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

студент: гр. 151003 Барановский Р.А.

Проверил: Фадеева Е.П.

Минск 2021

содержание

[1 Постановка задачи 3](#_Toc86685122)

[2 Текстовый алгоритм решения задачи 4](#_Toc86685123)

[3 Структура данных 6](#_Toc86685124)

[4 Схема алгоритма решения задачи по Гост 19.701-90 7](#_Toc86685125)

[5 Результаты расчетов 8](#_Toc86685126)

[Приложение А1 12](#_Toc86685127)

[Приложение А2 15](#_Toc86685128)

[Приложение А3 17](#_Toc86685129)

[Приложение А4 19](#_Toc86685130)

[Приложение Б 21](#_Toc86685131)

# Постановка задачи

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

# Текстовый алгоритм решения задачи

Таблица – Алгоритм решения

|  |  |
| --- | --- |
| Номер  шага | Назначение шага |
| 1 | Const N = 10 |
| 2 | Error := 1 |
| 3 | i := 1 |
| 4 | j := 1 |
| 5 | Начало цикла А1. Проверка выполнения условия j <= N. Если условие истинно, то идти к шагу 6, иначе к шагу 16 |
| 6 | Начало цикла А2. Идти к шагу 7 |
| 7 | Ввод YStr[i,j] |
| 8 | Error := 1 |
| 9 | Проверка выполнения условия YStr[i,j] – вещественное число. Если условие истинно, то идти шагу 10, иначе к шагу 12 |
| 10 | Y[i,j] := YStr[i,j] |
| 11 | Error := 0 |
| 12 | Проверка выполнения условия Error < > 0. Если условие истинно, то идти к шагу 13, иначе к шагу 14 |
| 13 | Вывод ‘Пожалуйста, введите вещественное число’ |
| 14 | Конец цикла А2. Проверка выполнения условия Error = 0. Если условие истинно, то идти к шагу 15, иначе к шагу 6. |
| 15 | j := j + 1 |
| 16 | Конец цикла А1. Идти к шагу 5 |
| 17 | j := 1 |
| 18 | i := 2 |
| 19 | Начало цикла А3. Проверка выполнения условия i <= N. Если условие истинно, то идти к шагу 20, иначе к шагу 30 |
| 20 | Начало цикла А4. Идти к шагу 21 |
| 21 | Ввод YStr[i,j] |
| 22 | Error := 1 |
| 23 | Проверка выполнения условия YStr[i,j] – вещественное число. Если условие истинно, то идти шагу 24, иначе к шагу 26 |
| 24 | Y[i,j] := YStr[i,j] |
| 25 | Error := 0 |
| 26 | Проверка выполнения условия Error < > 0. Если условие истинно, то идти к шагу 27, иначе к шагу 28 |
| 27 | Вывод ‘Пожалуйста, введите вещественное число’ |
| 28 | Конец цикла А4. Проверка выполнения условия Error = 0. Если условие истинно, то идти к шагу 29, иначе к шагу 20. |
| 29 | i := i + 1 |
| 30 | Конец цикла А3. Идти к шагу 19 |
| 31 | i := 1 |
| 32 | Начало цикла А5. Проверка выполнения условия i <= N. Если условие истинно, то идти к шагу 33, иначе к шагу 39 |
| 33 | j := 1 |
| 34 | Начало цикла А6. Проверка выполнения условия j <= N. Если условие истинно, то идти к шагу 35, иначе к шагу 38 |
| 35 | Вывод Y[i,j] |
| 36 | j := j + 1 |
| 37 | Конец цикла А6. Идти к шагу 34 |
| 38 | Конец цикла А5. Идти к шагу 32 |
| 39 | i := 2 |
| 40 | Начало цикла А7. Проверка выполнения условия i <= N. Если условие истинно, то идти к шагу 41, иначе к шагу 48 |
| 41 | j := 2 |
| 42 | Начало цикла А8. Проверка выполнения условия j <= N. Если условие истинно, то идти к шагу 43, иначе к шагу 46 |
| 43 | Y[i,j] := Y[i-1, j-1] |
| 44 | j := j + 1 |
| 45 | Конец цикла А8. Идти к шагу 42. |
| 46 | i := i + 1 |
| 47 | Конец цикла А7. Идти к шагу 40 |
| 48 | i := 1 |
| 49 | Начало цикла А9. Проверка выполнения условия i <= N. Если условие истинно, то идти к шагу 50, иначе к шагу 56 |
| 50 | j := 1 |
| 51 | Начало цикла А10. Проверка выполнения условия j <= N. Если условие истинно, то идти к шагу 52, иначе к шагу 55 |
| 52 | Вывод Y[i,j] |
| 53 | j := j + 1 |
| 54 | Конец цикла А10. Идти к шагу 34 |
| 55 | Конец цикла А9. Идти к шагу 32 |
| 56 | Останов. |

