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 двумерных массивов

Выполнил

студент: гр. 151003 Матошко И.В

Проверил: Фадеева Е.П.

Минск 2021

содержание

[1 Постановка задачи 3](#_Toc90810525)

[2 Текстовый алгоритм решения задачи 4](#_Toc90810526)

[3 Структура данных 6](#_Toc90810527)

[4 Схема алгоритма решения задачи по ГОСТ 19.701-90 7](#_Toc90810528)

[5 Результаты расчетов 11](#_Toc90810529)

[ПРИЛОЖЕНИЕ А 12](#_Toc90810530)

[ПРИЛОЖЕНИЕ Б 15](#_Toc90810531)

[ПРИЛОЖЕНИЕ В 19](#_Toc90810532)

[ПРИЛОЖЕНИЕ Г 23](#_Toc90810533)

# Постановка задачи

**1.1 Первоначальная постановка**

Дана матрица X[7,9]. Упорядочить элементы столбцов матрицы по убыванию, а сами столбцы по возрастанию элементов 1-ой строки.

**1.2 Окончательная постановка**

Дана матрица X[7,9]. Упорядочить элементы столбцов матрицы по убыванию, а сами столбцы по возрастанию элементов 1-ой строки.

Вывести на печать: Исходную матрицу, отсортированную матрицу.

# Текстовый алгоритм решения задачи

Таблица 1 ­­– Алгоритм решения

|  |  |
| --- | --- |
| Номер  шага | Назначение шага |
|  | Ввод X[1..7,1..9] |
|  | Вывод X[1..7,1..9] |
|  | i := 1 |
|  | Начало цикла А1. Проверка выполнения условия (i<=9). Если условие истинно, идти к шагу 5, иначе – к шагу 22 |
|  | sort:=true |
|  | l:=1 |
|  | Начало цикла А2. Проверка выполнения условия (l<=6) and (sort=true). Если условие истинно, идти к шагу 8, иначе – к шагу 20 |
|  | sort:=false |
|  | j := 1 |
|  | Начало цикла А3. Проверка выполнения условия (i<=7-l). Если условие истинно, идти к шагу 11, иначе – к шагу 18 |
|  | Проверка выполнения условия (X[j,i]<X[j+1,i]). Если условие истинно, идти к шагу 12, иначе – к шагу 16 |
|  | sort:=true |
|  | swap:= X[j,i] |
|  | X[j,i]:= X[j+1,i] |
|  | X[j+1,i]:= swap |
|  | j:=j+1 |
|  | Конец цикла А3. Идти к шагу 10 |
|  | l:=l+1 |
|  | Конец цикла А2. Идти к шагу 7 |
|  | i:=i+1 |
|  | Конец цикла А1. Идти к шагу 4 |
|  | st1:=9 |
|  | Начало цикла А4. Проверка выполнения условия (st>1). Если условие истинно, идти к шагу 24, иначе – к шагу 40 |
|  | id:=1 |
|  | j := 2 |
|  | Начало цикла А5. Проверка выполнения условия (j<=st1). Если условие истинно, идти к шагу 27, иначе – к шагу 31 |

Таблица 2 - Продолжение таблицы 1

|  |  |
| --- | --- |
|  | Проверка выполнения условия (X[1,j]>X[1,id]). Если условие истинно, идти к шагу 28, иначе – к шагу 29 |
|  | id:=j |
|  | j:=j+1 |
|  | Конец цикла А5. Идти к шагу 26 |
|  | i := 1 |
|  | Начало цикла А6. Проверка выполнения условия (i<=7). Если условие истинно, идти к шагу 33, иначе – к шагу 38 |
|  | max:=X[i,id] |
|  | X[i,id]:=X[i,st1] |
|  | X[i,st1]:=max |
|  | i:=i+1 |
|  | Конец цикла А6. Идти к шагу 32 |
|  | st1:=st1-1 |
|  | Конец цикла А4. Идти к шагу 23 |
|  | Вывод X[1..7,1..9] |
|  | Останов. |

