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 № 8

Тема работы: Сортировка многомерного массива

Выполнил

студент: гр. 151003 Барановский Р.А.

Проверил: Фадеева Е.П.

Минск 2021

содержание

[1 Постановка задачи 3](#_Toc90476900)

[2 Текстовый алгоритм решения задачи 4](#_Toc90476901)

[3 Структура данных 7](#_Toc90476902)

[4 Схема алгоритма решения задачи по ГОСТ 19.701-90 8](#_Toc90476903)

[5 Результаты расчетов 12](#_Toc90476904)

[Приложение А 13](#_Toc90476905)

[Приложение Б 16](#_Toc90476906)

# Постановка задачи

 Дана матрица X[8,9]. Упорядочить элементы строк матрицы по неубыванию, а сами строки по убыванию максимальных элементов строк.

# Текстовый алгоритм решения задачи

Таблица – Алгоритм решения

|  |  |
| --- | --- |
| Номер  шага | Назначение шага |
| 1 | Const N = 8 |
| 2 | Const M = 9 |
| 3 | i := 1 |
| 4 | Начало цикла А1. Проверка выполнения условия i <= N. Если условие истинно, то идти к шагу 4, иначе к шагу 12 |
| 5 | j := 1 |
| 6 | Начало цикла А2. Проверка выполнения условия j <= M. Если условие истинно, то идти к шагу 6, иначе к шагу 10 |
| 7 | X[i,j] := random(100) + 1 |
| 8 | j := j + 1 |
| 9 | Конец цикла А2. Идти к шагу 6 |
| 10 | i := i + 1 |
| 11 | Конец цикла А1. Идти к шагу 4 |
| 12 | i := 1 |
| 13 | Начало цикла А3. Проверка выполнения условия i <= N. Если условие истинно, то идти к шагу 14, иначе к шагу 21 |
| 14 | j := 1 |
| 15 | Начало цикла А4. Проверка выполнения условия j <= M. Если условие истинно, то идти к шагу 16, иначе к шагу 19 |
| 16 | Вывод X[i,j] |
| 17 | j := j + 1 |
| 18 | Конец цикла А4. Идти к шагу 15 |
| 19 | i := i + 1 |
| 20 | Конец цикла А3. Идти к шагу 13 |
| 21 | i := 1 |
| 22 | Начало цикла А5. Проверка выполнения условия i <= N. Если условие истинно, то идти к шагу 23, иначе к шагу 37 |
| 23 | j := 1 |
| 24 | Начало цикла А6. Проверка выполнения условия j <= M. Если условие истинно, то идти к шагу 25, иначе к шагу 35 |
| 25 | k := 1 |
| 26 | Начало цикла А7. Проверка выполнения условия k <= M – j. Если условие истинно, то идти к шагу 27, иначе к шагу 33 |

Продолжение таблицы 1

|  |  |
| --- | --- |
| 27 | Проверка выполнения условия X[i,k] > X[i,k+1]. Если условие истинно, то идти к шагу 28, иначе к шагу 31 |
| 28 | Temp := X[i,k+1] |
| 29 | X[i,k+1] := X[i,k] |
| 30 | X[i,k] := Temp |
| 31 | k := k + 1 |
| 32 | Конец цикла А7. Идти к шагу 26 |
| 33 | j := j + 1 |
| 34 | Конец цикла А6. Идти к шагу 24 |
| 35 | i := i + 1 |
| 36 | Конец цикла А5. Идти к шагу 22 |
| 37 | i := 1 |
| 38 | Начало цикла А8. Проверка выполнения условия i <= N. Если условие истинно, то идти к шагу 39, иначе к шагу 53 |
| 39 | k := 1 |
| 40 | Начало цикла А9. Проверка выполнения условия k <= N - i. Если условие истинно, то идти к шагу 41, иначе к шагу 51 |
| 41 | Проверка выполнения условия X[k,M] > X[k+1,M]. Если условие истинно, то идти к шагу 42, иначе к шагу 49 |
| 42 | j := 1 |
| 43 | Начало цикла А10. Проверка выполнения условия j <= M. Если условие истинно, то идти к шагу 44, иначе к шагу 49 |
| 44 | Temp := X[k+1,j] |
| 45 | X[k+1,j] := X[k,j] |
| 46 | X[k,j] := Temp |
| 47 | j := j + 1 |
| 48 | Конец цикла А10. Идти к шагу 43 |
| 49 | k := k + 1 |
| 50 | Конец цикла А9. Идти к шагу 40 |
| 51 | i := i + 1 |
| 52 | Конец цикла А8. Идти к шагу 38 |
| 53 | i := 1 |
| 54 | Начало цикла А11. Проверка выполнения условия i <= N. Если условие истинно, то идти к шагу 55, иначе к шагу 62 |
| 55 | j := 1 |

