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 №5

Тема работы: Работа с многомерными массивами

Выполнил

студент: гр. 551003 Дементей В.С.

Проверил: Фадеева Е.П.

Минск 2015

Содержание

[1 Постановка задачи 3](#_Toc437289353)

[2 Структура данных 4](#_Toc437289354)

[3 Текстовый алгоритм 5](#_Toc437289355)

[4 Схема алгоритма решения задачи по ГОСТ 19.701-90 7](#_Toc437289356)

[5 Схема алгоритма (представление по методу Дамке) 9](#_Toc437289357)

[Приложение А 12](#_Toc437289358)

[Приложение Б 15](#_Toc437289359)

# Постановка задачи

Дана матрица  размерности . В массиве  удалить строку с максимальным числом отрицательных элементов и столбец с максимальным числом положительных элементов, сдвинув исходный массив к левому верхнему углу. (Вывести исходный массив, новый массив со сдвигом, номер удаленной строки и номер удаленного столбца)

# Структура данных

Таблица - Данные

|  |  |  |  |
| --- | --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение | Примечание |
| length\_col | integer | Количество столбцов | Константа |
| length\_str | integer | Количество строк | Константа |
| Arr | Array[1..length\_col,1..length\_str] of integer | Массив чисел | Константа |
| Arr\_new | Array[1..length\_col,1..length\_str] of integer | Массив чисел на выходе |  |
| scorer\_str | integer | Счетчик значений по строкам |  |
| scorer\_col | integer | Счетчик значений по столбцам |  |
| max\_in\_str | integer | Максимальное количество отрицательных элементов в строке |  |
| max\_in\_col | integer | Максимальное количество положительных элементов в столбце |  |
| str | integer | Номер строки для удаления |  |
| col | integer | Номер столбца для удаления |  |
| shift\_str | integer | Сдвиг на строку |  |
| shift\_col | integer | Сдвиг на столбец |  |

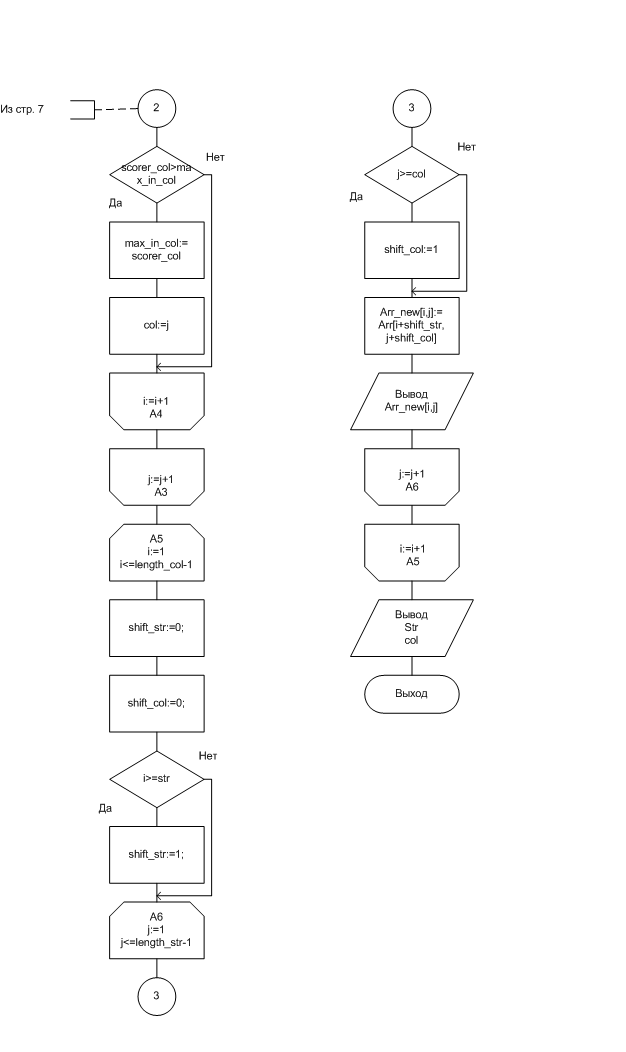
# Текстовый алгоритм

Таблица 2 - Текстовый алгоритм

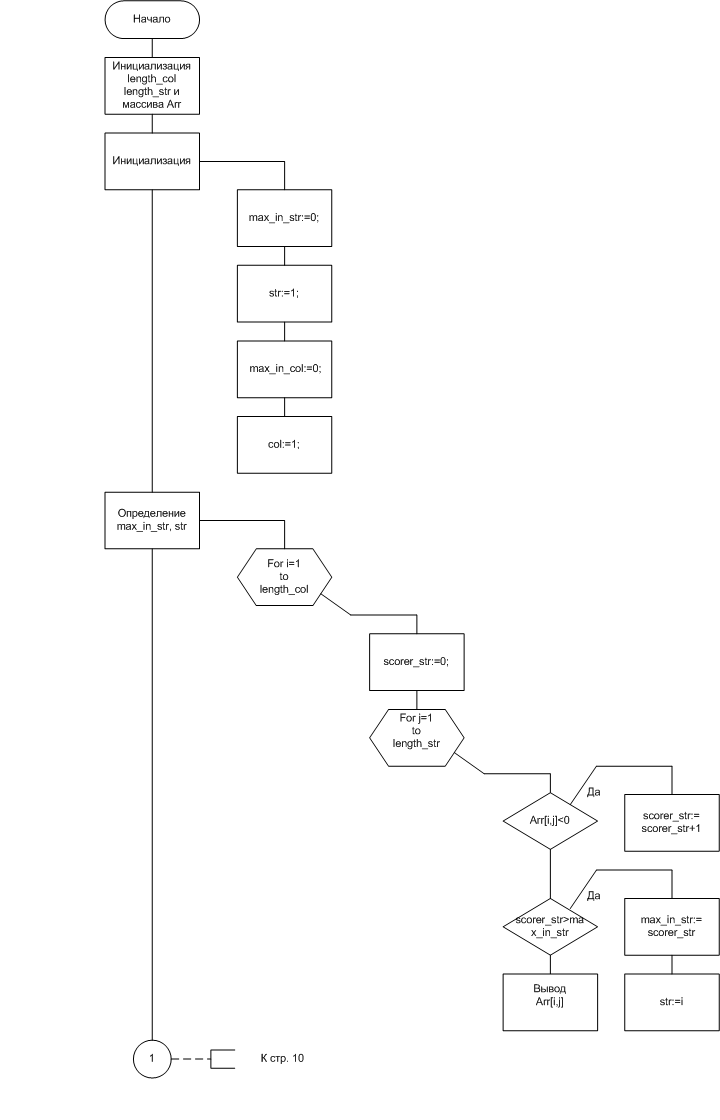
|  |  |
| --- | --- |
| Номер  шага | Назначение шага |
|  | Инициализация значений length\_col и length\_str  Инициализация значений массива Аrr |
|  | max\_in\_str:=0  str:=1  max\_in\_col:=0  col:=1 |
|  | i:=1 |
|  | Начало цикла А1. Проверка условия (i<= length\_col). Если оно истинно, то идти к шагу 5, иначе – к шагу 16. |
|  | scorer\_str:=0  j:=1 |
|  | Начало цикла А2. Проверка условия (j <= length\_str). Если оно истинно, то идти к шагу 7, иначе – к шагу 14. |
|  | Проверка выполнения условия (Arr[i,j]<0). Если оно истинно, то идти к шагу 8, иначе – к шагу 9. |
|  | scorer\_str:=scorer\_str+1 |
|  | Проверка выполнения условия (scorer\_str>max\_in\_str). Если оно истинно, то идти к шагу 10, иначе – к шагу 11. |
|  | max\_in\_str:=scorer\_str  str:=i |
|  | Вывод значения Arr[i,j] |
|  | j:=j+1 |
|  | Конец цикла A2. Идти к шагу 6. |
|  | i:=i+1 |
|  | Конец цикла A1. Идти к шагу 4. |
|  | j:=1 |
|  | Начало цикла А3. Проверка условия (j <= length\_str). Если оно истинно, то идти к шагу 18, иначе – к шагу 29. |
|  | scorer\_col:=0 |
|  | i:=1 |
|  | Начало цикла А4. Проверка условия (i<= length\_col). Если оно истинно, то идти к шагу 21, иначе – к шагу 27. |
|  | Проверка выполнения условия (Arr[i,j]>0). Если оно истинно, то идти к шагу 22, иначе – к шагу 23. |
|  | scorer\_col:=scorer\_col+1 |
|  | Проверка выполнения условия (scorer\_col>max\_in\_col). Если оно истинно, то идти к шагу 24, иначе – к шагу 25. |
|  | max\_in\_col:=scorer\_col;  col:=j |
|  | i:=i+1 |
|  | Конец цикла A4. Идти к шагу 20. |
|  | j:=j+1 |
|  | Конец цикла A3. Идти к шагу 17. |
|  | i:=1 |
|  | Начало цикла А5. Проверка условия (i<= length\_col-1). Если оно истинно, то идти к шагу 31, иначе – к шагу 44. |
|  | shift\_str:=0  shift\_col:=0 |
|  | Проверка выполнения условия (i>=str). Если оно истинно, то идти к шагу 33, иначе – к шагу 34. |
|  | shift\_str:=1 |
|  | j:=1 |
|  | Начало цикла А6. Проверка условия (j<=length\_str-1). Если оно истинно, то идти к шагу 36, иначе – к шагу 42 |
|  | Проверка выполнения условия (j>=col). Если оно истинно, то идти к шагу 37, иначе – к шагу 38 |
|  | shift\_col:=1 |
|  | Arr\_new[i,j]:=Arr[i+shift\_str,j+shift\_col]; |
|  | Вывод значения Arr\_new[i,j] |
|  | j:=j+1 |
|  | Конец цикла A6. Идти к шагу 35. |
|  | i:=i+1 |
|  | Конец цикла A5. Идти к шагу 30. |
|  | Вывод значений str, col |
|  | Останов. |

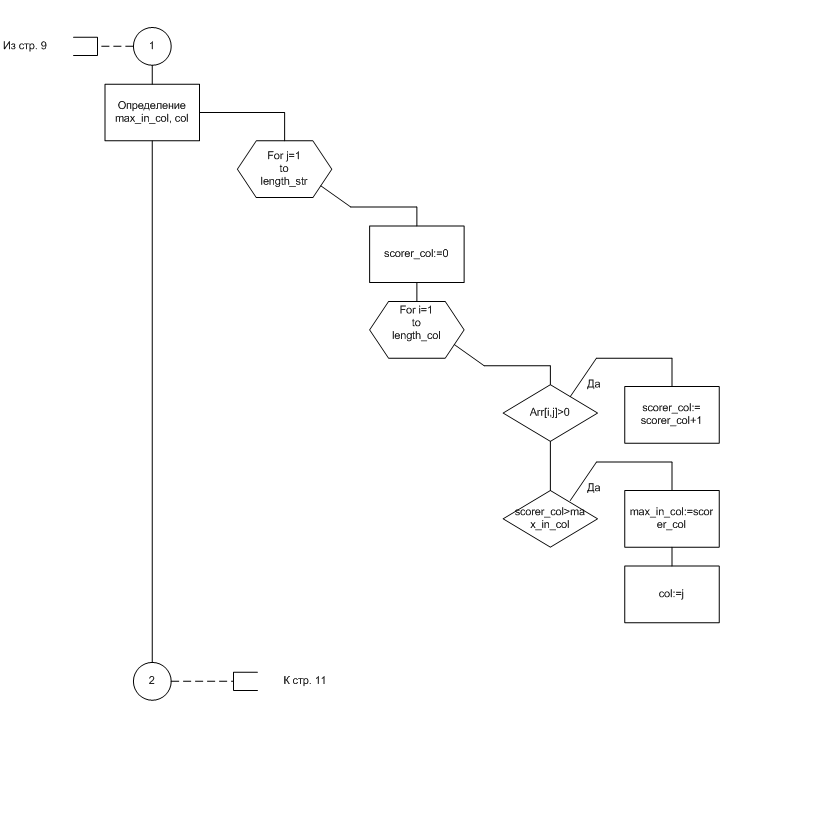
# C:\Users\Mike\Desktop\Labs\Labs\5 массив\Массивы ТСonst версия 2 часть 1.pngСхема алгоритма решения задачи по ГОСТ 19.701-90

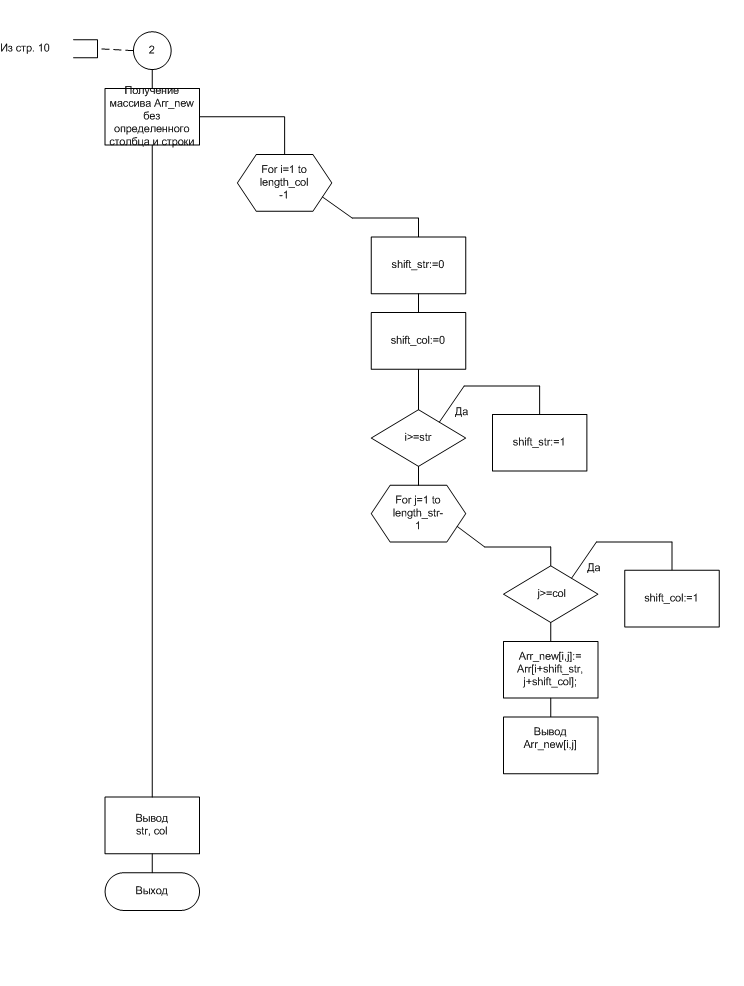
Рисунок 1 – Схема алгоритма (часть 1)

** Рисунок 2 – Схема алгоритма (часть 2)**

# Схема алгоритма (представление по методу Дамке)

 Рисунок 3 – Схема алгоритма по методу Дамке (часть 1)

 Рисунок 4 – Схема алгоритма по методу Дамке (часть 2)

 Рисунок 5 – Схема алгоритма по методу Дамке (часть 3)

Приложение А

(обязательное)

Исходный код программы

**Program** Laba\_5\_TConst;

{$APPTYPE CONSOLE}

**Uses**

SysUtils;

**Const**

{Определение размерности массива}

length\_col=6;

length\_str=5;

{input : string ='0123456789';}

**Const**

{Обьявление и инициализация массива}

Arr : **Array**[1..length\_col,1..length\_str] **of** integer =

(( 11 , 12 , 13, 14,-15),

(-21 ,-22 , 23,-24, 25),

( 31 ,-32 , 33, 34,-35),

( 41 ,-42 , 43, 44, 45),

( 51 , 52 ,-53,-54, 55),

( 61 , 62 , 63, 64, 65));

**Var**

Arr\_new : **Array**[1..length\_col,1..length\_str] **of** integer;

scorer\_str,scorer\_col : integer;

i,j : integer;

max\_in\_str,max\_in\_col : integer;

str,col : integer;

shift\_str,shift\_col : integer;

{Arr : **Array**[1..length\_col,1..length\_str] **of** integer;}

{socer,v,k : integer;

input\_str : string;}

**Begin**

{Генерация случайных чисел в диапазоне [-5;5]

randomize;

**for** i:=1 **to** length\_col **do**

**for** j:=1 **to** length\_str **do**

Arr[i,j]:= random(10)-5; }

{Инициализация массива

**for** i:=1 **to** length\_col **do**

**for** j:=1 **to** length\_str **do**

**repeat**

writeln('Enter element ',i\*10+j);

readln(input\_str);

socer:=0;

**if** (input\_str[1])=(chr(45)) **then**

socer:=socer+1;

**if** (socer=1) **and** ((length(input\_str))=1) **then**

socer:=socer+1;

**for** k:=1 **to** (length(input\_str)) **do**

**for** v:=1 **to** (length(input)) **do**

**if** (chr(ord(input[v])))=(input\_str[k]) **then**

socer:=socer+1;

**if** socer<>(length(input\_str)) **then**

writeln('Error retype')

**else**

Arr[i,j]:=strtoint(input\_str);

**until** (socer=(length(input\_str)));}

max\_in\_str:=0;

str:=1;

max\_in\_col:=0;

col:=1;

{Поиск нужного столбца}

**for** i:=1 **to** length\_col **do**

**begin**

scorer\_str:=0;

**for** j:=1 **to** length\_str **do**

**begin**

**if** Arr[i,j]<0 **then**

scorer\_str:=scorer\_str+1;

**if** scorer\_str>max\_in\_str **then**

**begin**

max\_in\_str:=scorer\_str;

str:=i;

**end**;

**if** (Arr[i,j])>=0 **then**

write(' ', Arr[i,j],' ')

**else**

write(Arr[i,j],' ');

**end**;

writeln;

**end**;

{Поиск нужной строки}

**for** j:=1 **to** length\_str **do**

**begin**

scorer\_col:=0;

**for** i:=1 **to** length\_col **do**

**begin**

**if** Arr[i,j]>0 **then**

scorer\_col:=scorer\_col+1;

**if** scorer\_col>max\_in\_col **then**

**begin**

max\_in\_col:=scorer\_col;

col:=j

**end**;

**end**;

**end**;

writeln;

{Cдвиг и вывод}

**for** i:=1 **to** (length\_col-1) **do**

**begin**

shift\_str:=0;

shift\_col:=0;

**if** i>=str **then**

shift\_str:=1;

**for** j:=1 **to** (length\_str-1) **do**

**begin**

**if** j>=col **then**

shift\_col:=1;

Arr\_new[i,j]:=Arr[i+shift\_str,j+shift\_col];

**if** (Arr\_new[i,j])>=0 **then**

write(' ', Arr\_new[i,j],' ')

**else**

write(Arr\_new[i,j],' ');

**end**;

writeln;

**end**;

writeln;

writeln('string - ',str);

writeln('column -', col);

readln;

**End**.

Приложение Б

(обязательное)

Тестовые наборы

Тест 1.1

Исходные данные:

( 11 , 12 ,-13,-14,-15),

(-21 ,-22 , 23,-24,-25),

( 31 ,-32 , 33, 34,-35),

( 41 ,-42 , 43, 44, 45),

( 51 , 52 ,-53,-54, 55),

( 61 , 62 , 63, 64, 65)

Ожидаемый результат: Удаление второй строки и первого столбца

( 12 ,-13,-14,-15),

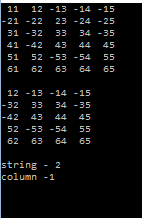
(-32 , 33, 34,-35),

(-42 , 43, 44, 45),

( 52 ,-53,-54, 55),

( 62 , 63, 64, 65)

Полученный результат:



Тест 1.2

Исходные данные:

( 11 , 12 ,-13,-14,-15),

(-21 ,-22 , 23,-24,-25),

( 31 ,-32 , 33, 34,-35),

(-41 ,-42 , 43, 44, 45),

(-51 , 52 ,-53,-54, 55),

(-61 ,-62 ,-63,-64,-65)

Ожидаемый результат: Удаление шестой строки и третьего столбца

( 11 , 12 ,-14,-15),

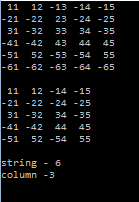
(-21 ,-22 ,-24,-25),

( 31 ,-32 , 34,-35),

(-41 ,-42 , 44, 45),

(-51 , 52 ,-54, 55)

Полученный результат:



Тест 2.1

Исходные данные:

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

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

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

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

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

( 0 , 0 , 0 , 0 , 0 )

Ожидаемый результат: Поскольку нуль не имеет знака, произвести удаление первой строки и первого столбца

( 0 , 0 , 0 , 0),

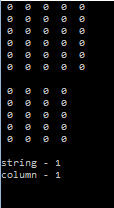
( 0 , 0 , 0 , 0),

( 0 , 0 , 0 , 0),

( 0 , 0 , 0 , 0),

( 0 , 0 , 0 , 0),

Полученный результат:



Тест 2.2

Исходные данные:

( 11 , 12 , 13, 14, 15),

( 21 , 22 , 23, 24, 25),

( 31 , 32 , 33, 34, 35),

( 41 , 42 , 43, 44, 45),

( 51 , 52 , 53, 54, 55),

( 61 , 62 , 63, 64, 65)

Ожидаемый результат: Поскольку отсутствуют отрицательные числа, произвести удаление первой строки и первого столбца

Исходные данные:

( 22 , 23, 24, 25),

( 32 , 33, 34, 35),

( 42 , 43, 44, 45),

( 52 , 53, 54, 55),

( 62 , 63, 64, 65)

Полученный результат:

