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.551004 Довыдёнок М.А.

Проверила: Фадеева Е.П.

Минск 2015

Содержание

[1 Постановка задачи 3](#_Toc436850985)

[2 Текстовый алгоритм решения задачи 4](#_Toc436850986)

[3 Структура данных 6](#_Toc436850987)

[4 Схема алгоритма решения задачи по ГОСТ 19.701-90 7](#_Toc436850988)

[5 Схема алгоритма методом Дамке 9](#_Toc436850989)

[Приложение А 11](#_Toc436850990)

[Приложение Б 18](#_Toc436850991)

# Постановка задачи

Дана матрица  размерности . Из матрицы  удалить все строки и столбцы, содержащие хотя бы один нулевой элемент. Матрицу уплотнить, сдвинув оставшиеся элементы к левому верхнему углу.

# Текстовый алгоритм решения задачи

Таблица 1 – Текстовый алгоритм

|  |  |
| --- | --- |
| Номер  шага | Назначение шага |
|  | Ввод: n, m |
|  | Инициализация X[1..n, 1..m] |
|  | i:=1 |
|  | Начало цикла А1. Проверка выполнения условия (i<=n). Если условие истинно, то идти к шагу 5, иначе – к шагу 8 |
|  | delRow[i]:=false |
|  | i:=i+1 |
|  | Конец цикла А1. Идти к шагу 4 |
|  | j:=1 |
|  | Начало цикла А2. Проверка выполнения условия (j<=m). Если условие истинно, то идти к шагу 10, иначе – к шагу 13 |
|  | delCol[j]:=false |
|  | j:=j+1 |
|  | Конец цикла А2. Идти к шагу 9 |
|  | i:=1 |
|  | Начало цикла А3. Проверка выполнения условия (i<=n). Если условие истинно, то идти к шагу 15, иначе – к шагу 24 |
|  | j:=1 |
|  | Начало цикла А4. Проверка выполнения условия (j<=m). Если условие истинно, то идти к шагу 17, иначе – к шагу 22 |
|  | Проверка выполнения условия: (X[i, j]=0). Если условие истинно, то идти к шагу 18, иначе – к шагу 20 |
|  | delRow[i]:=true |
|  | delCol[j]:=true |
|  | j:=j+1 |
|  | Конец цикла А4. Идти к шагу 16 |
|  | i:=i+1 |
|  | Конец цикла А3. Идти к шагу 14 |
|  | i:=n |
|  | Начало цикла А5. Проверка выполнения условия (i>=1). Если условие истинно, то идти к шагу 26, иначе – к шагу 39 |
|  | Проверка выполнения условия: (delRow[i]). Если условие истинно, то идти к шагу 27, иначе – к шагу 37 |
|  | y:=i |
|  | Начало цикла А6. Проверка выполнения условия (y<=n-1). Если условие истинно, то идти к шагу 29, иначе – к шагу 36 |
|  | j:=1 |
|  | Начало цикла А7. Проверка выполнения условия (j<=m). Если условие истинно, то идти к шагу 31, иначе – к шагу 34 |
|  | X[y, j]:=X[y+1, j] |
|  | j:=j+1 |
|  | Конец цикла А7. Идти к шагу 30 |
|  | y:=y+1 |
|  | Конец цикла А6. Идти к шагу 28 |
|  | dec(n) |
|  | dec(i) |
|  | Конец цикла А5. Идти к шагу 25 |
|  | j:=m |
|  | Начало цикла А8. Проверка выполнения условия (j>=1). Если условие истинно, то идти к шагу 41, иначе – к шагу 54 |
|  | Проверка выполнения условия: (delCol[j]). Если условие истинно, то идти к шагу 42, иначе – к шагу 52 |
|  | y:=j |
|  | Начало цикла А9. Проверка выполнения условия (y<=m-1). Если условие истинно, то идти к шагу 44, иначе – к шагу 51 |
|  | i:=1 |
|  | Начало цикла А10. Проверка выполнения условия (i<=n). Если условие истинно, то идти к шагу 46, иначе – к шагу 49 |
|  | X[I, y]:=X[i, y+1] |
|  | i:=i+1 |
|  | Конец цикла А10. Идти к шагу 45 |
|  | y:=y+1 |
|  | Конец цикла А9. Идти к шагу 43 |
|  | dec(m) |
|  | dec(j) |
|  | Конец цикла А8. Идти к шагу 40 |
|  | Вывод: X[1..n, 1..m] |
|  | Останов. |

# Структура данных

Таблица 2 – Используемые переменные

|  |  |  |
| --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение |
| n | Integer | Количество строк в матрице |
| m | Integer | Количество столбцов в матрице |
| i | Integer | Параметр цикла |
| j | Integer | Параметр цикла |
| y | Integer | Параметр цикла |
| X | Array[1..1000, 1..1000] of Integer | Матрица |
| delCol | Array [1..1000] of Boolean | Массив указывающий на подлежащие удалению столбцы |
| delRow | Array [1..1000] of Boolean | Массив указывающий на подлежащие удалению строки |

# Схема алгоритма решения задачи по ГОСТ 19.701-90

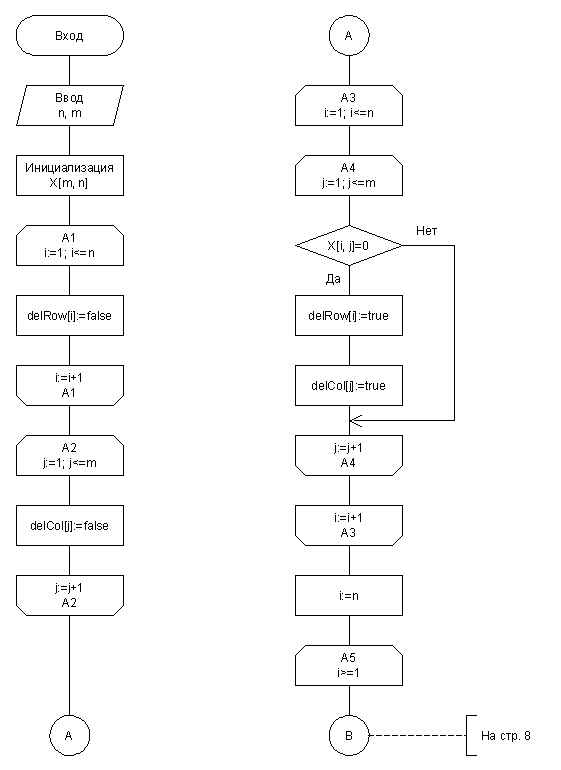


Рисунок 1 – Схема алгоритма решения задачи по ГОСТ 19.701-90

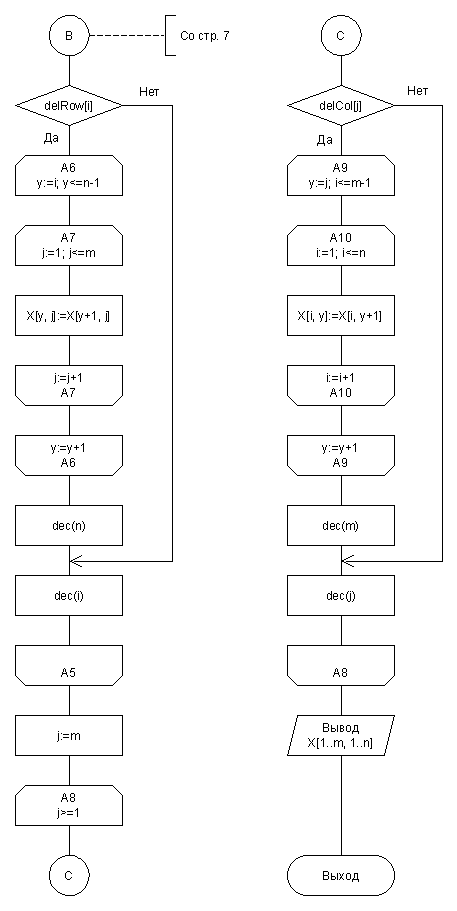


Рисунок 2 – Схема алгоритма решения задачи по ГОСТ 19.701-90

# Схема алгоритма методом Дамке

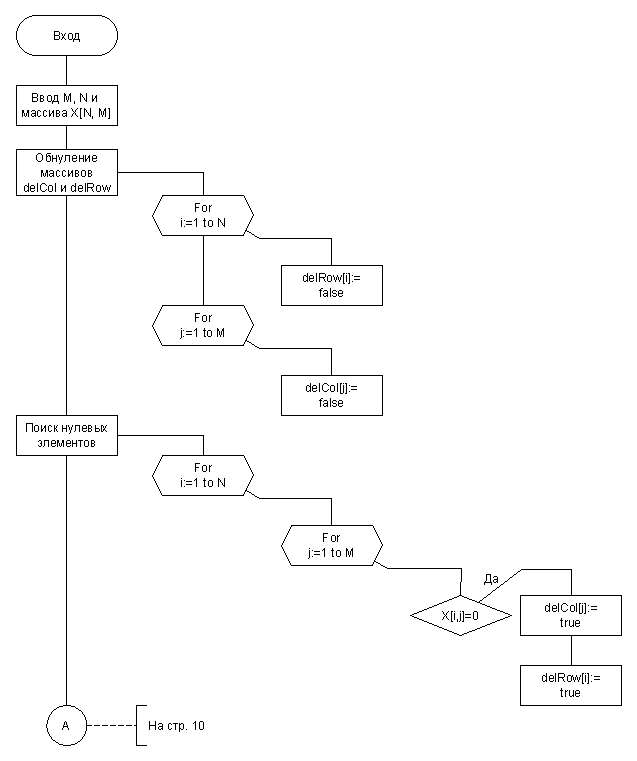


Рисунок 3 – Схема алгоритма методом Дамке

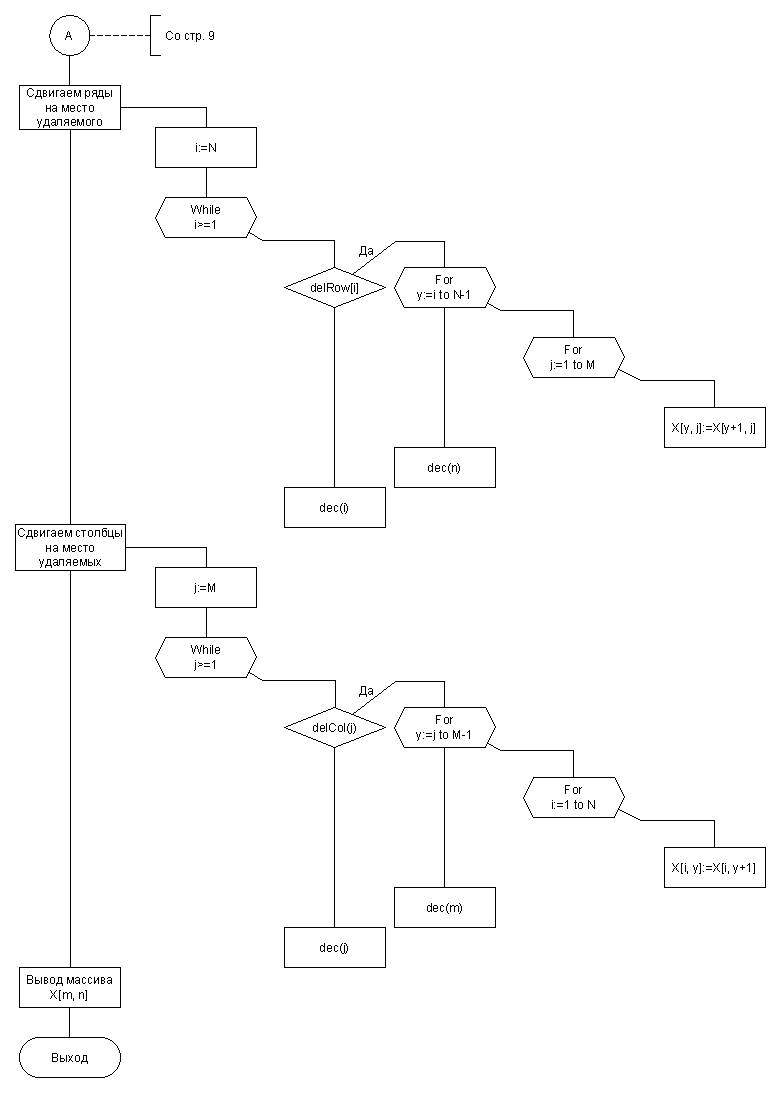


Рисунок 4 – Схема алгоритма методом Дамке

Приложение А

(обязательное)

Исходный код программы

Вариант 1.

{ Массив заполняется пользователем с клавиатуры }

program L1Z5\_42v1;

{$APPTYPE CONSOLE}

uses

Windows;

var

n, m: integer;

i, j: integer;

y: integer;

X: array [1..1000, 1..1000] of integer;

delCol, delRow: array [1..1000] of boolean;

begin

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

{ Ввод размера матрицы }

{$I-}

repeat

write('Введите N: ');

readln(n);

until IOResult=0;

repeat

write('Введите M: ');

readln(m);

until IOResult=0;

{ Заполнение матрицы }

for i:=1 to n do

begin

for j:=1 to m do

begin

repeat

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

readln(X[i, j]);

until IOResult=0;

end;

end;

{$I+}

{ Обнуление массивов }

for i:=1 to n do

delRow[i]:=false;

for j:=1 to m do

delCol[j]:=false;

{ Находим нулевые элементы }

{ Обозначаем удаляемые ряды }

for i:=1 to n do

begin

for j:=1 to m do

begin

if X[i, j]=0 then

begin

delCol[j]:=true;

delRow[i]:=true;

end;

end;

end;

{ Сдвигаем ряды на место удаляемого }

{ При удалении уменьшаем вертикальный размер }

i:=n;

while i>=1 do

begin

if delRow[i] then

begin

for y:=i to n-1 do

begin

for j:=1 to m do

begin

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

end;

end;

dec(n);

end;

dec(i);

end;

{ Сдвигаем столбцы на место удаляемого }

{ При удалении уменьшаем горизонтальный размер }

j:=m;

while j>=1 do

begin

if delCol[j] then

begin

for y:=j to m-1 do

begin

for i:=1 to n do

begin

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

end;

end;

dec(m);

end;

dec(j);

end;

{ Вывод полученной матрицы }

writeln(chr(10), chr(13), 'Полученная матрица:');

for i:=1 to n do

begin

for j:=1 to m do

write(X[i, j]:2, ' ');

writeln;

end;

readln;

end.

Вариант 2.

{ Массив заполняется случайными числами }

program L1Z5\_42v2;

{$APPTYPE CONSOLE}

uses

Windows;

var

n, m: integer;

i, j: integer;

y: integer;

X: array [1..1000, 1..1000] of integer;

delCol, delRow: array [1..1000] of boolean;

begin

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

{ Ввод размера матрицы }

{$I-}

repeat

write('Введите N: ');

readln(n);

until IOResult=0;

repeat

write('Введите M: ');

readln(m);

until IOResult=0;

{$I+}

{ Заполнение матрицы }

for i:=1 to n do

for j:=1 to m do

X[i, j]:=random(10);

{ Вывод исходной матрицы }

writeln('Исходная матрица:');

for i:=1 to n do

begin

for j:=1 to m do

write(X[i, j]:2, ' ');

writeln;

end;

{ Обнуление массивов }

for i:=1 to n do

delRow[i]:=false;

for j:=1 to m do

delCol[j]:=false;

{ Находим нулевые элементы }

{ Обозначаем удаляемые ряды }

for i:=1 to n do

begin

for j:=1 to m do

begin

if X[i, j]=0 then

begin

delCol[j]:=true;

delRow[i]:=true;

end;

end;

end;

{ Сдвигаем ряды на место удаляемого }

{ При удалении уменьшаем вертикальный размер }

i:=n;

while i>=1 do

begin

if delRow[i] then

begin

for y:=i to n-1 do

begin

for j:=1 to m do

begin

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

end;

end;

dec(n);

end;

dec(i);

end;

{ Сдвигаем столбцы на место удаляемого }

{ При удалении уменьшаем горизонтальный размер }

j:=m;

while j>=1 do

begin

if delCol[j] then

begin

for y:=j to m-1 do

begin

for i:=1 to n do

begin

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

end;

end;

dec(m);

end;

dec(j);

end;

{ Вывод полученной матрицы }

writeln(chr(10), chr(13), 'Полученная матрица:');

for i:=1 to n do

begin

for j:=1 to m do

write(X[i, j]:2, ' ');

writeln;

end;

readln;

end.

Вариант 3.

{ Массив заполняется с использованием

типизированной переменной-массива }

program L1Z5\_42v3;

{$APPTYPE CONSOLE}

uses

Windows;

var

n, m: integer;

i, j: integer;

y: integer;

X: array [1..3, 1..6] of integer =

((1, 3, 5, 7, 10, 0),

(0, 8, 5, 3, 0, 11),

(2, 4, 6, 8, 10, 12));

delCol, delRow: array [1..1000] of boolean;

begin

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

n:=3;

m:=6;

{ Вывод исходной матрицы }

writeln('Исходная матрица: ');

for i:=1 to n do

begin

for j:=1 to m do

write(X[i, j]:2, ' ');

writeln;

end;

{ Обнуление массивов }

for i:=1 to n do

delRow[i]:=false;

for j:=1 to m do

delCol[j]:=false;

{ Находим нулевые элементы }

{ Обозначаем удаляемые ряды }

for i:=1 to n do

begin

for j:=1 to m do

begin

if X[i, j]=0 then

begin

delCol[j]:=true;

delRow[i]:=true;

end;

end;

end;

{ Сдвигаем ряды на место удаляемого }

{ При удалении уменьшаем вертикальный размер }

i:=n;

while i>=1 do

begin

if delRow[i] then

begin

for y:=i to n-1 do

begin

for j:=1 to m do

begin

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

end;

end;

dec(n);

end;

dec(i);

end;

{ Сдвигаем столбцы на место удаляемого }

{ При удалении уменьшаем горизонтальный размер }

j:=m;

while j>=1 do

begin

if delCol[j] then

begin

for y:=j to m-1 do

begin

for i:=1 to n do

begin

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

end;

end;

dec(m);

end;

dec(j);

end;

{ Вывод полученной матрицы }

writeln(chr(10), chr(13), 'Полученная матрица: ');

for i:=1 to n do

begin

for j:=1 to m do

write(X[i, j]:2, ' ');

writeln;

end;

readln;

end.

Приложение Б

(обязательное)

Тестовые наборы

Тест 1.1 Позитивный тест

Тестовая ситуация: Введённые с клавиатуры исходные данные

Исходные данные:

n = 3

m = 3

X =

Ожидаемый результат:

Полученный результат: Рисунок 5 – Тест 1.1

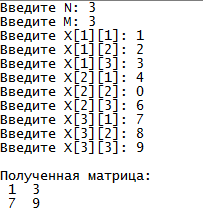


Рисунок 5 – Тест 1.1

Тест 1.2 Позитивный тест

Тестовая ситуация: Случайные исходные данные

Исходные данные:

n = 3

m = 3

Ожидаемый результат:

Зависит от «Исходная матрица»

Полученный результат: Рисунок 6 – Тест 1.2

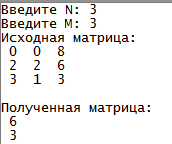


Рисунок 6 – Тест 1.2

Тест 1.3 Позитивный тест

Тестовая ситуация: Исходные данные записаны в коде программы

Исходные данные:

n = 3

m = 6

X =

Ожидаемый результат:

Полученный результат: Рисунок 7 – Тест 1.3

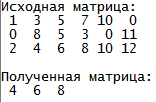


Рисунок 7 – Тест 1.3

Тест 2.1 Негативный тест

Тестовая ситуация: Некорректные исходные данные

Исходные данные:

n = a

n = 1111111111111111111111

Ожидаемый результат:

Повтор ввода

Полученный результат: Рисунок 8 – Тест 2.1



Рисунок 8 – Тест 2.1