Продолжение таблицы 1

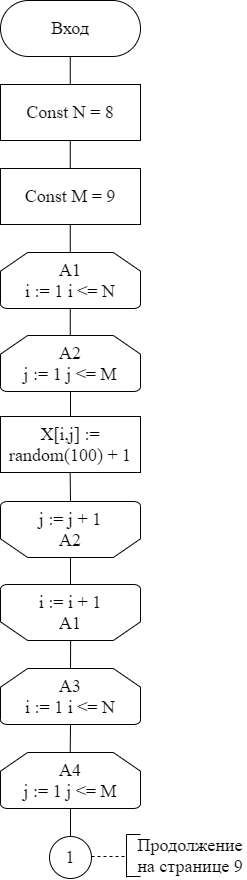
|  |  |
| --- | --- |
| 56 | Начало цикла А12. Проверка выполнения условия j <= M. Если условие истинно, то идти к шагу 57, иначе к шагу 60 |
| 57 | Вывод X[i,j] |
| 58 | j := j + 1 |
| 59 | Конец цикла А12. Идти к шагу 56 |
| 60 | i := i + 1 |
| 61 | Конец цикла А11. Идти к шагу 54 |
| 62 | Останов. |

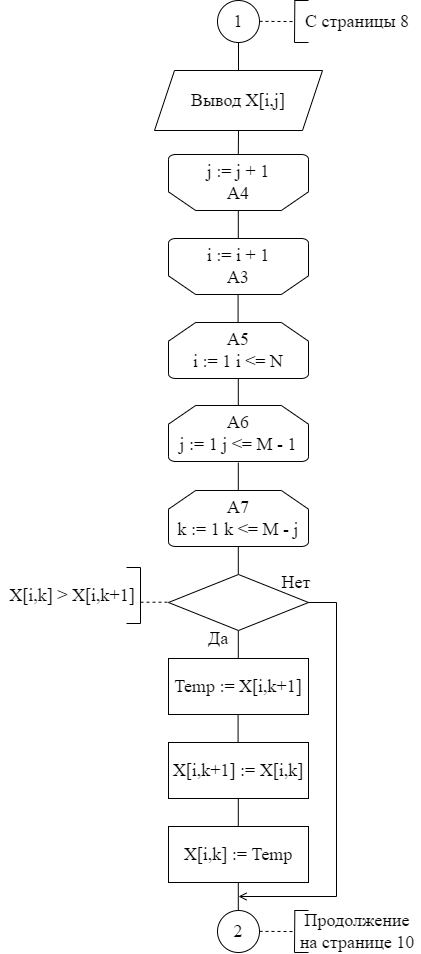
# Структура данных

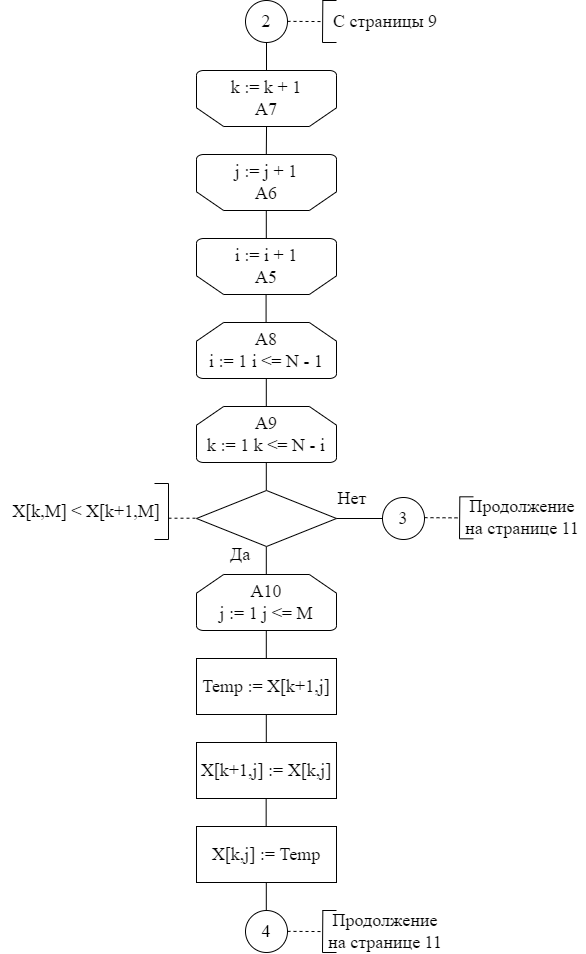
Таблица - Данные

|  |  |  |
| --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение |
| N | Const | Количество строк матрицы |
| M | Const | Количество столбцов матрицы |
| Mas | Array of integer | Массив |
| X | Mas | Массив |
| i | Integer | Счетчик для строк |
| j | Integer | Счетчик для столбцов |
| k | Integer | Счетчик для сортировки |
| Temp | Integer | Вспомогательная переменная, использующаяся для перестановки элементов |

# Схема алгоритма решения задачи по ГОСТ 19.701-90







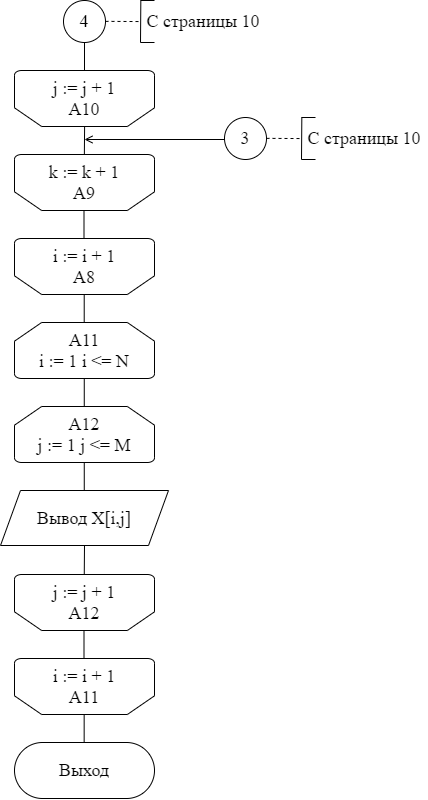


Рисунок - Схема алгоритма решения задачи по ГОСТ 19.701-90

# Результаты расчетов

Вследствие результатов программы на экран выводятся следующие результаты расчетов:

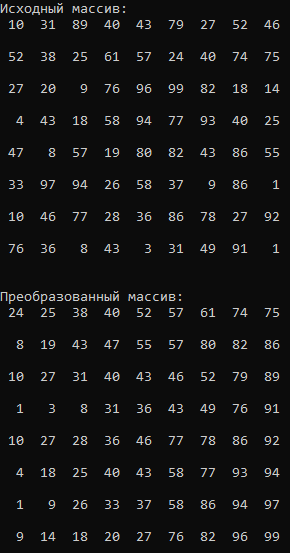


Рисунок – результаты расчетов

Приложение А

(обязательное)

Исходный код программы

Program Lab8;

//Program sort the array

//Console app

{$APPTYPE CONSOLE}

{$R \*.res}

//Modules declaration

Uses

System.SysUtils;

//Constants declaration

Const

N = 8;

M = 9;

//Types declaration

Type

Mas = array[1..N,1..M] of integer;

//Variables declaration

Var

X:Mas;

i, j, k, Temp: integer;

{ X - array

i - counter for raws

j - counter for columns

k - counter to sort arrays

Temp - auxillary variable which is used to

swap elements }

//Start the program

Begin

//Create new random numbers

Randomize;

//Fill the array with random elements

for i := 1 to N do

for j := 1 to M do

X[i,j] := random(100) + 1;

//Output original array

writeln('Исходный массив:');

for i := 1 to N do

begin

for j := 1 to M do

write(X[i,j]:3,' ');

writeln;

writeln;

end;

writeln;

//Sort elements in raws of array in growth order

for i := 1 to N do (\*To sort each raw step by step\*)

for j := 1 to M - 1 do (\*Reffering to each element of

current raw\*)

for k := 1 to M - j do (\*Bubble sort\*)

begin

//If next element bigger than previous one

if X[i,k] > X[i,k+1] then

//Swap elements

begin

Temp := X[i,k+1];

X[i,k+1] := X[i,k];

X[i,k] := Temp;

end;

end;

//Sort raws in decrease order

for i := 1 to N do (\*Reffering to each raw of array\*)

for k := 1 to N - i do (\*Bubble sort\*)

begin

//Biggest elements of each raw are in 9-th column,

//because of previous sort,

//sort, therefore, in order to compare biggest

//elements of each raw we

//need to compare all elements from 9-th column

if X[k,M] < X[k+1,M] then

begin

//Swap all elements from both raws

for j := 1 to M do

begin

Temp := X[k+1,j];

X[k+1,j] := X[k,j];

X[k,j] := Temp;

end;

end;

end;

//Output transformed array

writeln('Преобразованный массив:');

for i := 1 to N do

begin

for j := 1 to M do

write(X[i,j]:3,' ');

writeln;

writeln;

end;

//Final operations

readln;

End.

Приложение Б

(обязательное)

Тестовые наборы(Расчеты)

Тест 1

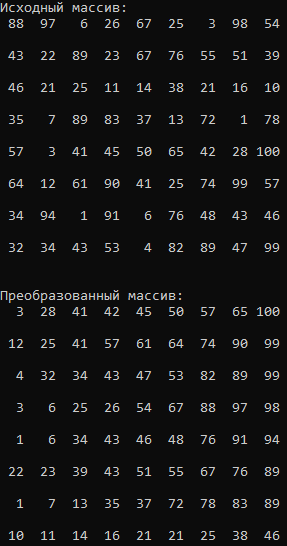
Тестовая ситуация: Стандартный режим работы программы

Исходные данные: Стандартный режим работы программы

Ожидаемый результат:

Программа рассортирует элементы строк по неубыванию, а сами строки по убыванию максимальных элементов строк

Полученный результат:



Тест 2

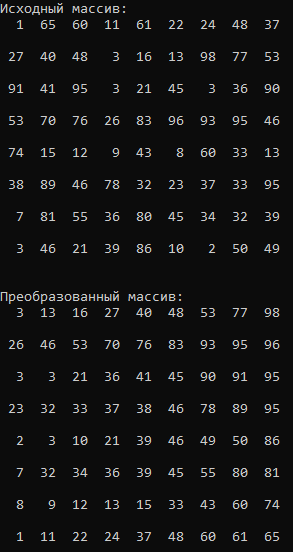
Тестовая ситуация: Стандартный режим работы программы

Исходные данные: Стандартный режим работы программы

Ожидаемый результат:

Программа рассортирует элементы строк по неубыванию, а сами строки по убыванию максимальных элементов строк

Полученный результат:



Тест 3

Тестовая ситуация: Стандартный режим работы программы

Исходные данные: Стандартный режим работы программы

Ожидаемый результат:

Программа рассортирует элементы строк по неубыванию, а сами строки по убыванию максимальных элементов строк

Полученный результат:

