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 №7

Тема работы: Сортировка

Выполнил

студент: гр. 251003 Дедов Н.Ю.

Проверил: Фадеева Е.П.

Минск 2022

Содержание

[1 Постановка задачи 3](#_Toc124509492)

[1.1 Условие задачи 3](#_Toc124509493)

[1.2 Ввод данных 3](#_Toc124509494)

[1.3 Вывод данных 3](#_Toc124509495)

[2 Структура Данных 4](#_Toc124509496)

[3 Текстовый алгоритм данных 5](#_Toc124509497)

[4 Методика решения 7](#_Toc124509498)

[4.1 Условия для ввода данных 7](#_Toc124509499)

[4.2 Алгоритм решения 7](#_Toc124509500)

[5 Схема агоритма решения задачи по ГОСТ 19.701-90 8](#_Toc124509501)

[6 Результаты расчетов 10](#_Toc124509502)

[Приложение А 12](#_Toc124509503)

[Приложение Б 15](#_Toc124509504)

# Постановка задачи

## Условие задачи

Дана матрица X [9,7]. Упорядочить элементы строк матрицы по

неубыванию, а сами строки по убыванию минимальных элементов строк.

## Ввод данных

Пользователь вводит матрицу X [9,7].

## Вывод данных

Если все условия ввода соблюдены, то выводится матрица, отсортированная по условию задачи.

Если условия ввода не соблюдены, то программа просит повторно

ввести данные

# Структура Данных

Таблица 1 – Данные основной программы

|  |  |  |
| --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение |
| NMax | Integer | Количество допустимых строк и  столбцов |
| J | Integer | Переменная, необходимая для цикла |
| Mat | Array [1..NMax, 1..NMax] of Real | Матрица, вводимая пользователем |
| Str | Integer | Количество строк |
| St | Integer | Количество столбцов |
| StrMat | Integer | Строка матрицы |
| StMat | Integer | Столбец матрицы |
| IdMax | Integer | Индекс максимального элемента в строке |
| Max | Real | Максимальный элемент в строке |

Таблица 2 – Данные процедуры Swap

|  |  |  |
| --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение |
| X | Real | Переменная для обмена |
| Y | Real | Переменная для обмена |
| Temp | Real | Временное значение переменной |

# Текстовый алгоритм данных

Таблица 3 – Алгоритм решения

|  |  |
| --- | --- |
| Номер  Шага | Назначение шага |
|  | NMax:=20 |
|  | Ввод Mat[StrMat,StMat] |
|  | Начало цикла А1. Проверка выполнение условия: ( StrMat:= 1 <= Str ). Если условие истинно, то идти к шагу 4, иначе – к шагу 19 |
|  | J:=St |
|  | Начало цикла А2. Проверка выполнение условия: (J > 1). Если условие истинно, то идти к шагу 6, иначе – к шагу 18 |
|  | StMat:=1 |
|  | max := mat[strmat,stmat] |
|  | id\_max := Stmat |
|  | Начало цикла А3. Проверка выполнение условия: (StMat:=2 <= J). Если условие истинно, то идти к шагу 10, иначе – к шагу 14 |
|  | Проверка выполнения условия:( mat[strmat,stmat] > max ). Если условие истинно, то идти к шагу 11, иначе – к шагу 13 |
|  | max := mat[strmat,stmat] |
|  | id\_max := stmat |
|  | Конец цикла А3. Вернуться к шагу 9 |
|  | Mat[strmat,id\_max] := mat[strmat,j] |
|  | mat[strmat,j] := max |
|  | j := j - 1 |
|  | Конец цикла А2. Вернуться к шагу 5 |
|  | Конец цикла А1. Вернуться к шагу 3 |
|  | Начало цикла А4. Проверка выполнение условия: ( J:= 1 <= Str ). Если условие истинно, то идти к шагу 20, иначе – к шагу 27 |
|  | Начало цикла А5. Проверка выполнение условия: ( StrMat:= 1 <= Str - 1). Если условие истинно, то идти к шагу 21, иначе – к шагу 26 |
|  | Проверка выполнения условия:( Mat[StrMat,St]<Mat[StrMat+1,St]). Если условие истинно, то идти к шагу 22, иначе – к шагу 25 |
|  | Начало цикла А6. Проверка выполнение условия: ( StrMat:= 1 <= St). Если условие истинно, то идти к шагу 23, иначе – к шагу 25 |
|  | Swap(Mat[StrMat,StMat],Mat[StrMat+1,StMat]) |
|  | Конец цикла А6. Вернуться к шагу 22 |
|  | Конец цикла А5. Вернуться к шагу 20 |
|  | Конец цикла А4. Вернуться к шагу 19 |
|  | Вывод Mat[StrMat,StMat] |
|  | Останов. |

Таблица 4 – Алгоритм процедуры Swap

|  |  |
| --- | --- |
| Номер  Шага | Назначение шага |
|  | Temp:=X |
|  | X:=Y |
|  | Y:=Temp |
|  | Останов. |

# Методика решения

## Условия для ввода данных

Необходимо соблюдать следующие условия:

1. Вводимые данные должны быть числами.
2. Вводимые числа должны быть действительным.

## Алгоритм решения

Чтобы отсортировать матрицу сначала необходимо отсортировать ее строки выбором. Когда сортировка будет закончена, все наибольшие элементы строки будут стоять в конце этой строки, поэтому по ним необходимо сравнивать строки и в случае несоответствия их расположения, необходимо менять их местами.

# Схема агоритма решения задачи по ГОСТ 19.701-90



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



Рисунок 2 – Схема процедуры Swap по ГОСТ 19.701-90

# Результаты расчетов

Как результат выполнения программы, на экран выводятся следующие результаты расчетов:

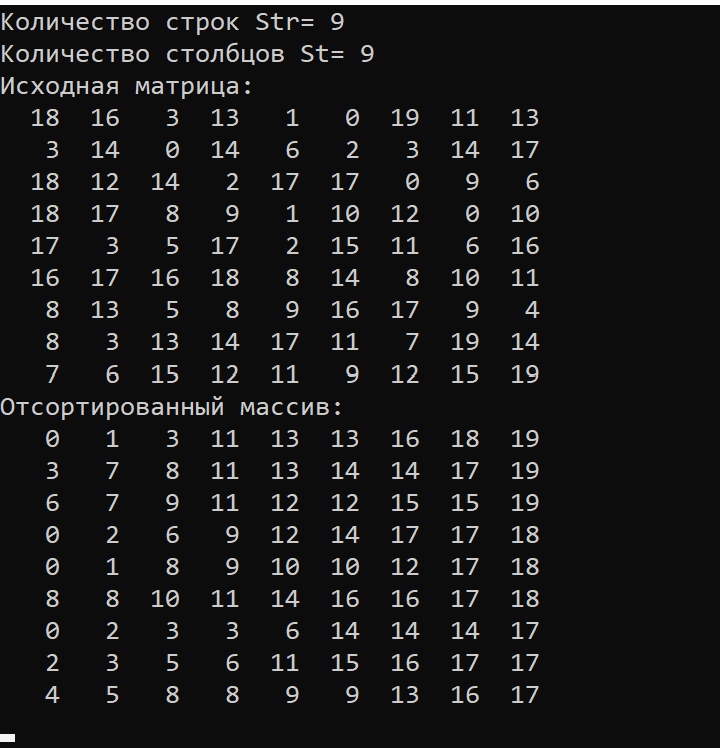


Рисунок 3 – Результаты расчетов при верном вводе данных

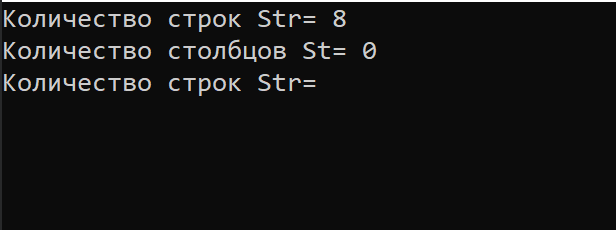


Рисунок 4 – Результаты расчетов при неверном вводе данных

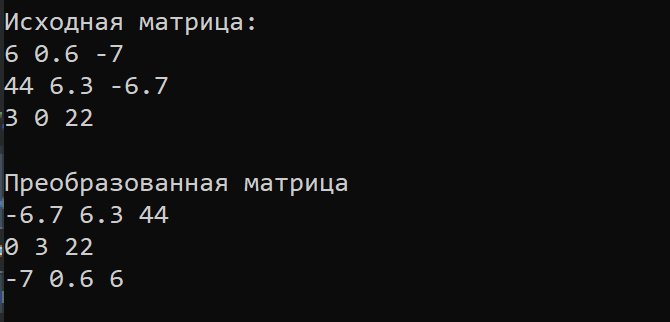


Рисунок 5 – Результаты расчетов при наличии дробных чисел

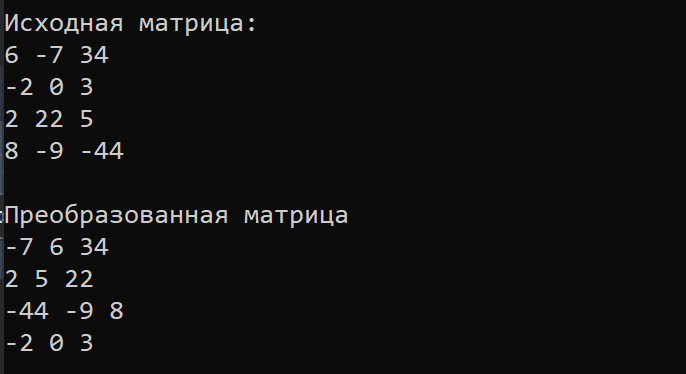


Рисунок 6 – Результаты расчетов при наличии отрицательных чисел

Приложение А

(обязательное)

Исходный код программы

Program Lab7;

{

User writes a matrix, program must arrange numbers in strings from lowest to highest and strings in matrix by lowest numbers in strings to highest

}

{$APPTYPE CONSOLE}

Uses

Windows, SysUtils;

Procedure swap(var x,y:integer);

var

temp: integer;

//X – first variable for changing

//Y – Second variable for changing

//Temp – for storing temporary variable

// Using this procedure to swap numbers in str

begin

temp:= x;

x:= y;

y:= temp;

end;

const

nmax=20;

var

Mat:array[1..nmax,1..nmax] of integer;

Str,St,StrMat,StMat,TempSt, TempMat:integer;

//Mat – our matrix that is given by a user

//Str – number of strings we have

//St – number of columns we have

//StrMat – the exact string we are now on

//StMat – the exact column we are now on

//TempMat – temporary number for the cycle

//TempSt – valuable for the cycle

Begin

repeat

write('Количество строк Str= ');

readln(Str);

write('Количество столбцов St= ');

readln(St);

until (Str in [1..nmax])and(St in [1..nmax]);

writeln('Исходная матрица: ');

for StrMat:=1 to Str do

begin

for StMat:=1 to St do

begin

writeln('Mat[',StrMat,' , ',Stmat,']= ');

readln(Mat[StrMat,StMat]);

end;

writeln;

end;

writeln;

writeln('Исходная матрица:');

for Strmat:=1 to str do

begin

for stmat:=1 to st do

begin

if Frac(mat[strmat,stmat])=0 then

write(Trunc(mat[strmat,stmat]),' ')

else

write(mat[strmat,stmat]:3:1,' ');

end;

writeln;

end;

writeln;

//We sorting numbers in strings for least num

for StrMat:=1 to Str do

begin

j:=st;

while j > 1 do

begin

stmat:=1;

max := mat[strmat,stmat];

id\_max := Stmat;

for stmat := 2 to j do

if mat[strmat,stmat] > max then

begin

max := mat[strmat,stmat];

id\_max := stmat;

end;

Mat[strmat,id\_max] := mat[strmat,j];

Mat[strmat,j] := max;

j:= j - 1;

end;

end;

//Now we sorting strings for the least number

for J:=1 to Str do

for StrMat:=1 to Str-1 do

// if prev string has less element change

if Mat[StrMat,St]<Mat[StrMat+1,St] then

for StMat:=1 to St do

Swap(Mat[StrMat,StMat], Mat[StrMat+

+1,StMat]);

writeln('Преобразованная матрица');

for Strmat:=1 to str do

begin

for stmat:=1 to st do

begin

if Frac(mat[strmat,stmat])=0 then

write(Trunc(mat[strmat,stmat]),' ')

else

write(mat[strmat,stmat]:3:1,' ');

end;

writeln;

end;

readln

end.

Приложение Б

(обязательное)

Тестовые наборы

|  |  |  |
| --- | --- | --- |
| Номер теста | Исходные данные | Полученный результат |
| 1 | Ввод верных данных | [Рисунок 3 – Результаты расчётов](#_Результаты_расчетов) |
| 2 | Ввод неверных данных | [Рисунок 4 – Результаты расчётов](#_Результаты_расчетов) |
| 3 | Ввод дробных чисел | [Рисунок 5 – Результаты расчётов](#_Результаты_расчетов) |
| 4 | Ввод отрицательных чисел | [Рисунок 6 – Результаты расчётов](#_Результаты_расчетов) |