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: Многомерные массивы

Выполнил

Проверил: Фадеева Е.П.

Минск 2021

содержание

[1 Постановка задачи 3](#_Toc88933822)

[2 Методика решения 4](#_Toc88933823)

[3 Текстовый алгоритм решения задачи 5](#_Toc88933824)

[4 Структура данных 7](#_Toc88933825)

[5 Схема алгоритма решения задачи по Гост 19.701-90 8](#_Toc88933826)

[6 Результаты расчетов 10](#_Toc88933827)

[Приложение А 11](#_Toc88933828)

[Ввод данных с клавиатуры 11](#_Toc88933829)

[Приложение Б 14](#_Toc88933830)

[Генератор равномерно-распределённых чисел 14](#_Toc88933831)

[Приложение В 17](#_Toc88933832)

[Типизированная константа-массив 17](#_Toc88933833)

[Приложение Г 20](#_Toc88933834)

[Типизированная переменная-массив 20](#_Toc88933835)

[Приложение Д 23](#_Toc88933836)

[Тестовые наборы 23](#_Toc88933837)

# Постановка задачи

Дана матрица размерности . Получить одномерный массив размерности , выбирая элементы матрицы по цепи, начиная с верхнего правого угла, двигаясь по диагоналям, параллельным главной диагонали матрицы.

Вывести на печать результаты расчётов в виде:

= (значение 1, значение 2… значение)

# Методика решения

В данной лабораторной работе необходимо вывести все элементы матрицы «цепочкой» из крайнего верхнего правого угла в крайний нижний левый.

На некоторых диагоналях числа по цепочке будут перечисляться сверху вниз, а на некоторых – снизу вверх. Чередование таких диагоналей идёт через одну, поэтому можно применить свойство чётности/нечётности диагонали, чтобы определить, как именно будут перечисляться элементы на ней.

Количество диагоналей квадратной матрицы можно определить по формуле Diagonal:= 2 \* Size – 1.

В матрице есть несколько типов диагоналей. Главной диагональю является диагональ с номером, равным количеству столбцов/строк матрицы. Диагонали с большим номером расположены над ней, с меньшим – под ней. Формула расчёта позиции элемента отличается для этих двух типов диагоналей, а главную можно отнести как к одной формуле, так и к другой.

В решении используется цикл для перебора диагоналей и два условия для определения типа диагонали, которые, в свою очередь, содержат в себе по два условия для определения чётности/нечётности диагонали. В итоге программа имеет четыре формулы расчёта.

1. Номер диагонали больше или равен числу столбцов/строк и является нечётным. В таком случае элементы выписываются сверху вниз.
2. Номер диагонали больше или равен числу столбцов/строк и является чётным. В таком случае элементы выписываются снизу вверх.
3. Номер диагонали меньше числа столбцов/строк и является нечётным. В таком случае элементы выписываются сверху вниз.
4. Номер диагонали меньше числа столбцов/строк и является чётным. В таком случае элементы выписываются снизу вверх.

# Текстовый алгоритм решения задачи

Таблица 1 – Алгоритм решения

|  |  |
| --- | --- |
| Номер  шага | Назначение шага |
|  | Ввод N |
|  | I:=1 |
|  | Начало цикла А1. Проверка выполнения условия (I <= N). Если условие истинно, перейти к шагу 4, иначе – к шагу 11 |
|  | J:=1 |
|  | Начало цикла А2. Проверка выполнения условия (J <= N). Если условие истинно, перейти к шагу 6, иначе – к шагу 9 |
|  | Ввод Arr1[I, J] |
|  | J:=J+1 |
|  | Конец цикла А2. Идти к шагу 5 |
|  | I:=I+1 |
|  | Конец цикла А1. Идти к шагу 3 |
|  | K:=0 |
|  | D:=2\*N-1 |
|  | Начало цикла А3. Проверка выполнения условия (D<= 1). Если условие истинно, перейти к шагу 14, иначе – к шагу 43 |
|  | Проверка выполнения условия (D>=N). Если условие истинно, перейти к шагу 15, иначе – к шагу 28 |
|  | Проверка выполнения условия odd(D). Если условие истинно, перейти к шагу 16, иначе – к шагу 22 |
|  | I:=1 |
|  | Начало цикла А4. Проверка выполнения условия (I<=2\*N-D). Если условие истинно, перейти к шагу 18, иначе – к шагу 41 |
|  | K:=K+1 |
|  | Arr2[K]:=Arr1[I, I+D-N] |
|  | I:=I+1 |
|  | Конец цикла А4. Идти к шагу 17 |
|  | I:= 2\*N-D |
|  | Начало цикла А5. Проверка выполнения условия (I>=1). Если условие истинно, перейти к шагу 24, иначе – к шагу 41 |
|  | K:=K+1 |
|  | Arr2[K]:=Arr1[I, I+D-N] |
|  | I:=I-1 |
|  | Конец цикла А5. Идти к шагу 23 |
|  | Проверка выполнения условия odd(D). Если условие истинно, перейти к шагу 29, иначе – к шагу 35 |
|  | I:=N-D+1 |

Продолжение Таблицы 1

|  |  |
| --- | --- |
| Номер шага | Назначение шага |
|  | Начало цикла А6. Проверка выполнения условия (I<=N). Если условие истинно, перейти к шагу 31, иначе – к шагу 41 |
|  | K:=K+1 |
|  | Arr2[K]:=Arr1[I, I+D-N] |
|  | I:=I+1 |
|  | Конец цикла А6. Идти к шагу 30 |
|  | I:= N |
|  | Начало цикла А7. Проверка выполнения условия (I>= N-D+1). Если условие истинно, перейти к шагу 37, иначе – к шагу 41 |
|  | K:=K+1 |
|  | Arr2[K]:=Arr1[I, I+D-N] |
|  | I:=I-1 |
|  | Конец цикла А7. Идти к шагу 36 |
|  | D:=D-1 |
|  | Конец цикла А3. Идти к шагу 13 |
|  | I:=1 |
|  | Начало цикла А8. Проверка выполнения условия (I <= K). Если условие истинно, перейти к шагу 45, иначе – к шагу 48 |
|  | Ввод Arr2[I] |
|  | I:=I+1 |
|  | Конец цикла А8. Идти к шагу 44 |
|  | Останов. |

# Структура данных

Таблица 2 – Данные

|  |  |  |
| --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение |
| Arr1 | Array[1..1000] of Integer | Двумерный массив (матрица) |
| Arr2 | Array[1.1000000] of Integer | Массив с цепочкой чисел из двумерного массива |
| N | Integer | Количество столбцов и строк матрицы |
| D | Integer | Количество и номер диагоналей в матрице |
| I, J | Integer | Параметры циклов и координаты элементов в матрице |
| K | Integer | Количество и номер элементов в цепочке |

# Схема алгоритма решения задачи по Гост 19.701-90

|  |
| --- |
|  |
| Рисунок 1 – Схема алгоритма решения задачи по ГОСТ 19.701-90 |

|  |
| --- |
|  |
| Рисунок 2 – Схема алгоритма решения задачи по ГОСТ 19.701-90 |

# Результаты расчетов

Вследствие результатов программы на экран выводятся следующие результаты расчетов:

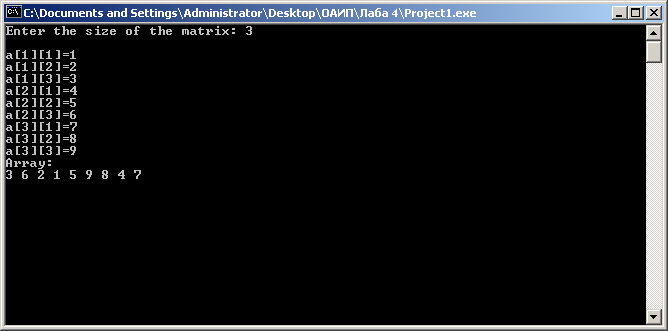


Рисунок 3 – Результаты расчетов

Приложение А

(обязательное)

Исходный код программы

Ввод данных с клавиатуры

Program Lab4\_Read;

{Find and display a chain of elements on diagonals of the matrix}

//Use app

{$APPTYPE CONSOLE}

//Declare modules

Uses

SysUtils;

//Declare vars

Var

Arr1: Array[1..1000, 1..1000] of Integer;

Arr2: Array[1..1000000] of Integer;

N, I, J, K, D, Error: Integer;

Input: String;

//Arr1 - start matrix

//Arr2 - final chain of elements

//N - length and width of the matrix

//I, J, K - parameters of loops

//D - quantity of diagonals

//Error - operator for checking input

//Input - input string

Begin

//Checking for the correct input

Repeat

Write('Enter the size of the matrix: ');

ReadLn(Input);

Val(Input, N, Error);

If (N <= 0) or (Error <> 0) then

WriteLn('Invalid input. Enter another number.');

Until (N > 0) and (Error = 0);

WriteLn;

//Enter every element of array

For I:= 1 to N do

Begin

For J:= 1 to N do

Begin

//Checking for the correct input

Repeat

Write('a[', I, ', ', J, ']=');

ReadLn(Input);

Val(Input, Arr1[I, J], Error);

If Error <> 0 then

WriteLn('Invalid input. Enter another number.');

Until Error = 0;

End;

End;

//To prevent from random false value

K:= 0;

//Every diagonal

For D:= 2 \* N - 1 downto 1 do

Begin

//Above or on the main diagonal

If D >= N then

Begin

//From top to bottom (chain)

If (D mod 2 = 1) then

For I:= 1 To 2 \* N - D do

Begin

//Increment K

K:= K + 1;

Arr2[K]:= Arr1[I, I + D - N];

End

//From bottom to top (chain)

Else

For I:= 2 \* N - D downto 1 do

Begin

//Increment K

K:= K + 1;

Arr2[K]:= Arr1[I, I + D - N];

End;

End

Else

//Under the main diagonal

Begin

//From top to bottom (chain)

If (D mod 2 = 1) then

For I:= N - D + 1 to N do

Begin

//Increment K

K:= K + 1;

Arr2[K]:= Arr1[I, I + D - N];

End

//From bottom to top (chain)

Else

For I:= N downto N - D + 1 do

Begin

//Increment K

K:= K + 1;

Arr2[K]:= Arr1[I, I + D - N];

End;

End;

End;

//Displaying the array

WriteLn('Array:');

For I:= 1 to K do

Write(Arr2[I], ' ');

ReadLn;

End.

Приложение Б

(обязательное)

Исходный код программы

Генератор равномерно-распределённых чисел

Program Lab4\_Gen;

{Find and display a chain of elements on diagonals of the matrix}

//Use app

{$APPTYPE CONSOLE}

//Declare modules

Uses

SysUtils;

//Declare vars

Var

Arr1: Array[1..1000, 1..1000] of Integer;

Arr2: Array[1..1000000] of Integer;

N, I, J, K, D, Error: Integer;

Input: String;

//Arr1 - start matrix

//Arr2 - final chain of elements

//N - length and width of the matrix

//I, J, K - parameters of loops

//D - quantity of diagonals

//Error - operator for checking input

//Input - input string

Begin

//Checking for the correct input

Repeat

Write('Enter the size of the matrix: ');

ReadLn(Input);

Val(Input, N, Error);

If (N <= 0) or (Error <> 0) then

WriteLn('Invalid input. Enter another number.');

Until (N > 0) and (Error = 0);

WriteLn;

//Generate every element of array

For I:= 1 to N do

Begin

For J:= 1 to N do

Begin

Arr1[I, J]:=Round(Random(1000)/100 - 5);

Write(Arr1[I, J]:2, ' ');

End;

WriteLn;

End;

//To prevent from random false value

K:= 0;

//Every diagonal

For D:= 2 \* N - 1 downto 1 do

Begin

//Above or on the main diagonal

If D >= N then

Begin

//From top to bottom (chain)

If (D mod 2 = 1) then

For I:= 1 To 2 \* N - D do

Begin

//Increment K

K:= K + 1;

Arr2[K]:= Arr1[I, I + D - N];

End

//From bottom to top (chain)

Else

For I:= 2 \* N - D downto 1 do

Begin

//Increment K

K:= K + 1;

Arr2[K]:= Arr1[I, I + D - N];

End;

End

Else

//Under the main diagonal

Begin

//From top to bottom (chain)

If (D mod 2 = 1) then

For I:= N - D + 1 to N do

Begin

//Increment K

K:= K + 1;

Arr2[K]:= Arr1[I, I + D - N];

End

//From bottom to top (chain)

Else

For I:= N downto N - D + 1 do

Begin

//Increment K

K:= K + 1;

Arr2[K]:= Arr1[I, I + D - N];

End;

End;

End;

//Displaying the array

WriteLn('Array:');

For I:= 1 to K do

Write(Arr2[I], ' ');

ReadLn;

End.

Приложение В

(обязательное)

Исходный код программы (по постановке 1)

Типизированная константа-массив

Program Lab4\_Const;

{Find and display a chain of elements on diagonals of the matrix}

//Use app

{$APPTYPE CONSOLE}

//Declare modules

Uses

SysUtils;

//Declare consts

Const

N = 3;

Arr1: Array[1..3, 1..3] of Integer = ((1, 2, 3), (4, 5, 6), (7, 8, 9));

//N - length and width of the matrix

//Arr1 - start matrix

//Declare vars

Var

Arr2: Array[1..9] of Integer;

I, J, K, D: Integer;

//Arr2 - final chain of elements

//I, J, K - parameters of loops

//D - quantity of diagonals

Begin

//Display every element of array

For I:= 1 to N do

Begin;

For J:=1 to N do

Write(Arr1[I, J], ' ');

WriteLn;

End;

//To prevent from random false value

K:= 0;

//Every diagonal

For D:= 2 \* N - 1 downto 1 do

Begin

//Above or on the main diagonal

If D >= N then

Begin

//From top to bottom (chain)

If (D mod 2 = 1) then

For I:= 1 To 2 \* N - D do

Begin

//Increment K

K:= K + 1;

Arr2[K]:= Arr1[I, I + D - N];

End

//From bottom to top (chain)

Else

For I:= 2 \* N - D downto 1 do

Begin

//Increment K

K:= K + 1;

Arr2[K]:= Arr1[I, I + D - N];

End;

End

Else

//Under the main diagonal

Begin

//From top to bottom (chain)

If (D mod 2 = 1) then

For I:= N - D + 1 to N do

Begin

//Increment K

K:= K + 1;

Arr2[K]:= Arr1[I, I + D - N];

End

//From bottom to top (chain)

Else

For I:= N downto N - D + 1 do

Begin

//Increment K

K:= K + 1;

Arr2[K]:= Arr1[I, I + D - N];

End;

End;

End;

//Displaying the array

WriteLn('Array:');

For I:= 1 to K do

Write(Arr2[I], ' ');

ReadLn;

End.

Приложение Г

(обязательное)

Исходный код программы (по постановке 1)

Типизированная переменная-массив

Program Lab4\_Var;

{Find and display a chain of elements on diagonals of the matrix}

//Use app

{$APPTYPE CONSOLE}

//Declare modules

Uses

SysUtils;

//Declare consts

Const

N = 3;

//N - length and width of the matrix

//Declare vars

Var

Arr1: Array[1..3, 1..3] of Integer = ((1, 2, 3), (4, 5, 6), (7, 8, 9));

Arr2: Array[1..9] of Integer;

I, J, K, D: Integer;

//Arr1 - start matrix

//Arr2 - final chain of elements

//I, J, K - parameters of loops

//D - quantity of diagonals

Begin

//Display every element of array

For I:= 1 to N do

Begin;

For J:=1 to N do

Write(Arr1[I, J], ' ');

WriteLn;

End;

//To prevent from random false value

K:= 0;

//Every diagonal

For D:= 2 \* N - 1 downto 1 do

Begin

//Above or on the main diagonal

If D >= N then

Begin

//From top to bottom (chain)

If (D mod 2 = 1) then

For I:= 1 To 2 \* N - D do

Begin

//Increment K

K:= K + 1;

Arr2[K]:= Arr1[I, I + D - N];

End

//From bottom to top (chain)

Else

For I:= 2 \* N - D downto 1 do

Begin

//Increment K

K:= K + 1;

Arr2[K]:= Arr1[I, I + D - N];

End;

End

Else

//Under the main diagonal

Begin

//From top to bottom (chain)

If (D mod 2 = 1) then

For I:= N - D + 1 to N do

Begin

//Increment K

K:= K + 1;

Arr2[K]:= Arr1[I, I + D - N];

End

//From bottom to top (chain)

Else

For I:= N downto N - D + 1 do

Begin

//Increment K

K:= K + 1;

Arr2[K]:= Arr1[I, I + D - N];

End;

End;

End;

//Displaying the array

WriteLn('Array:');

For I:= 1 to K do

Write(Arr2[I], ' ');

ReadLn;

End.

Приложение Д

(обязательное)

Тестовые наборы

Тестовая ситуация 1: проверка ввода данных

Тестовая ситуация: проверка, сообщит ли программа о некорректном вводе, если введено буквенное, отрицательное или дробное значение.

Исходные данные: N = абоба.

Ожидаемый результат: Invalid input. Enter another number.

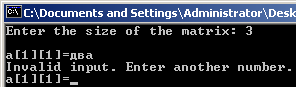
Полученный результат:



Исходные данные: Arr1[1, 1] = два.

Ожидаемый результат: Invalid input. Enter another number.

Полученный результат:



Исходные данные: N = 1.2.

Ожидаемый результат: Invalid input. Enter another number.

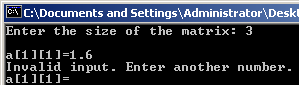
Полученный результат:



Исходные данные: Arr1[1, 1] = 1.6.

Ожидаемый результат: Invalid input. Enter another number.

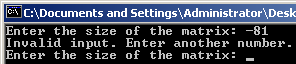
Полученный результат:



Исходные данные: N = -81.

Ожидаемый результат: Invalid input. Enter another number.

Полученный результат:



Тестовая ситуация 2: проверка вычислений

Тестовая ситуация для проверки вывода правильной цепочки значений.

Исходные данные: квадратная матрица 3х3

Ожидаемый результат: Arr2 = (3 6 2 1 5 9 8 4 7).

Полученный результат:

