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:

Сортировки

Выполнил

студент: гр. 151004 Данилов Ф.А.

Проверил: Фадеева Е.П.

Минск 2021

Содержание

[1 Постановка задачи 3](#_Toc90218542)

[2 Текстовый алгоритм решения задачи 4](#_Toc90218543)

[3 Структура данных 6](#_Toc90218544)

[4 Схема алгоритма решения задачи по ГОСТ 19.701-90 7](#_Toc90218545)

[5 Результаты расчетов 11](#_Toc90218546)

[Приложение А 12](#_Toc90218547)

[Приложение Б 16](#_Toc90218548)

[Приложение В 18](#_Toc90218549)

[Приложение Г 19](#_Toc90218550)

# Постановка задачи

Дана матрица . Упорядочить элементы столбцов матрицы по невозрастанию, а сами столбцы в соответствии с ростом модулей минимальных элементов столбцов.

# Текстовый алгоритм решения задачи

Таблица 1 – Алгоритм решения

|  |  |
| --- | --- |
| Номер  шага | Назначение шага |
|  | N = 7 |
|  | M = 10 |
|  | Ввод X[1..N, 1..M] |
|  | j := 1 |
|  | Начало цикла А1. Проверка выполнения условия (j ≤ M). Если условие истинно, идти к шагу 6, иначе – к шагу 18 |
|  | i := 1 |
|  | Начало цикла А2. Проверка выполнения условия (i ≤ N). Есди условие истинно, идти к шагу 8, иначе – к шагу 15 |
|  | k := i |
|  | temp := X[i, j] |
|  | Начало цикла А3. Проверка выполнения условия (k > 0) and (X[k-1, j] < temp). Если условие истинно, идти к шагу 11, иначе – к шагу 14 |
|  | X[k, j] := X[k-1, j] |
|  | k := k – 1 |
|  | Конец цикла А3. Идти к шагу 10 |
|  | i := i + 1 |
|  | Конец цикла А2. Идти к шагу 7 |
|  | j := j + 1 |
|  | Конец цикла А1. Идти к шагу 5 |
|  | j := 1 |
|  | Начало цикла А4. Проверка выполнения условия (j ≤ M). Если условие истинно, идти к шагу 20, иначе – к шагу 30 |
|  | min := abs(X[1, j]) |
|  | i := 1 |
|  | Начало цикла А5. Проверка выполнения условия (i ≤ N). Если условие истинно, идта к шагу 23, иначе – к шагу 27 |
|  | Проверка выполнения условия (abs(X[i, j]) < min). Если условие истинно, идти к шагу 24, иначе – к шагу 25 |
|  | min := abs(X[i, j]) |
|  | i := i + 1 |
|  | Конец цикла А5. Идти к шагу 22 |
|  | MinAbs[j] := min |
|  | j := j + 1 |
|  | Конец цикла А4. Идти к шагу 19 |
|  | j := 1 |

Продолжение таблицы 1

|  |  |
| --- | --- |
| Номер  шага | Назначение шага |
|  | Начало цикла А6. Проверка выполнения условия (j ≤ M). Если условие истинно, идти к шагу 32, иначе – к шагу 51 |
|  | min := j |
|  | k := j + 1 |
|  | Начало цикла А7. Проверка выполнения условия (k ≤ M). Если условие истинно, идти к шагу 35, иначе – к шагу 39 |
|  | Проверка выполнения условия (MinAbs[k] < MinAbs[min]). Если условие истинно, идти к шагу 36, иначе – к шагу 37 |
|  | min := k |
|  | k := k + 1 |
|  | Конец цикла А7. Идти к шагу 34 |
|  | i := 1 |
|  | Начало цикла А8. Проверка выполнения условия (i ≤ N). Если условие истинно, идти к шагу 41, иначе – к шагу 46 |
|  | temp := X[i, j] |
|  | X[i, j] := X[i, min] |
|  | X[i, min] := X[i, j] |
|  | i := i + 1 |
|  | Конец цикла А8. Идти к шагу 40 |
|  | temp := MinAbs[j] |
|  | MinAbs[j] := MinAbs[min] |
|  | MinAbs[min] := MinAbs[j] |
|  | j := j + 1 |
|  | Конец цикла А6. Идти к шагу 31 |
|  | Вывод X[1..N, 1..M] |
|  | Останов. |

# Структура данных

Таблица 2 – Данные

|  |  |  |
| --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение |
| N | Integer | Количество строк в массиве |
| M | Integer | Количество столбцов в массиве |
| X | Array [1..N, 1..M] of Integer | Массив, над котором производят сортировки |
| MinAbs | Array [1..M] of Integer | Массив, содержащий модули минимальных элементов столбцов массива |
| i | Integer | Счетчик цикла |
| j | Integer | Счетчик цикла |
| k | Integer | Счетчик цикла |
| min | Integer | Переменная для нахождения минимального значения |
| temp | Integer | Дополнительная переменная для перестановки значений двух элементов массива |

# Схема алгоритма решения задачи по ГОСТ 19.701-90

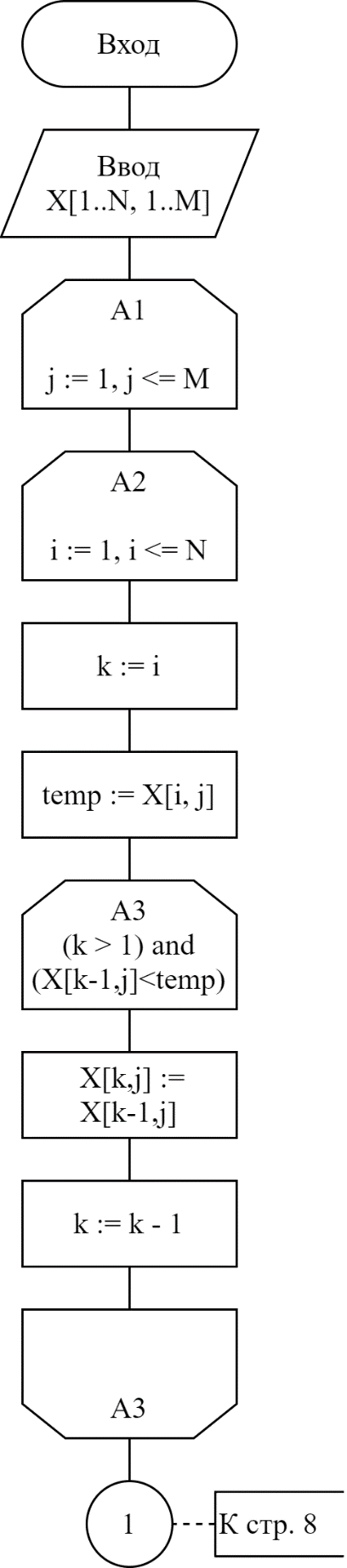


Рисунок 1 - Схема алгоритма решения задачи по ГОСТ 19.701-90 (часть 1)