# Структура данных

Таблица - Данные

|  |  |  |
| --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение |
| N | Const | Размер массива |
| Y | Array of real | Массив |
| YStr | Array of String | Проверка ошибок ввода |
| i | Integer | Счетчик для строк |
| j | Integer | Счетчик для колонок |
| Error | Integer | Переменная, принимающая значение 1, если пользователь допустил ошибку при вводе. 0, если ошибок допущено не было |

# Схема алгоритма решения задачи по Гост 19.701-90

Рисунок - Схема алгоритма решения задачи по ГОСТ 19.701-90

# Результаты расчетов

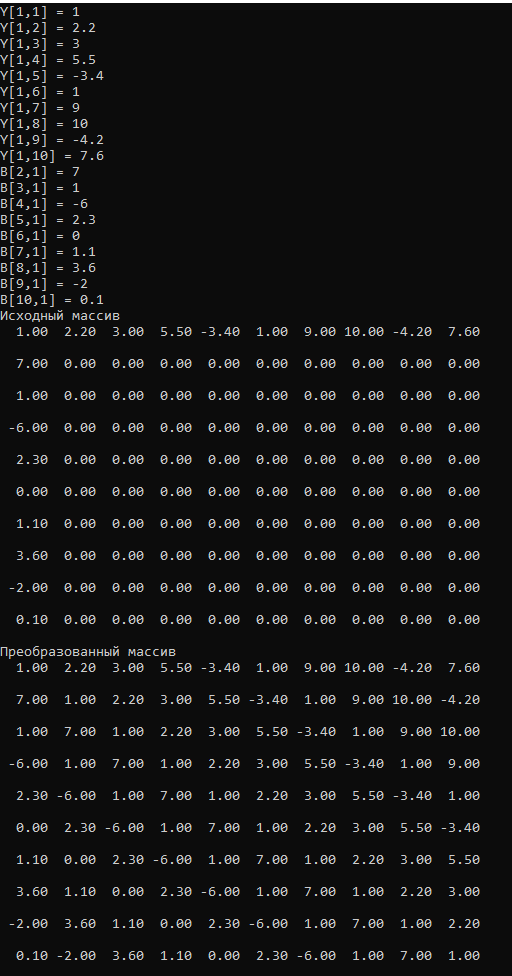
Вследствие результатов программы на экран выводятся следующие результаты расчетов:

Рисунок 2 – Результаты расчетов 1 программы

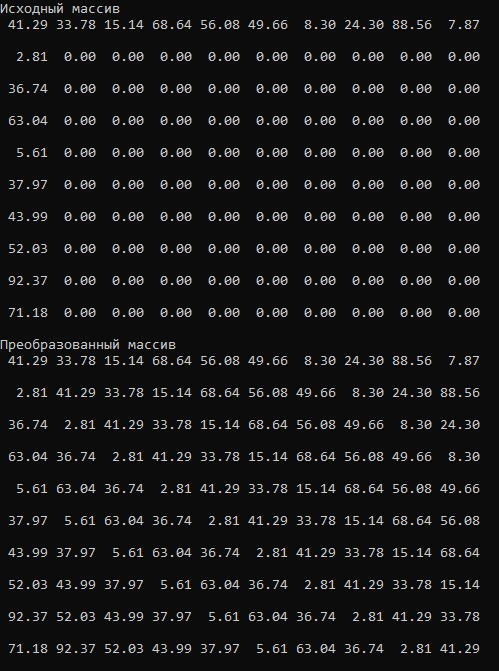


Рисунок 3 - Результаты расчетов 2 программы

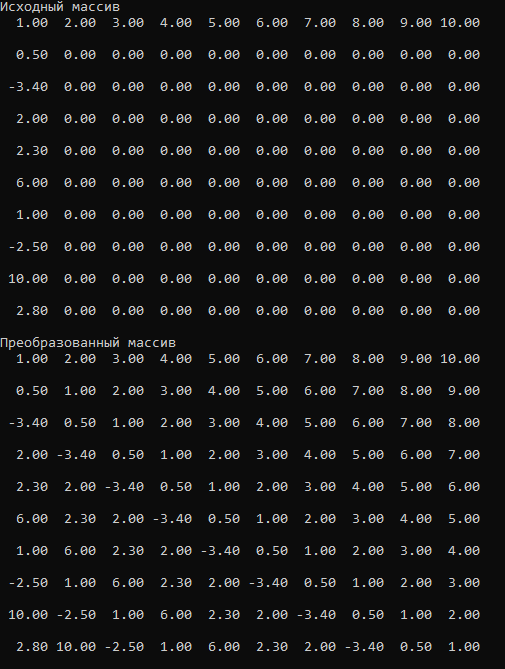


Рисунок 4 – Результаты расчетов 3 программы

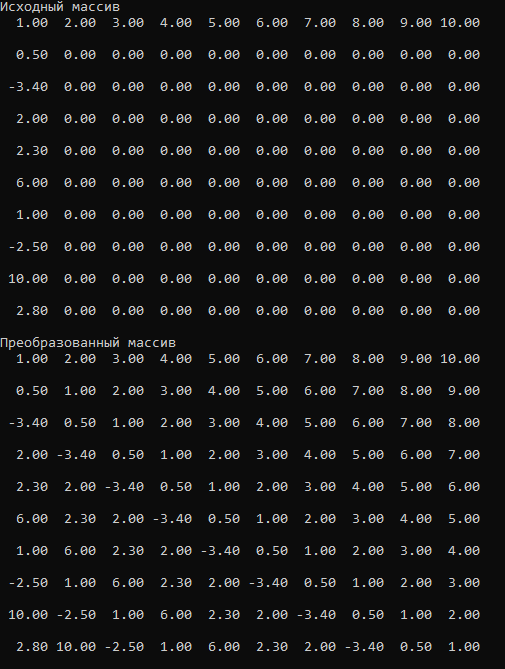


Рисунок 5 – Результаты расчетов 4 программы

Приложение А1

(обязательное)

Исходный код программы

program Lab4Exp1;

//Program supplement the array

//Console app

{$APPTYPE CONSOLE}

{$R \*.res}

//Modules declaration

uses

System.SysUtils;

//Constants declaration

Const

N = 10;

//Arrays

Type

Mas = array[1..N,1..N] of real;

MasStr = array[1..N,1..N] of String;

//Variables declaration

Var

i, j, Error: integer;

Y: Mas;

YStr: MasStr;

{ Mas - array

MasStr - mistakes checker

i - counter for raws

j - counter for columns

Error - variable which is 1 if

user made a mistake and 0 if not }

//Start the program

Begin

//Set Error to the original value

Error := 1;

//Set i to the original value

i := 1;

//Inputting array elements

for j := 1 to N do

begin

//Protection from mistakes

repeat

begin

write('Y[',i,',',j,'] = ');

readln(YStr[i,j]);

val(YStr[i,j], Y[i,j], Error);

//If mistake happened warn about it

if Error <> 0 then

writeln('Пожалуйста, введите вещественное число');

end;

//If error happened repeat the cycle

until

Error = 0;

end;

//Set j to the original value

j := 1;

//Inputting array elements

for i := 2 to N do

begin

//Protection from mistakes

repeat

begin

write('B[',i,',',j,'] = ');

readln(YStr[i,j]);

val(YStr[i,j], Y[i,j], Error);

//If mistake happened warn about it

if Error <> 0 then

writeln('Пожалуйста, введите вещественное число');

end;

//If error happened repeat the cycle

until

Error = 0;

end;

//Output original array

writeln('Исходный массив');

for i := 1 to N do

begin

for j := 1 to N do

write(Y[i,j]:6:2);

writeln;

writeln;

end;

//Supplement the array

for i:=2 to n do

for j:=2 to n do

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

//Output supplemented array

writeln('Преобразованный массив');

for i := 1 to N do

begin

for j := 1 to N do

write(Y[i,j]:6:2);

writeln;

writeln;

end;

//Final operations

readln;

End.

Приложение А2

(обязательное)

Исходный код программы

program Lab4Exp2;

//Program supplement the array

//Console app

{$APPTYPE CONSOLE}

{$R \*.res}

//Modules declaration

uses

System.SysUtils;

//Constants declaration

Const

N = 10;

//Arrays

Type

Mas = array[1..N,1..N] of real;

//Variables declaration

Var

i, j: integer;

Y: Mas;

{ Mas - array

i - counter for raws

j - counter for columns }

//Start the program

Begin

//Creating first raw

Randomize;

i := 1;

for j := 1 to N do

Y[i,j] := random(10000)/100;

//Creating first column

j := 1;

for i := 2 to N do

Y[i,j] := random(10000)/100;

//Output original array

writeln('Исходный массив');

for i := 1 to N do

begin

for j := 1 to N do

write(Y[i,j]:6:2);

writeln;

writeln;

end;

//Supplement the array

for i:=2 to N do

for j:=2 to N do

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

//Output supplemented array

writeln('Преобразованный массив');

for i := 1 to N do

begin

for j := 1 to N do

write(Y[i,j]:6:2);

writeln;

writeln;

end;

//Fianl operations

readln;

End.

Приложение А3

(обязательное)

Исходный код программы

program Lab4Exp3;

//Program supplement the array

//Console app

{$APPTYPE CONSOLE}

{$R \*.res}

//Modules declaration

uses

System.SysUtils;

//Constants declaration

Const

N = 10;

MasRaw: array[1..N] of real = (1,2,3,4,5,6,7,8,9,10);

MasCol: array[1..N] of real = (1,0.5,-3.4,2,2.3,6,1,

- 2.5,10,2.8);

//Variables declaration

Var

Y: array[1..N,1..N] of real;

i, j: integer;

{ MasRaw - 1-st raw elements

MasCol - 1-st column elements

Y - array

i - counter for raws

j - counter for columns }

//Start the program

Begin

//Filling the 1-st raw

i := 1;

for j := 1 to N do

Y[i,j] := MasRaw[j];

//Filling the 1-st column

j := 1;

for i := 1 to N do

Y[i,j] := MasCol[i];

//Output original array

writeln('Исходный массив');

for i := 1 to N do

begin

for j := 1 to N do

write(Y[i,j]:6:2);

writeln;

writeln;

end;

//Supplement the array

for i:=2 to N do

for j:=2 to N do

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

//Output supplemented array

writeln('Преобразованный массив');

for i := 1 to N do

begin

for j := 1 to N do

write(Y[i,j]:6:2);

writeln;

writeln;

end;

//Final operations

readln;

End.

Приложение А4

(обязательное)

Исходный код программы

program Lab4Exp4;

//Program supplement the array

//Console app

{$APPTYPE CONSOLE}

{$R \*.res}

//Modules declaration

uses

System.SysUtils;

//Constants declaration

Const

N = 10;

//Variables declaration

Var

i, j: integer;

Y: array[1..N,1..N] of real = ((1,2,3,4,5,6,7,8,9,10),

(0.5,0,0,0,0,0,0,0,0,0),

(-3.4,0,0,0,0,0,0,0,0,0),

(2,0,0,0,0,0,0,0,0,0),

(2.3,0,0,0,0,0,0,0,0,0),

(6,0,0,0,0,0,0,0,0,0),

(1,0,0,0,0,0,0,0,0,0),

(-2.5,0,0,0,0,0,0,0,0,0),

(10,0,0,0,0,0,0,0,0,0),

(2.8,0,0,0,0,0,0,0,0,0));

{ Y - array

i - counter for raws

j - counter for columns }

//Start the program

Begin

//Output original array

writeln('Исходный массив');

for i := 1 to N do

begin

for j := 1 to N do

write(Y[i,j]:6:2);

writeln;

writeln;

end;

//Supplement the array

for i:=2 to N do

for j:=2 to N do

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

//Output supplemented array

writeln('Преобразованный массив');

for i := 1 to N do

begin

for j := 1 to N do

write(Y[i,j]:6:2);

writeln;

writeln;

end;

//Final operations

readln;

End.

Приложение Б

(обязательное)

Тестовые наборы

Тест 1

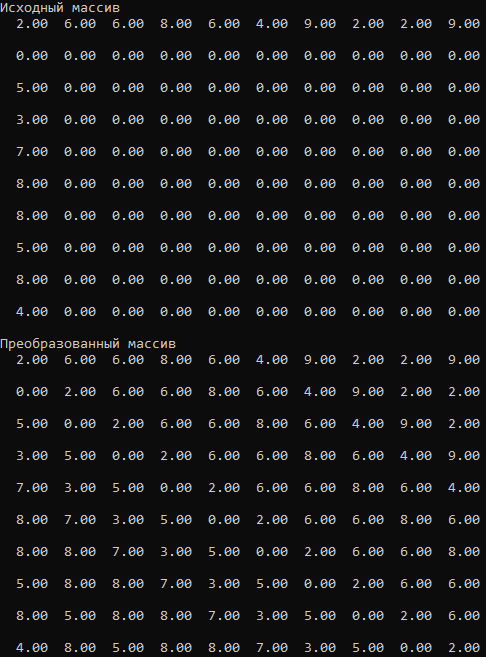
Тестовая ситуация: Y[i,j] := Random(10)

Исходные данные: Y[i,j] := Random(10)

Ожидаемый результат:

Каждый элемент на диагонали, параллельной главной, стал равен первому элементу диагонали

Полученный результат:



Тест 2

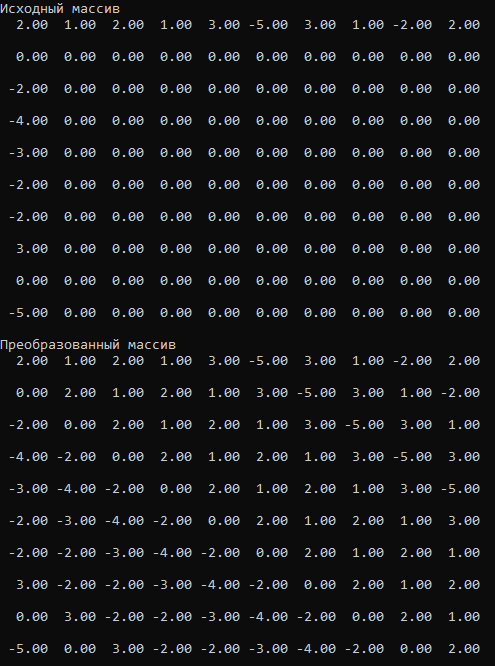
Тестовая ситуация: Y[i,j] := Random(10) - 5

Исходные данные: Y[i,j] := Random(10) - 5

Ожидаемый результат:

Каждый элемент на диагонали, параллельной главной, стал равен первому элементу диагонали

Полученный результат:



Тест 3

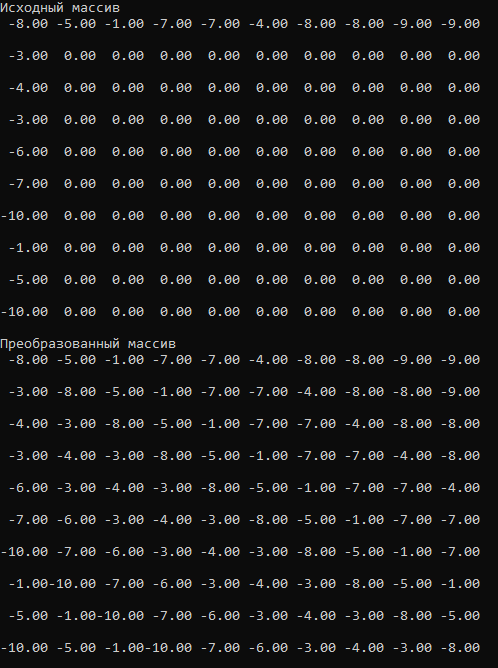
Тестовая ситуация: Y[i,j] := Random(10) - 10

Исходные данные: Y[i,j] := Random(10) - 10

Ожидаемый результат:

Каждый элемент на диагонали, параллельной главной, стал равен первому элементу диагонали

Полученный результат:



Тест 4

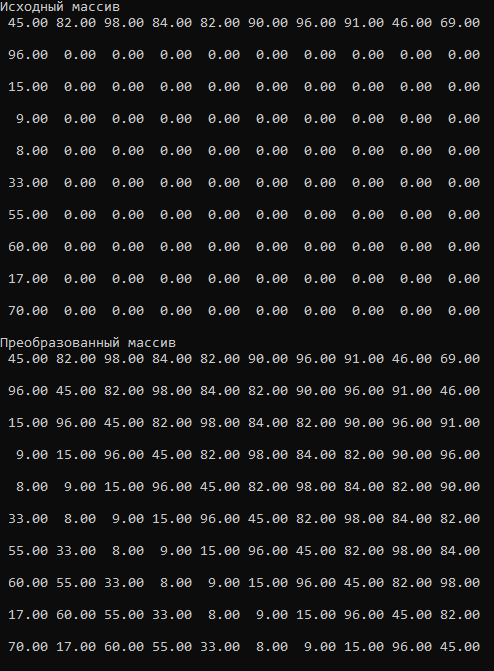
Тестовая ситуация: Y[i,j] := Random(100)

Исходные данные: Y[i,j] := Random(100)

Ожидаемый результат:

Каждый элемент на диагонали, параллельной главной, стал равен первому элементу диагонали

Полученный результат:



Тест 5

Тестовая ситуация: Проверка ввода букв в поле для чисел

Исходные данные: YStr[i,j] := ‘hooray’

Ожидаемый результат:

‘Пожалуйста, введите вещественное число’

Полученный результат: