Министерство образования Республики Беларусь

Учреждение образования

БелорусскиЙ государственный университет

информатики и радиоэлектроники

Факультет компьютерных систем и сетей

Кафедра программного обеспечения информационных технологий

|  |
| --- |
|  |
|  |

# оТЧЕТ

по лабораторной работе

на тему:

Массивы

|  |  |  |
| --- | --- | --- |
| Выполнил  Студент гр. 451001 |  | Д. Д. Хренков |
| Проверил |  | Асс. Е.Е. Фадеева |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Минск, 2024

Теоретические сведения по теме лабораторной работы

Массив – упорядоченная совокупность однотипных элементов, имеющих общее имя. Данное имя называется полной переменной, так как ее значением является весь массив.  
Тип элементов массива называется базовым типом массива.  
Для выделения отдельных элементов массива используются индексы (индексные выражения). Индекс задает правило вычисления номера нужного элемента массива.  
Элементы массива называются индексированными переменными. Для обращения к конкретному элементу массива необходимо указывать имя массива и индексы элемента в квадратных скобках.  
Элементы массива могут использоваться в выражениях аналогично простым переменным и вместе с простыми переменными.  
Полные переменные типа массив не могут использоваться в качестве операндов выражений.  
Значения полных переменных могут быть использованы только в операторах присваивания, причем полной переменной может быть присвоено только значение полной переменной того же типа. n-мерный массив – массив, в котором для идентификации элемента необходимо n индексов.

Двумерный массив – это массив, элементами которого являются одномерные массивы.

1. Задание на лабораторную работу

Постановка задачи

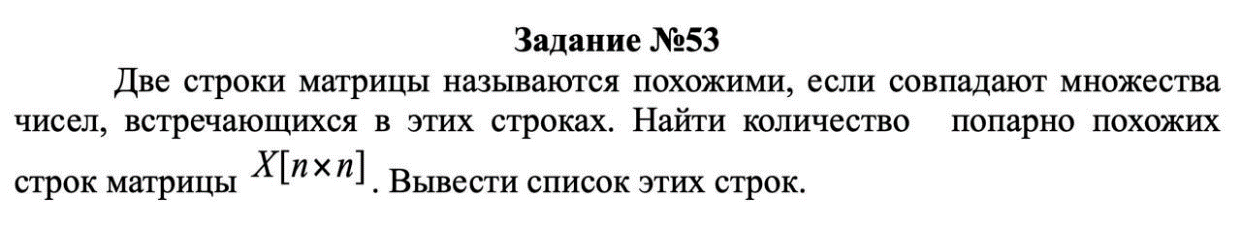
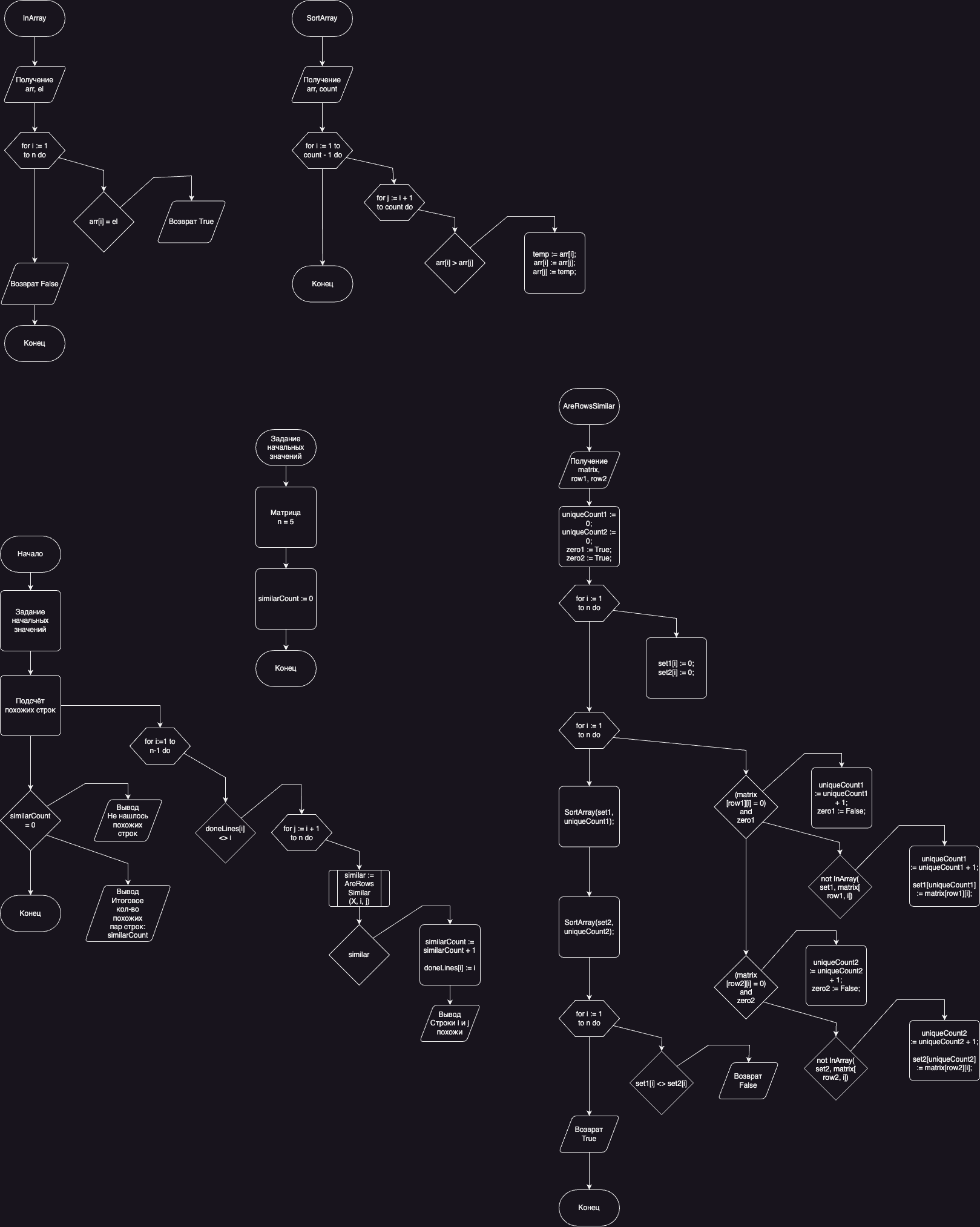


Схема алгоритма по методу Данке



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя подпрограммы | Назначение подпрограммы | Заголовок подпрограммы | Имя параметра | Назначение параметра |
| InArray | Проверяет вхождение элемента в массив | function InArray(var arr: TMAS; const el: Integer): Boolean; | arr | Массив, по которому осуществляется поиск |
| el | Элемент для нахождения |
| SortArray | Сортирует элементы массива от меньшего к большему | procedure SortArray(var arr: TMAS; count: Integer); | arr | Массив для сортироки |
| count | Сколько элементов необходимо отсортировать |
| AreRowsSimilar | Проверяет, являются ли строки матрицы похожими | function AreRowsSimilar(var matrix: TMatrix; const row1, row2: Integer): Boolean; | matrix | Матрица, строки которой сравниваются |
| row1 | Индекс первой строки |
| row2 | Индекс второй строки |

Выделение основных структур данных

|  |  |  |
| --- | --- | --- |
| Имя идентификатора структуры | Назначение структуры | Тип структуры |
| TMatrix | Матрица чисел | array[1..n,1..n] of Integer |
| TMAS | Массив чисел | array[1..n] of Integer |

Тестирование и отладка программы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Специфика тестирования | Номер теста | Вводимые данные | Ожидаемый результат | Полученный результат |
| Матрица со всеми одинаковыми строками | 1 | (  (0, 0, 0, 0, 0, 0),  (0, 0, 0, 0, 0, 0),  (0, 0, 0, 0, 0, 0),  (0, 0, 0, 0, 0, 0),  (0, 0, 0, 0, 0, 0),  (0, 0, 0, 0, 0, 0)  ) | Строки 1 и 2 похожи  Строки 1 и 3 похожи  Строки 1 и 4 похожи  Строки 1 и 5 похожи  Строки 1 и 6 похожи  Строки 2 и 3 похожи  Строки 2 и 4 похожи  Строки 2 и 5 похожи  Строки 2 и 6 похожи  Строки 3 и 4 похожи  Строки 3 и 5 похожи  Строки 3 и 6 похожи  Строки 4 и 5 похожи  Строки 4 и 6 похожи  Строки 5 и 6 похожи  Итоговое кол-во похожих пар строк: 15 | Как и ожидалось |
| Матрица со всеми непохожими строками | 2 | (  (0, 0, 0, 0, 0, 0),  (1, 1, 1, 1, 1, 1),  (2, 2, 2, 2, 2, 2),  (3, 3, 3, 3, 3, 3),  (4, 4, 4, 4, 4, 4),  (5, 5, 5, 5, 5, 5)  ) | Не нашлось похожих строк | Как и ожидалось |
| Матрица со всеми непохожими строками, где отличается лишь один элемент | 3 | (  (0, 0, 0, 0, 0, 0),  (1, 0, 0, 0, 0, 0),  (2, 0, 0, 0, 0, 0),  (3, 0, 0, 0, 0, 0),  (4, 0, 0, 0, 0, 0),  (5, 0, 0, 0, 0, 0)  ) | Не нашлось похожих строк | Как и ожидалось |
| Матрица, где есть n/2 похожих пар строк | 4 | (  (0, 0, 0, 0, 0, 0),  (1, 0, 0, 0, 0, 0),  (2, 0, 0, 0, 0, 0),  (0, 0, 0, 0, 0, 0),  (2, 0, 0, 0, 0, 0),  (1, 0, 0, 0, 0, 0)  ) | Строки 1 и 4 похожи  Строки 2 и 6 похожи  Строки 3 и 5 похожи  Итоговое кол-во похожих пар строк: 3 | Как и ожидалось |