# Структура данных

Таблица 3 – Данные

|  |  |  |
| --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение |
| X | Array of integer | Вводимый массив(матрица), который нужно отсортировать |
| st | Integer | Количество столбцов матрицы |
| str | Integer | Количество строк матрицы |
| st1 | Integer | Счетчик цикла по столбцам матрицы |
| i | Integer | Счетчик цикла |
| id | Integer | Индекс максимального элемента |
| max | Integer | Содержит максимальный элемент во время обмена |
| j | Integer | Счетчик цикла |
| l | Integer | Счетчик цикла |
| swap | Integer | Содержит элемент во время обмена |
| sort | Boolean | Логическая переменная для отслеживания обменов в сортировке «пузырьком с флажком» |

# 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



Рисунок 3 – Схема алгоритма решения задачи по ГОСТ 19.701-90



Рисунок 4 – Схема алгоритма решения задачи по ГОСТ 19.701-90

# Результаты расчетов

Вследствие выполнения программы на экран выводятся следующие результаты:

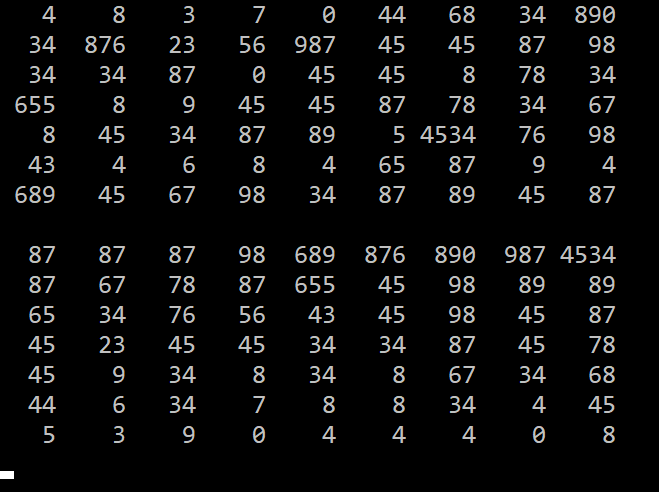


Рисунок 5 – Результаты расчетов

ПРИЛОЖЕНИЕ А

(обязательное)

Исходный код программы

Program lab8;

{ The matrix is given. Arrange the

elements of the matrix

columns in descending order,and the columns themselves

in ascending order of the elements of the first row. }

{$APPTYPE CONSOLE} // Console application

// Modules declaration

Uses

System.SysUtils;

// Variables declaration

Var

st,str,st1,i,id,max,j,l,swap,error:integer;

X:array [1..7,1..9] of integer;

vvod: string;

sort:boolean;

{ st - the number of input matrix columns;

str - number of input matrix rows;

st1 - column counter in a loop;

i - cycle counter;

id - number of the max element;

max - contain max element while swaping;

j - cycle counter;

l - cycle counter;

swap - contain element while swaping;

error - check input errors;

X - array of integer( input matrix);

vvod - check input errors.

sort - checking for sortin ig buble sort }

Begin

// Filling the array X

// Filling by line

for str := 1 to 7 do

begin

// Filling by column

for st := 1 to 9 do

repeat

begin

Writeln('Введите элемент ',str,' строки и ',st,'

столбца');

Readln(vvod);

// Checking for input

val(vvod,X[str,st],error);

if error>0 then

writeln('Вы неверно ввели член массива');

end;

until error=0;

end;

// Printing array X

for str := 1 to 7 do

begin

for st := 1 to 9 do

write(X[str,st]:4,' ');

writeln;

end;

writeln;

// Ordering elements of matrix columns

// in descending order

// External loop by columns

for i := 1 to 9 do

begin

// Start of bubble sorting

sort:=true;

l:=1;

while (l<=6) and (sort=true) do

begin

// Resetting the variable

sort:=false;

// Internal exchange cycle

for j := 1 to 7-l do

begin

// Swaping elements if less

if X[j,i]<X[j+1,i] then

begin

// Found at least 1 exchange

sort:=true;

// Swaping

swap:= X[j,i];

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

X[j+1,i]:= swap;

end;

end;

l:=l+1;

end;

end;

// Ordering columns in ascending order of the elements of

// the first row

// Sorting by choice

st1:=9;

// External loop by columns

while st1>1 do

begin

id:=1;

// Start sorting by selection

// Finding the index of the maximum element

for j := 2 to st1 do

if X[1,j]>X[1,id] then

begin

id:=j;

end;

// Exchange of elements

for i := 1 to 7 do

begin

// Swaping

max:=X[i,id];

X[i,id]:=X[i,st1];

X[i,st1]:=max;

end;

st1:=st1-1;

end;

// Printing new array X

for str := 1 to 7 do

begin

for st := 1 to 9 do

write(X[str,st]:4,' ');

writeln;

end;

readln;

readln;

End.

ПРИЛОЖЕНИЕ Б

(обязательное)

Тестовые наборы

Тестовый набор 1

«Проверка ввода»

Тест 1

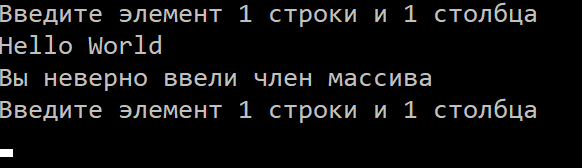
Тестовая ситуация: проверка ввода букв в поле для чисел

Исходные данные: X[1,1] = «Hello World»

Ожидаемый результат:

Вы неверно ввели член массива

Полученный результат:



Тест 2

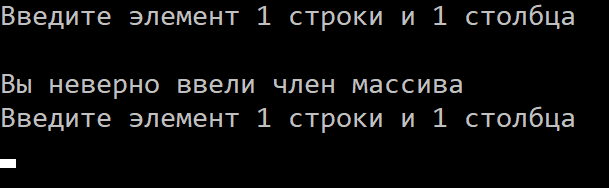
Тестовая ситуация: проверка ввода пустого значения в поле для чисел

Исходные данные: X[1,1] = « »

Ожидаемый результат:

Вы неверно ввели член массива

Полученный результат:



Тест 3

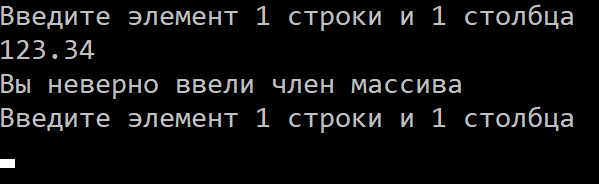
Тестовая ситуация: проверка ввода нецелочисленных значений

Исходные данные: X[1,1] = «123.34»

Ожидаемый результат:

Вы неверно ввели член массива

Полученный результат:



Тестовый набор 2

«Проверка решений»

Тест 1

Тестовая ситуация: проверка решений

Исходные данные: X = ((-77, -69, 83, 70, 16, 30, -12, 38, 44),

(21, 34, 99, -52, 83, -75, 90, 43, 56),

(-63, -87, 83, 69, -98, 89, -39, 123, 45),

(67, -39, -18, 72, 61, 53, 98, 67, 90),

(-83, 23, -85, 35, 49, 18, -98, 45, 77),

(92, -88, -17, 30, -39, -46, 76, 88, 99),

(59, 22, -43, -71, 86, 12, -10, 90, 5))

Ожидаемый результат:

34 72 86 89 92 98 99 99 123

23 70 83 53 67 90 90 83 90

22 69 61 30 59 76 77 83 88

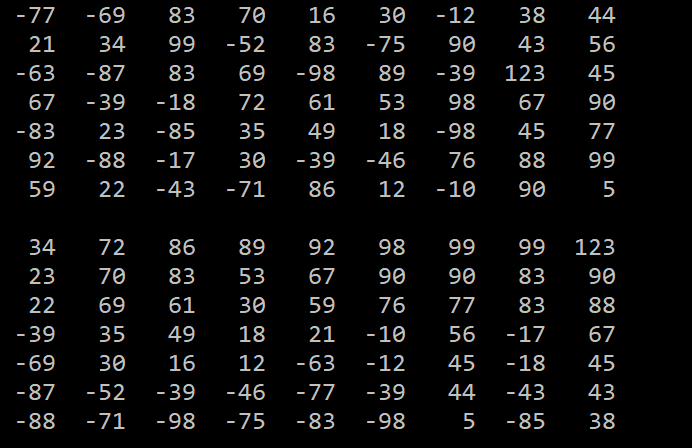
-39 35 49 18 21 -10 56 -17 67

-69 30 16 12 -63 -12 45 -18 45

-87 -52 -39 -46 -77 -39 44 -43 43

-88 -71 -98 -75 -83 -98 5 -85 38

Полученный результат:



Тест 2

Тестовая ситуация: проверка решений

Исходные данные: X = ((5, 7, 7, 300, 300, 300, 18, 38, 44),

(21, 34, 99, 52, 83, 75, 90, 43, 56),

(63, 87, 83, 69, 98, 89, 39, 123, 45,),

(67, 39, 18, 72, 61, 53, 98, 67, 90),

(83, 23, 85, 35, 49, 18, 98, 45, 77),

(92, 88, 17, 30, 39, 46, 76, 88, 99),

(59, 22, 43, 71, 86, 12, 10, 90, 5))

Ожидаемый результат(календарь Windows):

88 92 98 99 99 123 300 300 300

87 83 98 90 85 90 89 98 72

39 67 90 77 83 88 75 86 71

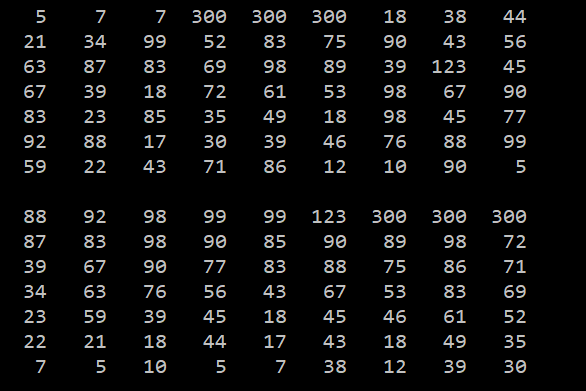
34 63 76 56 43 67 53 83 69

23 59 39 45 18 45 46 61 52

22 21 18 44 17 43 18 49 35

7 5 10 5 7 38 12 39 30

Полученный результат:



ПРИЛОЖЕНИЕ В

(обязательное)

Схема алгоритма решения задачи по методу Дамке



Рисунок 6 – Схема алгоритма решения задачи по методу Дамке



Рисунок 7 – Схема алгоритма решения задачи по методу Дамке



Рисунок 8 – Схема алгоритма решения задачи по методу Дамке



Рисунок 9 – Схема алгоритма решения задачи по методу Дамке

ПРИЛОЖЕНИЕ Г

(обязательное)

Схема алгоритма решения задачи по методу Дамке

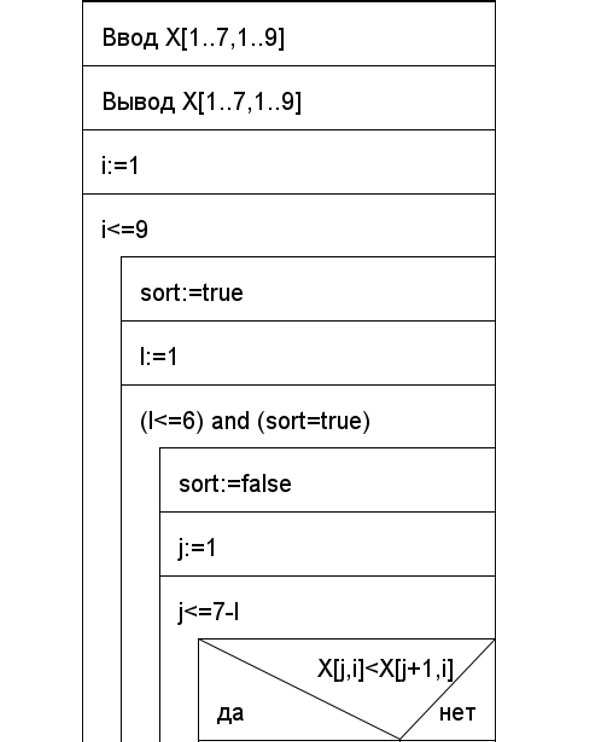


Рисунок 10 – Схема алгоритма решения задачи по методу

Насси-Шнейдермана

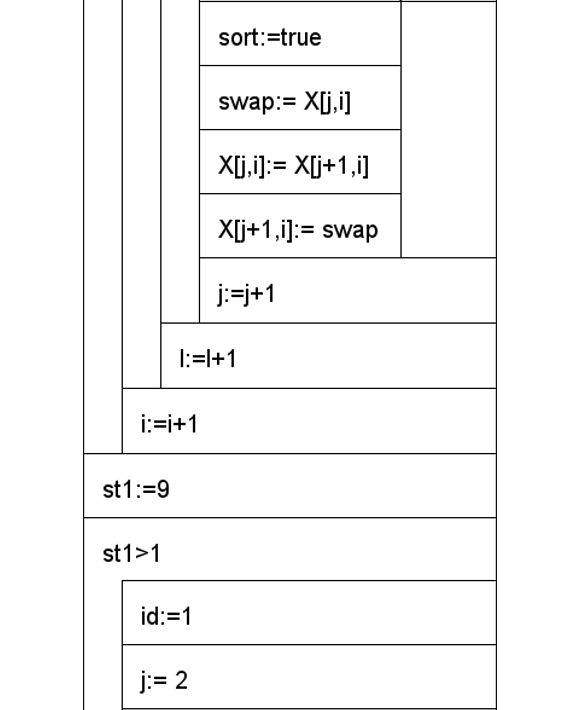


Рисунок 11 – Схема алгоритма решения задачи по методу

Насси-Шнейдермана

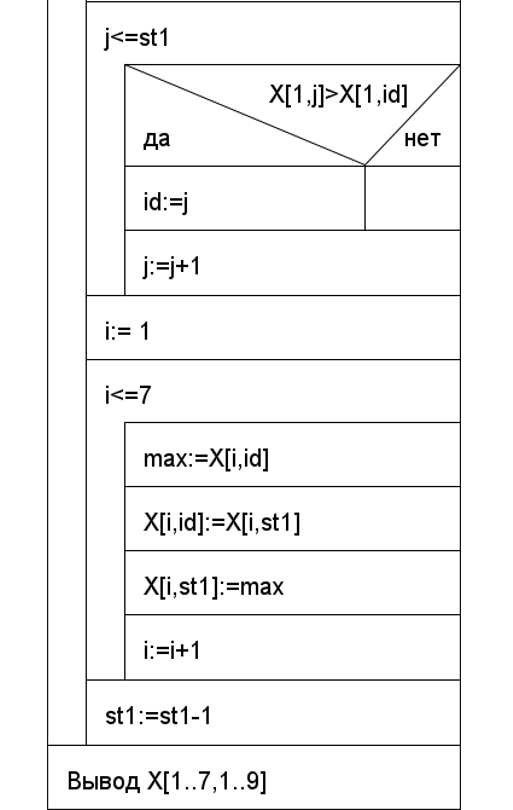


Рисунок 12 – Схема алгоритма решения задачи по методу

Насси-Шнейдермана