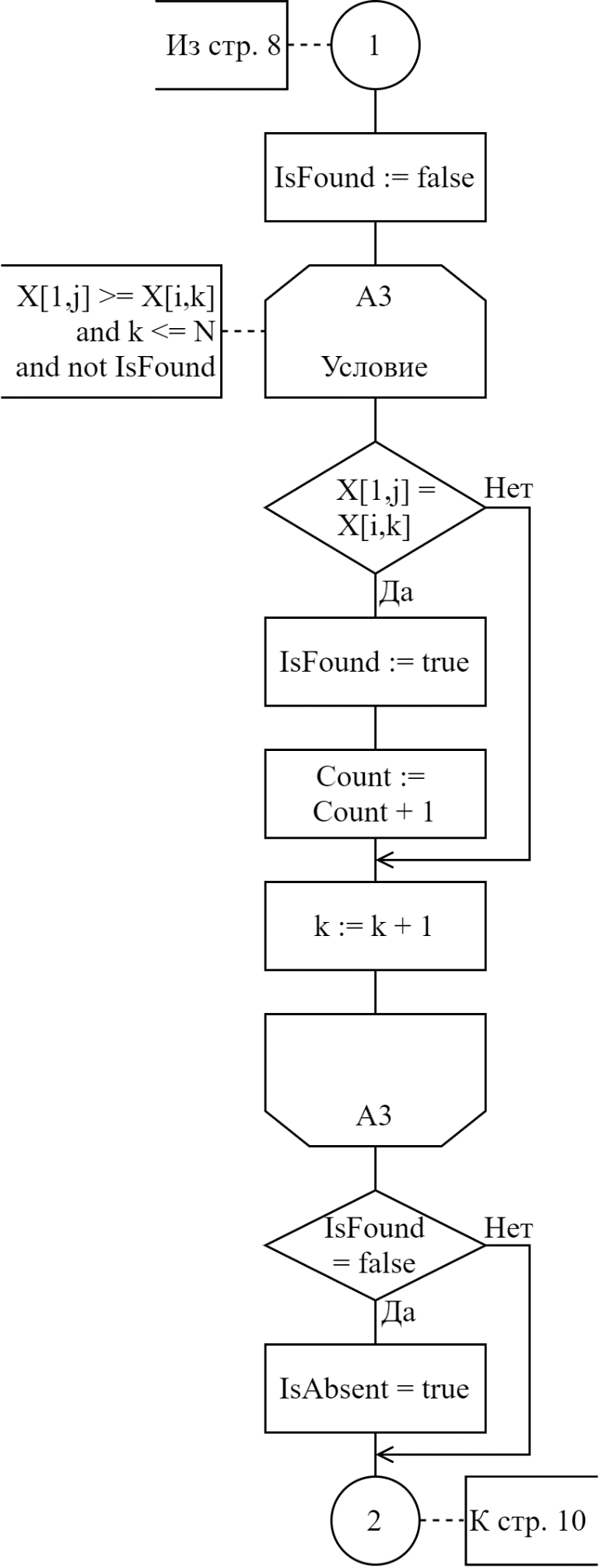


Рисунок 2 – Схема алгоритма решения задачи по ГОСТ 19.701-90 (2 часть)

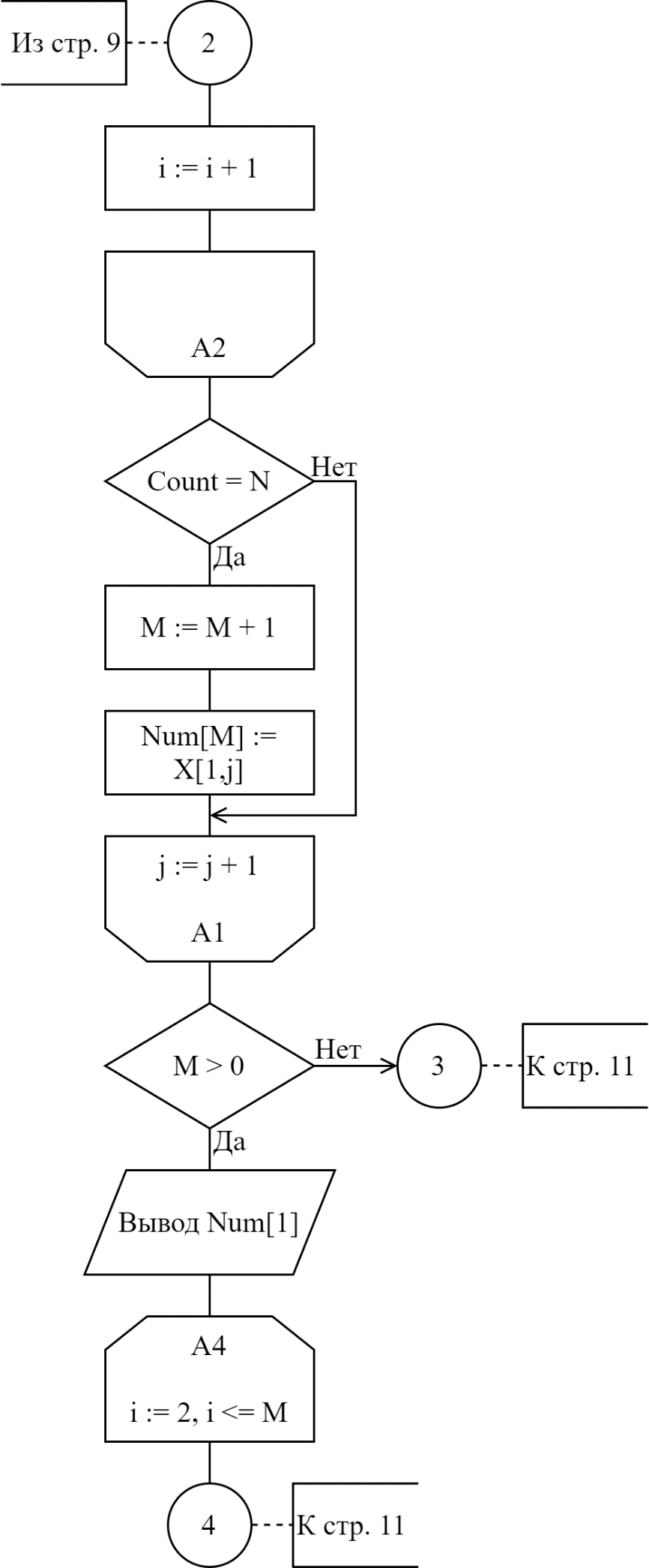


Рисунок 3 – Схема алгоритма решения задачи по ГОСТ 19.701-90 (3 часть)

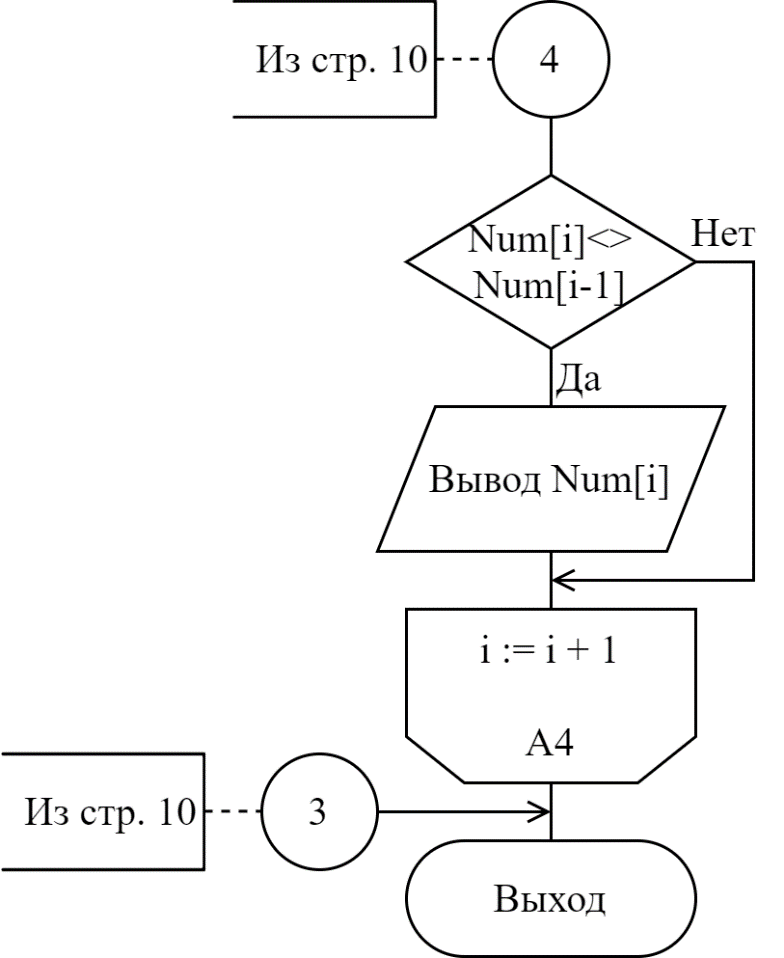


Рисунок 4 – Схема алгоритма решения задачи по ГОСТ 19.701-90 (4 часть)

# Результаты расчетов

Вследствие выполнения программы на экран выводятся следующие результаты:

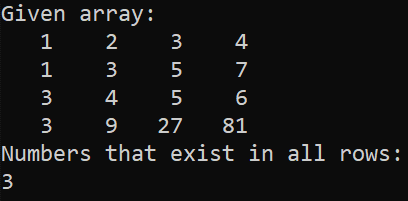


Рисунок 5 – Результаты расчётов

Приложение А

(обязательное)

Исходный код программы

program Lab4\_1;

{

Given matrix X size N\*N find all numbers that exist in all rows.

Input: matrix X size N\*N

Output: numbers that exist in all rows

}

{$APPTYPE CONSOLE}

{$R \*.res}

uses

System.SysUtils;

// N - size of matrix

const

N = 4;

// X - given array

// Num - array to store found numbers

// M - size of Num

// i, j, k - counts for loops

// Count - count of numbers in rows

// Error - value for input

// InputStr - string for input

// IsFound - boolean to check if same number was

// found in a row

// IsAbsent - boolean to check if no same numbers exist

// in table

var

i, j, k, Error, Count, M: Integer;

InputStr: String;

IsFound, IsAbsent: Boolean;

X: Array [1..N, 1..N] of Integer;

Num: Array [1..N] of Integer;

begin

// Input X using following methods:

// Iterate through rows from i to N

// Iterate throuth columns from i to N

// Input integer number using method Val

// In each input check if previous value is smaller

// than the current one

for i := 1 to N do

begin

writeln('Enter array X[', i, ']:');

for j := 1 to N do

repeat

readln(InputStr);

val(InputStr, X[i,j], Error);

IsFound := true;

if (j > 1) then

if X[i,j] < X[i,j-1] then

begin

writeln('Enter value, not less than ',

X[i,j-1]);

IsFound := false;

end;

until (Error = 0) and IsFound;

end;

// Output given array

writeln('Given array:');

for i := 1 to N do

begin

for j := 1 to N do

write(X[i,j]:4, ' ');

writeln;

end;

// Algorithm to find same nubmers

M := 0;

// Iterate through columns of first row

for j := 1 to N do

begin

Count := 1;

IsAbsent := false;

i := 2;

// Iterate through rows

// If element is absent in one of rows stop cycle

while (i <= N) and (not IsAbsent) do

begin

k := 1;

IsFound := false;

// Iterate through columns of other rows

while (X[1,j] >= X[i,k]) and

(k <= N) and (not IsFound) do

begin

// Check if elements are the same

if X[1,j] = X[i,k] then

begin

IsFound := true;

// Increment count for same elements in each row

Count := Count + 1;

end;

// Move to another element of row i

k := k + 1;

end;

// Check if element is absent in row i

if IsFound = false then

IsAbsent := true;

// Move to another row

i := i + 1;

end;

if Count = N then

begin

M := M + 1;

Num[M] := X[1,j];

end;

end;

// Output unique Num values

if M > 0 then

begin

writeln('Numbers that exist in all rows:');

write(Num[1], ' ');

for i := 2 to M do

if Num[i] <> Num[i-1] then

write(Num[i], ' ');

end

else

writeln('No such numbers found :(');

// Stop console from closing

readln;

end.

Приложение Б

(обязательное)

Тестовые наборы

Тестовые наборы для проверки вычислений

Тест 1

Тест: апргшгрпаапролшщлшорпапрол

Исходные данные:

X = ((1, 2, 3, 4),

(1, 3, 5, 7),

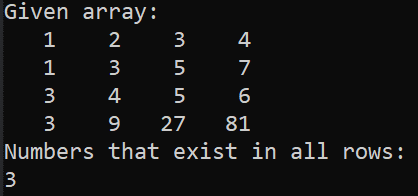
(3, 4, 5, 6),

(3, 9, 27, 81))

Ожидаемый результат:

3

Полученный результат:



Тест 2

Исходные данные:

X = ((1, 2, 3, 4),

(1, 2, 3, 4),

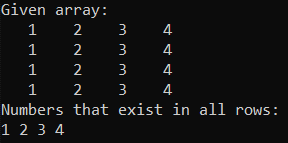
(1, 2, 3, 4),

(1, 2, 3, 4))

Ожидаемый результат:

1, 2, 3, 4

Полученный результат:



Тест 3

Исходные данные:

X = ((1, 2, 3, 4),

(5, 6, 7, 8),

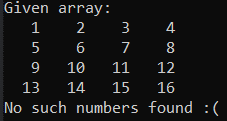
(9, 10, 11, 12),

(13, 14, 15, 16))

Ожидаемый результат:

Числа не найдены

Полученный результат:



Приложение Б

(обязательное)

Дополнительный код программы №1

program Lab4\_2;

{

Given random generated matrix X size N\*N

find all numbers that exist in all rows.

Output: numbers that exist in all rows

}

{$APPTYPE CONSOLE}

{$R \*.res}

uses

System.SysUtils;

// N - size of matrix

const

N = 4;

// X - given array

// Num - array to store found numbers

// M - size of Num

// i, j, k - counts for loops

// Count - count of numbers in rows

// IsFound - boolean to check if same number was

// found in a row

// IsAbsent - boolean to check if no same numbers exist

// in table

var

i, j, k, Count, M: Integer;

IsFound, IsAbsent: Boolean;

X: Array [1..N, 1..N] of Integer;

Num: Array [1..N] of Integer;

begin

// Input X

randomize;

for i := 1 to N do

begin

for j := 1 to N do

repeat

X[i,j] := random(100);

IsFound := true;

if (j > 1) then

if X[i,j] < X[i,j-1] then

IsFound := false;

until IsFound;

end;

// Output given array

writeln('Given array:');

for i := 1 to N do

begin

for j := 1 to N do

write(X[i,j]:4, ' ');

writeln;

end;

// Algorithm to find same nubmers

M := 0;

// Iterate through columns of first row

for j := 1 to N do

begin

Count := 1;

IsAbsent := false;

i := 2;

// Iterate through rows

// If element is absent in one of rows stop cycle

while (i <= N) and (not IsAbsent) do

begin

k := 1;

IsFound := false;

// Iterate through columns of other rows

while (X[1,j] >= X[i,k]) and (k <= N)

and (not IsFound) do

begin

// Check if elements are the same

if X[1,j] = X[i,k] then

begin

IsFound := true;

// Increment count for same elements in each row

Count := Count + 1;

end;

// Move to another element of row i

k := k + 1;

end;

// Check if element is absent in row i

if IsFound = false then

IsAbsent := true;

// Move to another row

i := i + 1;

end;

if Count = N then

begin

M := M + 1;

Num[M] := X[1,j];

end;

end;

// Output unique Num values

if M > 0 then

begin

writeln('Numbers that exist in all rows:');

write(Num[1], ' ');

for i := 2 to M do

if Num[i] <> Num[i-1] then

write(Num[i], ' ');

end

else

writeln('No such numbers found :(');

// Stop console from closing

readln;

end.

Приложение В

(обязательное)

Дополнительный код программы №2

program Lab4\_3;

{

Given matrix X size N\*N find all numbers that exist in all rows.

Output: numbers that exist in all rows

}

{$APPTYPE CONSOLE}

{$R \*.res}

uses

System.SysUtils;

// N - size of matrix

const

N = 4;

Arr: array[1..4, 1..4] of Integer =

((1, 2, 3, 4),

(1, 3, 4, 5),

(1, 2, 3, 4),

(1, 3, 5, 7));

// X - given array

// Num - array to store found numbers

// M - size of Num

// i, j, k - counts for loops

// Count - count of numbers in rows

// IsFound - boolean to check if same number was found in a row

// IsAbsent - boolean to check if no same numbers exist in table

var

i, j, k, Count, M: Integer;

IsFound, IsAbsent: Boolean;

X: Array [1..N, 1..N] of Integer;

Num: Array [1..N] of Integer;

begin

// Input X

for i := 1 to N do

for j := 1 to N do

X[i,j] := Arr[i,j];

// Output given array

writeln('Given array:');

for i := 1 to N do

begin

for j := 1 to N do

write(X[i,j]:4, ' ');

writeln;

end;

// Algorithm to find same nubmers

M := 0;

// Iterate through columns of first row

for j := 1 to N do

begin

Count := 1;

IsAbsent := false;

i := 2;

// Iterate through rows

// If element is absent in one of rows stop cycle

while (i <= N) and (not IsAbsent) do

begin

k := 1;

IsFound := false;

// Iterate through columns of other rows

while (X[1,j] >= X[i,k]) and (k <= N) and (not IsFound) do

begin

// Check if elements are the same

if X[1,j] = X[i,k] then

begin

IsFound := true;

// Increment count for same elements in each row

Count := Count + 1;

end;

// Move to another element of row i

k := k + 1;

end;

// Check if element is absent in row i

if IsFound = false then

IsAbsent := true;

// Move to another row

i := i + 1;

end;

if Count = N then

begin

M := M + 1;

Num[M] := X[1,j];

end;

end;

// Output unique Num values

if M > 0 then

begin

writeln('Numbers that exist in all rows:');

write(Num[1], ' ');

for i := 2 to M do

if Num[i] <> Num[i-1] then

write(Num[i], ' ');

end

else

writeln('No such numbers found :(');

// Stop console from closing

readln;

end.

Приложение Г

(обязательное)

Дополнительный код программы №3

program Lab4\_4;

{

Given matrix X size N\*N find all numbers that exist in all rows.

Output: numbers that exist in all rows

}

{$APPTYPE CONSOLE}

{$R \*.res}

uses

System.SysUtils;

// N - size of matrix

const

N = 4;

// X - given array

// Num - array to store found numbers

// M - size of Num

// i, j, k - counts for loops

// Count - count of numbers in rows

// IsFound - boolean to check if same number was found in a row

// IsAbsent - boolean to check if no same numbers exist in table

var

i, j, k, Count, M: Integer;

IsFound, IsAbsent: Boolean;

X: Array [1..N, 1..N] of Integer =

((1, 3, 5, 7),

(1, 2, 3, 4),

(0, 1, 2, 3),

(-1, 1, -3, 3));

Num: Array [1..N] of Integer;

begin

// Output given array

writeln('Given array:');

for i := 1 to N do

begin

for j := 1 to N do

write(X[i,j]:4, ' ');

writeln;

end;

// Algorithm to find same nubmers

M := 0;

// Iterate through columns of first row

for j := 1 to N do

begin

Count := 1;

IsAbsent := false;

i := 2;

// Iterate through rows

// If element is absent in one of rows stop cycle

while (i <= N) and (not IsAbsent) do

begin

k := 1;

IsFound := false;

// Iterate through columns of other rows

while (X[1,j] >= X[i,k]) and (k <= N) and (not IsFound) do

begin

// Check if elements are the same

if X[1,j] = X[i,k] then

begin

IsFound := true;

// Increment count for same elements in each row

Count := Count + 1;

end;

// Move to another element of row i

k := k + 1;

end;

// Check if element is absent in row i

if IsFound = false then

IsAbsent := true;

// Move to another row

i := i + 1;

end;

if Count = N then

begin

M := M + 1;

Num[M] := X[1,j];

end;

end;

// Output unique Num values

if M > 0 then

begin

writeln('Numbers that exist in all rows:');

write(Num[1], ' ');

for i := 2 to M do

if Num[i] <> Num[i-1] then

write(Num[i], ' ');

end

else

writeln('No such numbers found :(');

// Stop console from closing

readln;

end.