program lab6;

const

  n = 6;

type

  TMatrix = array [1 .. n, 1 .. n] of Integer;

  TMAS = array [1 .. n] of Integer;

function InArray(var arr: TMAS; const el: Integer): Boolean;

var

  i: Integer;

begin

  for i := 1 to n do

  begin

    if arr[i] = el then

      exit(True);

  end;

  InArray := False;

end;

procedure SortArray(var arr: TMAS; count: Integer);

var

  i, j, temp: Integer;

begin

  for i := 1 to count - 1 do

  begin

    for j := i + 1 to count do

    begin

      if arr[i] > arr[j] then

      begin

        temp := arr[i];

        arr[i] := arr[j];

        arr[j] := temp;

      end;

    end;

  end;

end;

function AreRowsSimilar(var matrix: TMatrix; const row1, row2: Integer)

  : Boolean;

var

  set1, set2: TMAS;

  i, uniqueCount1, uniqueCount2: Integer;

  zero1, zero2: Boolean;

begin

  uniqueCount1 := 0;

  uniqueCount2 := 0;

  zero1 := True;

  zero2 := True;

  for I := 1 to n do

  begin

    set1[i] := 0;

    set2[i] := 0;

  end;

  for i := 1 to n do

  begin

    if ((matrix[row1][i] = 0) and zero1) then

    begin

      Inc(uniqueCount1);

      zero1 := False;

    end

    else if not InArray(set1, matrix[row1][i]) then

    begin

      Inc(uniqueCount1);

      set1[uniqueCount1] := matrix[row1][i];

    end;

    if ((matrix[row2][i] = 0) and zero2) then

    begin

      Inc(uniqueCount2);

      zero2 := False;

    end

    else if not InArray(set2, matrix[row2][i]) then

    begin

      Inc(uniqueCount2);

      set2[uniqueCount2] := matrix[row2][i];

    end;

  end;

  SortArray(set1, uniqueCount1);

  SortArray(set2, uniqueCount2);

  for i := 1 to n do

  begin

    if set1[i] <> set2[i] then

      exit(false);

  end;

  AreRowsSimilar := True;

end;

var

  X: TMatrix = (

    (1, 0, 0, 0, 0, 0),

    (2, 0, 0, 0, 0, 0),

    (3, 0, 0, 0, 0, 0),

    (3, 0, 0, 0, 0, 0),

    (2, 0, 0, 0, 0, 0),

    (1, 0, 0, 0, 0, 0)

  );

  i, j: Integer;

  similarCount: Integer;

  doneLines: TMAS;

  similar: Boolean;

begin

  similarCount := 0;

  for i := 1 to n - 1 do

  begin

    if doneLines[i] <> i then

    begin

      for j := i + 1 to n do

      begin

        similar := AreRowsSimilar(X, i, j);

        if similar then

        begin

          Inc(similarCount);

          doneLines[i] := i;

          WriteLn('Строки ', i, ' и ', j, ' похожи');

        end;

      end;

    end;

  end;

  if similarCount = 0 then

    WriteLn('Не нашлось похожих строк')

  else

    WriteLn('Итоговое кол-во похожих пар строк: ', similarCount);

  ReadLn;

end.