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 №3

Тема работы:

Работа с массивами

Выполнил

студент: гр. 151004 Башлыков В.В.

Проверил: Фадеева Е. П.

Минск 2021

Содержание

[1 Постановка задачи 3](#_Toc89786689)

[2 Ход решения 4](#_Toc89786690)

[3 Текстовый алгоритм решения задачи 5](#_Toc89786691)

[4 Структура данных 6](#_Toc89786692)

[5 Схема алгоритма решения задачи по ГОСТ 19.701-90 7](#_Toc89786693)

[6 Результаты расчетов 9](#_Toc89786694)

[Приложение А 11](#_Toc89786695)

[Приложение Б 14](#_Toc89786696)

[Приложение В 18](#_Toc89786697)

[Приложение Г 21](#_Toc89786698)

[Приложение Д 24](#_Toc89786699)

# Постановка задачи

В массиве S размерности N даны элементы первого столбца и первой строки. Вычислить все элементы массива так, чтобы каждый элемент, находящийся на диагонали D, был бы равен первому элементу диагонали D. (Рассматривать диагонали, параллельные главной диагонали).

Правила постановки:

1. По условию задачи дввумерный массив S формируется из элементов первой строки и первого столюца
2. Программа учитывает некорректный ввод пользователя:

* не числовые значения;
* переполнения значения Integer;
* не целые числа.

1. Программа предусматривает потенциальное переполнение переменной типа Integer при вычислениях.
2. Нельзя сортировать массив S.
3. Для решения задачи используется 4 постоновки:

* ввод данных в массив выполняет пользователь с клавиатуры(№1);
* ввод данных в массив с использованием генератора чисел(№2);
* ввод данных в массив с использованием типизированной константы-массив(№3);
* ввод данных в массив с использованием типизированной переменной-массив(№4).

# Ход решения

1. Так как даны элементы первой строки и первого столбца, необходимо каждому элементу, на каждой диагонали, паралельной главной диагонали, присвоить значение первого элемента данной диагонали.
2. Проведя анализ матриц размером N x N сделан вывод что они всегда будут квадратными. Так как матрица квадратная, элементы, параллельные главной диагонали, будут иметь следующую зависимость:

Sij = Si+1 j+1 = Si+2 j+2 = … = Si+N-1 j+N-1.

1. Анализируя данную зависипость понятно, что легко ошибиться и выйти за границы массива при переборе элементов по диагоналям, поэтому логично перебирать элементы по строкам, результат не изменится, но выход за границы легче избежать. В результате получаем зависимость:

Si+1 j+1 = Si-1 j-1.

# Текстовый алгоритм решения задачи

Таблица 1 – Алгоритм решения

|  |  |
| --- | --- |
| Номер  шага | Назначение шага |
|  | N := 10 |
|  | Ввод S [1..N,1], S[1,2..N] |
|  | I := 2 |
|  | Начало цикла А1. Проверка выполнения условия (I <= N). Если условие истинно, идти к шагу 5, иначе к шагу 12 |
|  | J := 2 |
|  | Начало цикла A2. Проверка выполнения условия (J <= N). Если условие истинно, идти к шагу 7, иначе идти к шагу 10 |
|  | S[i,j] := S[i - 1,j - 1] |
|  | J := J + 1 |
|  | Конец цикла A2. Идти к шагу 6 |
|  | I := I + 1 |
|  | Конец цикла A1. Идти к шагу 4 |
|  | Вывод S[1..N,1..N] |
| 13 | Останов. | |

# Структура данных

Таблица 2 – Данные

|  |  |  |
| --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение |
| N | Const | Размер массива S |
| S | Array [1..N,1..N] of  Integer | Массив с заданными начальными значениями, во время выполнения программы принимает все следующие значения |
| I, J | Integer | Переменные, используемые как счетчики циклов |

# Схема алгоритма решения задачи по ГОСТ 19.701-90

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

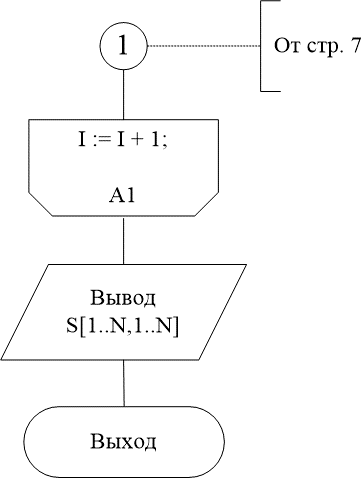


Рисунок 2 – Схема алгоритма решения задачи по ГОСТ 19.701-90

# Результаты расчетов

Вследствие выполнения программы на экран выводятся следующие результаты:

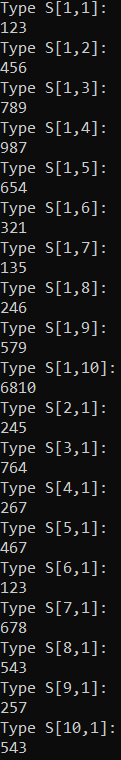


Рисунок 3 – Введённые данные

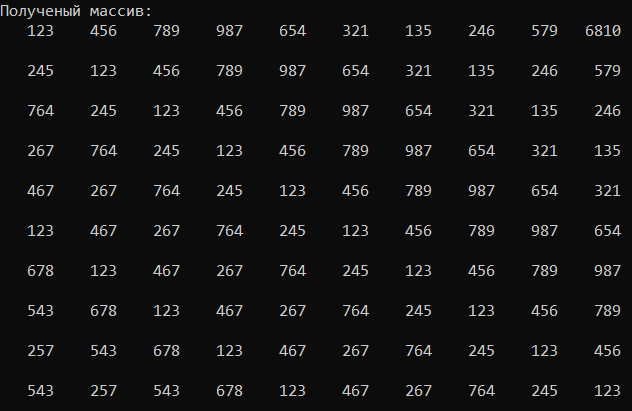


Рисунок 4 – Полученный результат

Приложение А

(обязательное)

Исходный код программы (по постоновке 1)

Массив заполняется путём ввода данных с клавиатуры

Program LBR\_4.1;

{In an array S of dimension N x N, the elements

of the first column and the first row are given.

Calculate all the elements of the array so that

each element located on the diagonal D would be

equal to the first element of the diagonal D.

(Consider diagonals parallel to the main diagonal)}

//Const values

Const

N = 10;

//N - number of array elements

//Type data

Type

MyArray = array[1..N,1..N] of Integer;

//Var values

Var

I, J: Integer;

S: MyArray;

Value: String;

Error: Integer;

//I, J - cycle counters

//S - two-dimensional array of dimension NxN

//Value – variable for checking input

//Error - variable for checking input

begin

//Start of the cycle A1

for I := 1 to N do

begin

//Start of the cycle A2

for J := 1 to N do

begin

//Checking J = 0 or I = 0

if (J = 1) or (I = 1) then

begin

//input coentrol cycle

Repeat

WriteLn('Type S[', I, ',', J, ']: ');

ReadLn(Value);

val(Value, S[I,J], Error);

If (Error <> 0) then

Writeln('Incorrect value');

Until (Error = 0);

end;

//The end of the cycle A2

end;

J := 0;

//The end of the cycle A1

end;

//filling in array elements parallel to the main

//diagonal and start of the cycle B1

for I := 2 to N do

Begin

//Start of the cycle B2

for J := 2 to N do

Begin

//Аilling in array elements over the main //diagonal

S[I,J] := S[I - 1,J - 1];

//The end of the cycle B2

End;

//The end of the cycle B1

End;

//output of the final array to the console

I := 1;

J := 1;

WriteLn('Полученый массив: ');

//start of the C1 cycle that outputs the results

//to the console

while (I <= N) do

begin

//Start of the cycle C2

while (J <= N) do

Begin

//Output of array elements

Write(S[I,J]:6, ' ');

J := J + 1;

//The end of the cycle С2

End;

J := 1;

I := I + 1;

//Output of array elements

WriteLn(' ' + #10);

//The end of the cycle С1

end;

ReadLn;

end.

Приложение Б

(обязательное)

Тестовые наборы

Тестовая ситуация 1 для проверки ввода данных

Тест 1

Тестовая ситуация для проверки, сообщит ли программа о некорректном вводе, если введен элемент,не являющийся числом.

Исходные данные: S[1,1]: 1

S[1,2]: s

Ожидаемый результат: Incorrect value.

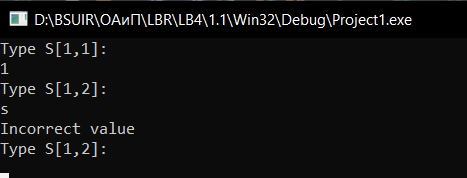
Полученный результат:

Рисунок 5 – Полученный результат, тест 1

Тест 2

Тестовая ситуация для проверки, сообщит ли программа о некорректном вводе, если введен элемент массива больше допустимого значения Integer.

Исходные данные: A[1,1]: 99999999999

Ожидаемый результат: Incorrect value.

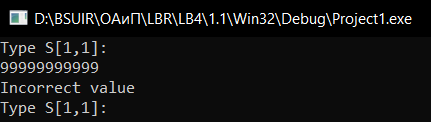
Полученный результат:

Рисунок 6 – Полученный результат, тест 2

Тест 3

Тестовая ситуация для проверки, сообщит ли программа о некорректном вводе, если введен не целочисленный элемент массива.

Исходные данные: A[1,1]: 1.1

Ожидаемый результат: Incorrect value.

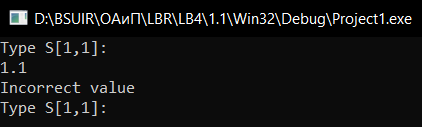
Полученный результат:

Рисунок 7 – Полученный результат, тест 3

Приложение В

(обязательное)

Исходный код программы (по постоновке 2)

Массив заполняется с использованием генератора чисел

program LBR\_4.2;

{In an array S of dimension N x N, the elements

of the first column and the first row are given.

Calculate all the elements of the array so that

each element located on the diagonal D would be

equal to the first element of the diagonal D.

(Consider diagonals parallel to the main diagonal)}

{$APPTYPE CONSOLE}

{$R \*.res}

uses

System.SysUtils;

//Const values

Const

N = 10;

//N - number of array elements

//Var values

var

i, j: Integer;

S: array [1..N,1..N] of Integer;

{i, j - cycle counters

S - two-dimensional array of dimension NxN }

begin

{filling the first row and the first column

with random elements and start of the cycle A1 }

for i := 1 to N do

begin

//Start of the cycle A2

for j := 1 to N do

begin

//Checking j = 0 or i = 0

if (j = 1) or (i = 1) then

begin

S[i,j] := Random(9999);

end;

//The end of the cycle A2

end;

j := 0;

//The end of the cycle A1

end;

{filling in array elements parallel to the main

diagonal and start of the cycle B1 }

for i := 2 to N do

Begin

//Аilling in array elements under the main diagonal

S[i,j] := S[i - 1,j - 1];

//Start of the cycle B2

for j := 2 to N do

Begin

//Аilling in array elements over the main //diagonal

S[i,j] := S[i - 1,j - 1];

//The end of the cycle B2

End;

//The end of the cycle B1

End;

{output of the final array to the console}

i := 1;

j := 1;

WriteLn('Полученый массив: ');

{start of the C1 cycle that outputs the results

to the console }

while (i <= N) do

begin

//Start of the cycle C2

while (j <= N) do

Begin

//Output of array elements

Write(S[i,j]:6, ' ');

j := j + 1;

//The end of the cycle С2

End;

j := 1;

i := i + 1;

//Output of array elements

WriteLn(' ' + #10);

//The end of the cycle С1

end;

ReadLn;

end.

Приложение Г

(обязательное)

Исходный код программы (по постоновке 3)

Массив заполняется с использованием типизированной константы-массив

program LBR\_4.3;

{In an array S of dimension N x N, the elements

of the first column and the first row are given.

Calculate all the elements of the array so that

each element located on the diagonal D would be

equal to the first element of the diagonal D.

(Consider diagonals parallel to the main diagonal)}

{$APPTYPE CONSOLE}

{$R \*.res}

uses

System.SysUtils;

//Const values

Const

N = 10;

S: array[1..N,1..N] of Integer =

((187,234,325,477,534,621,709,845,911,102),

(276,0,0,0,0,0,0,0,0,0), (309,0,0,0,0,0,0,0,0,0), (450,0,0,0,0,0,0,0,0,0), (589,0,0,0,0,0,0,0,0,0), (645,0,0,0,0,0,0,0,0,0), (765,0,0,0,0,0,0,0,0,0), (843,0,0,0,0,0,0,0,0,0), (975,0,0,0,0,0,0,0,0,0), (106,0,0,0,0,0,0,0,0,0));

//N - number of array elements

//S - Const Array [1..N,1..N] of Integer

//Var values

var

A: array [1..N,1..N] of Integer;

i, j: Integer;

{A - Array [1..N,1..N] of Integer

i, j - cycle counters

S - two-dimensional array of dimension NxN }

begin

//Start of the cycle A1

for i := 1 to N do

begin

//Start of the cycle A2

for j := 1 to N do

begin

//Checking j = 0 or i = 0

if (j = 1) or (i = 1) then

begin

A[i,j] := S[i,j];

end;

//End of the sycle A2

end;

j := 0;

//End of the cycle A1

end;

{filling in array elements parallel to the main

diagonal and start of the cycle B1 }

for i := 2 to N do

Begin

//Start of the cycle B2

for j := 2 to N do

Begin

//Аilling in array elements over the main diagonal

A[i,j] := A[i - 1,j - 1];

//The end of the cycle B2

End;

//The end of the cycle B1

End;

{output of the final array to the console}

i := 1;

j := 1;

WriteLn('Полученый массив: ');

{start of the C1 cycle that outputs the results

to the console }

while (i <= N) do

begin

//Start of the cycle C2

while (j <= N) do

Begin

//Output of array elements

Write(A[i,j]:6, ' ');

j := j + 1;

//The end of the cycle С2

End;

j := 1;

i := i + 1;

//Output of array elements

WriteLn(' ' + #10);

//The end of the cycle С1

end;

ReadLn;

end.

Приложение Д

(обязательное)

Исходный код программы (по постоновке 4)

Массив заполняется с использованием типизированной переменной-массив

program LBR\_4.4;

{In an array S of dimension N x N, the elements

of the first column and the first row are given.

Calculate all the elements of the array so that

each element located on the diagonal D would be

equal to the first element of the diagonal D.

(Consider diagonals parallel to the main diagonal)}

{$APPTYPE CONSOLE}

{$R \*.res}

uses

System.SysUtils;

//Const values

Const

N = 10;

//N - number of array elements

//Var values

var

S: array[1..N,1..N] of Integer =

((187,234,325,477,534,621,709,845,911,102), (276,0,0,0,0,0,0,0,0,0),

(309,0,0,0,0,0,0,0,0,0),

(450,0,0,0,0,0,0,0,0,0),

(589,0,0,0,0,0,0,0,0,0),

(645,0,0,0,0,0,0,0,0,0),

(765,0,0,0,0,0,0,0,0,0),

(843,0,0,0,0,0,0,0,0,0),

(975,0,0,0,0,0,0,0,0,0),

(106,0,0,0,0,0,0,0,0,0));

//S - two-dimensional array of dimension NxN

//Var values

var

i, j: Integer;

{i, j - cycle counter}

begin

{filling in array elements parallel to the main

diagonal and start of the cycle B1 }

for i := 2 to N do

Begin

//Start of the cycle B2

for j := 2 to N do

Begin

//Аilling in array elements over the main diagonal

S[i,j] := S[i - 1,j - 1];

//The end of the cycle B2

End;

//The end of the cycle B1

End;

{output of the final array to the console}

i := 1;

j := 1;

WriteLn('Полученый массив: ');

{start of the C1 cycle that outputs the results

to the console }

while (i <= N) do

begin

//Start of the cycle C2

while (j <= N) do

Begin

//Output of array elements

Write(S[i,j]:6, ' ');

j := j + 1;

//The end of the cycle С2

End;

j := 1;

i := i + 1;

//Output of array elements

WriteLn(' ' + #10);

//The end of the cycle С1

end;

ReadLn;

end.