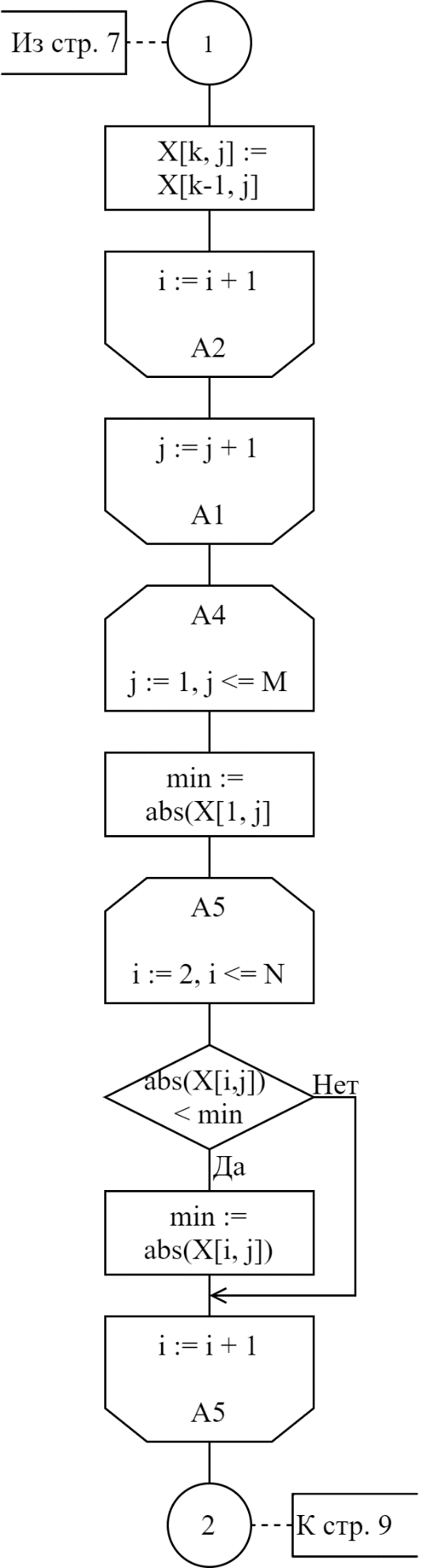


Рисунок 2 - Схема алгоритма решения задачи по ГОСТ 19.701-90 (часть 2)

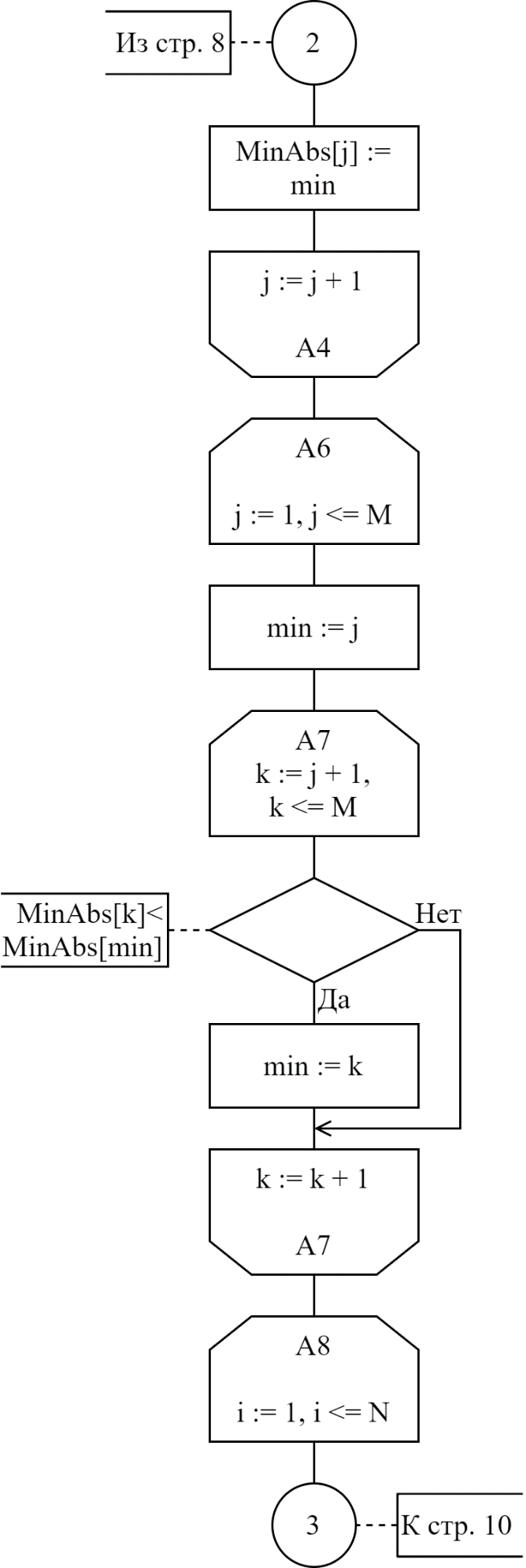


Рисунок 3 - Схема алгоритма решения задачи по ГОСТ 19.701-90 (часть 3)

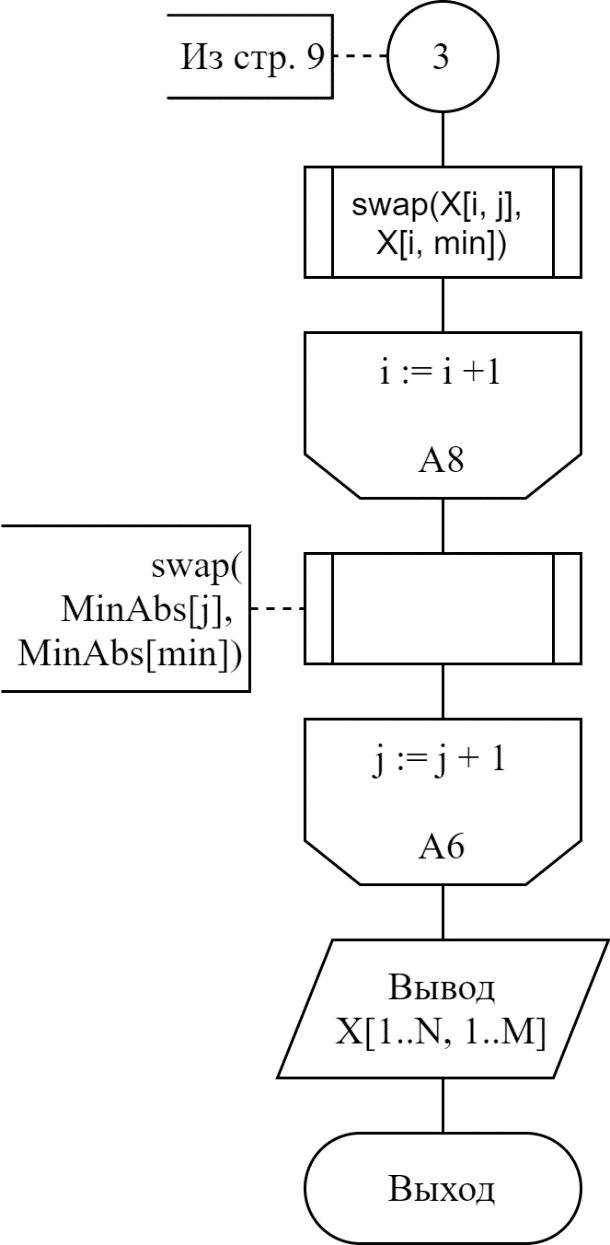


Рисунок 4 - Схема алгоритма решения задачи по ГОСТ 19.701-90 (часть 4)

# Результаты расчетов

Вследствие выполнения программы на экран выводятся следующие результаты:

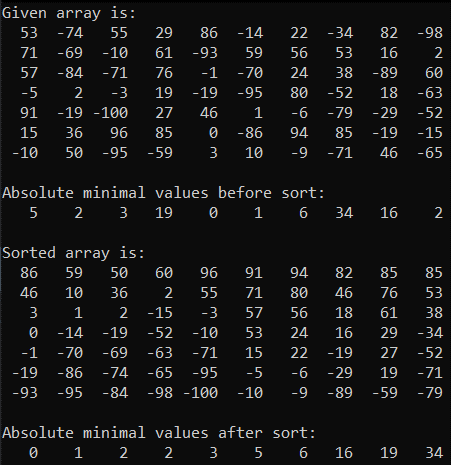


Рисунок 5 - Результаты расчётов

Приложение А

(обязательное)

program Lab8;

{

Sort given array X[1..7,1..10] by 2 rules:

1. Sort elements in columns by non-increase

2. Sort columns by increase of absolute minimal

value in each column

Input: Array X[1..7,1..10]

Output: Sorted array

}

{$APPTYPE CONSOLE}

{$R \*.res}

uses

System.SysUtils;

// N - number of rows in array X

// M - number of columns in array X

const

N = 7;

M = 10;

// X - given array

// MinAbs - array of minimal absolute values

// i,j,k - counters for loops

// min - minimal value

// temp - temp value to swap 2 elements

// InputStr - string for input

// Error - error value

var

X: Array[1..N+1, 1..M] of Integer;

MinAbs: Array[1..M] of Integer;

i, j, k, min, temp, Error: Integer;

InputStr: String;

begin

// Input X[1..N, 1..M] using method val

writeln('Input X[1..N, 1..M]:');

for i := 1 to N do

for j := 1 to M do

repeat

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

readln(InputStr);

val(InputStr, X[i,j], Error);

if Error <> 0 then

writeln('Error! Enter a valid number');

until Error = 0;

// Output X

writeln('Given array is:');

for i := 1 to N do

begin

for j := 1 to M do

write(X[i,j]:4, ' ');

writeln;

end;

// Sort elements in columns by non-increase

// using insertion sort

for j := 1 to M do

begin

// Iterate through rows

for i := 1 to N do

begin

// Initialize first minimal element's index

k := i;

temp := X[i,j];

// Iterete until element is bigger than

// previous one

while (k > 1) and (X[k-1,j] < temp) do

begin

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

k := k - 1;

end;

// Insert value right after minimal one

X[k, j] := temp;

end;

end;

// Find absolute minimal value in each column

for j := 1 to M do

begin

// Set first minimal value in column

min := abs(X[1,j]);

// Find absolute minimal value in column

// Iterate through rows

for i := 2 to N do

if abs(X[i,j]) < min then

min := abs(X[i,j]);

// Set absolute minimal value in end of column

MinAbs[j] := min;

end;

// Output absolute minimal values before sort

writeln;

writeln('Absolute minimal values before sort:');

for j := 1 to M do

write(MinAbs[j]:4, ' ');

writeln;

// Sort columns by increase of absolute minimal value

// using selection sort

// Iterate through columns

for j := 1 to M do

begin

// Initialize minimal column and first row

min := j;

// Find minimal column

for k := j + 1 to M do

if MinAbs[k] < MinAbs[min] then

min := k;

// Swap columns

for i := 1 to N do

begin

temp := X[i,j];

X[i,j] := X[i,min];

X[i,min] := temp;

end;

// Swap MinAbs values

temp := MinAbs[j];

MinAbs[j] := MinAbs[min];

MinAbs[min] := temp;

end;

// Output X

writeln;

writeln('Sorted array is:');

for i := 1 to N do

begin

for j := 1 to M do

write(X[i,j]:4, ' ');

writeln;

end;

// Output absolute minimal values after sort

writeln;

writeln('Absolute minimal values after sort:');

for j := 1 to M do

write(MinAbs[j]:4, ' ');

// Stop console from closing

readln;

end.

Приложение Б

(обязательное)

Тестовые наборы для проверки ввода данных

Тест 1

Тестовая ситуация: ввод вещественного числа

Исходные данные:

Ожидаемый результат:



Полученный результат:



Тест 2

Тестовая ситуация: введено значение, не являющееся числом

Исходные данные:

Ожидаемый результат:



Полученный результат:



Тест 3

Тестовая ситуация: ввод значения, выходящего за границу типа Integer

Исходные данные:

Ожидаемый результат:



Полученный результат:

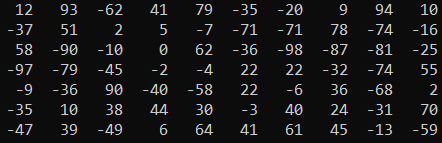


Тестовые наборы для проверки вычислений

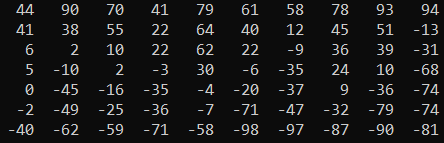
Тест 1

Тестовая ситуация: проверка сортировки

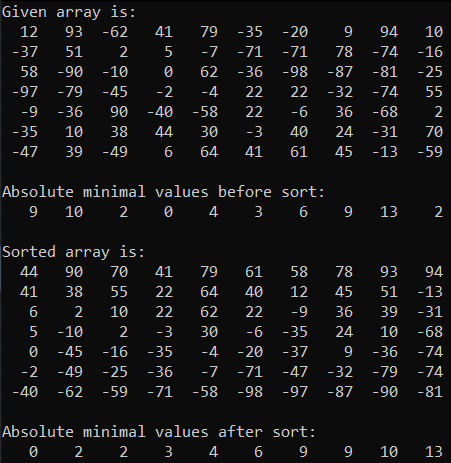
Исходные данные:



Ожидаемый результат:



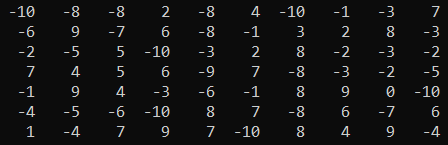
Полученный результат:



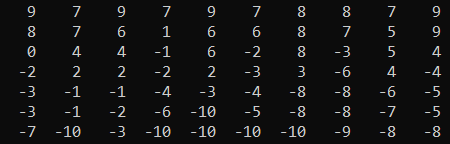
Тест 2

Тестовая ситуация: проверка сортировки

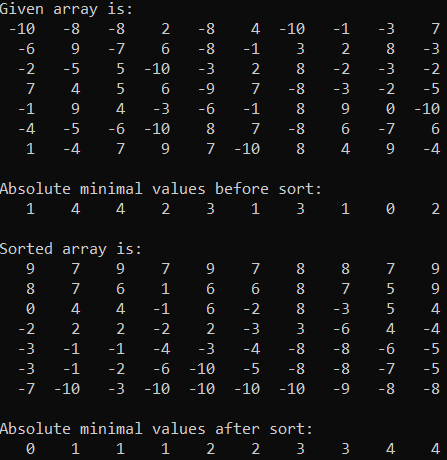
Исходные данные:



Ожидаемый результат:



Полученный результат:



Приложение В

(обязательное)

Схема алгоритма решения задачи по Дамке

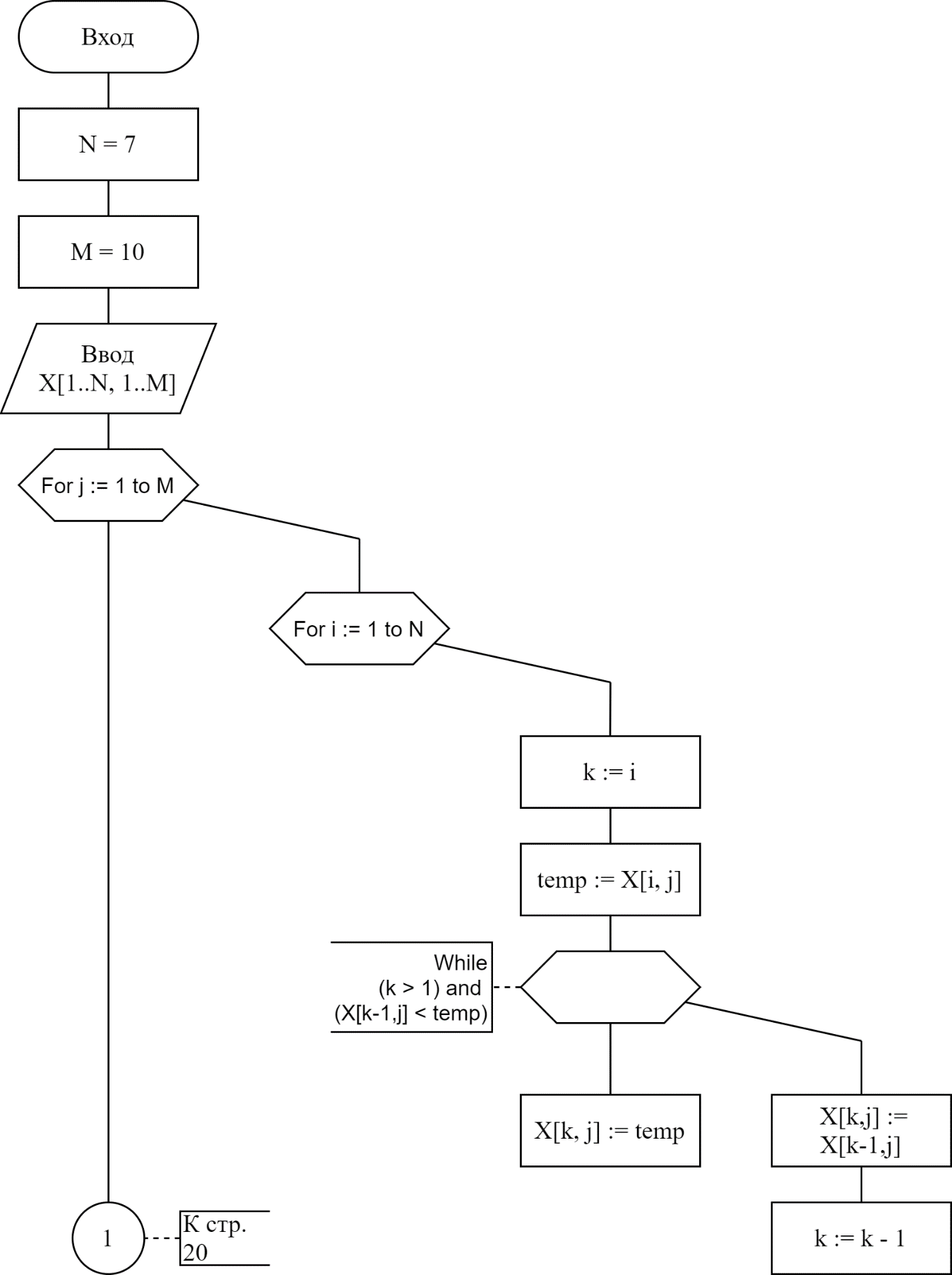


Рисунок 6 - Схема алгоритма решения задачи по методу Дамке (часть 1)

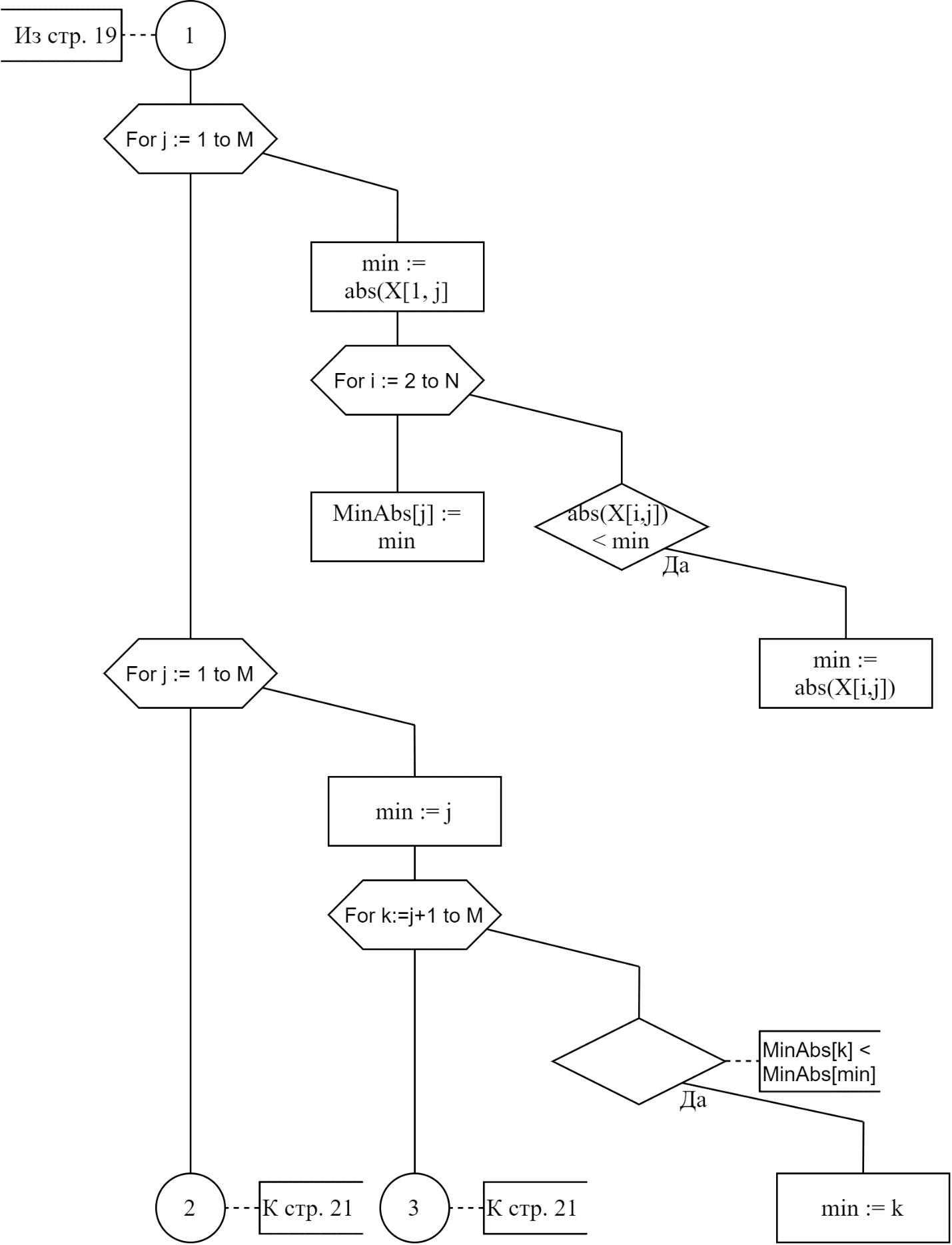


Рисунок 7 - Схема алгоритма решения задачи по методу Дамке (часть 2)

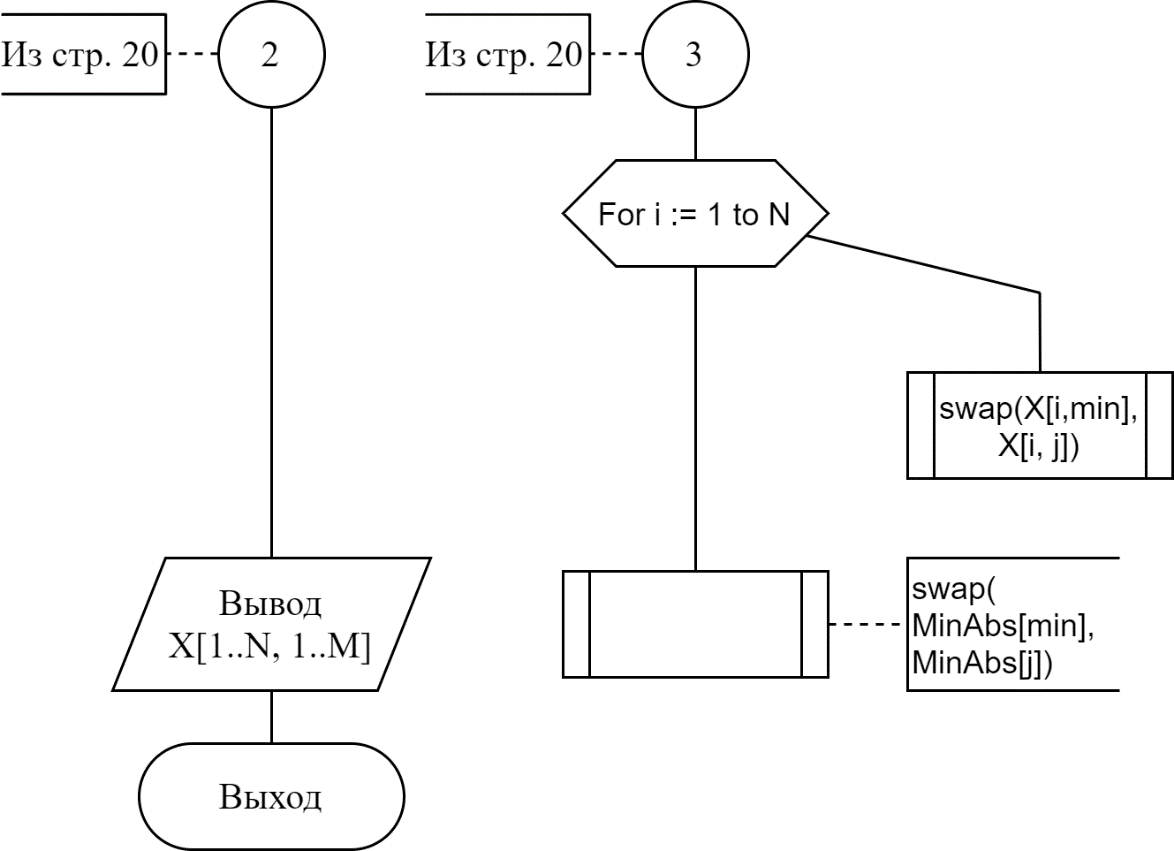


Рисунок 8 - Схема алгоритма решения задачи по методу Дамке (часть 3)

Приложение Г

(обязательное)

Схема алгоритма решения задачи по Насси-Шнейдерману

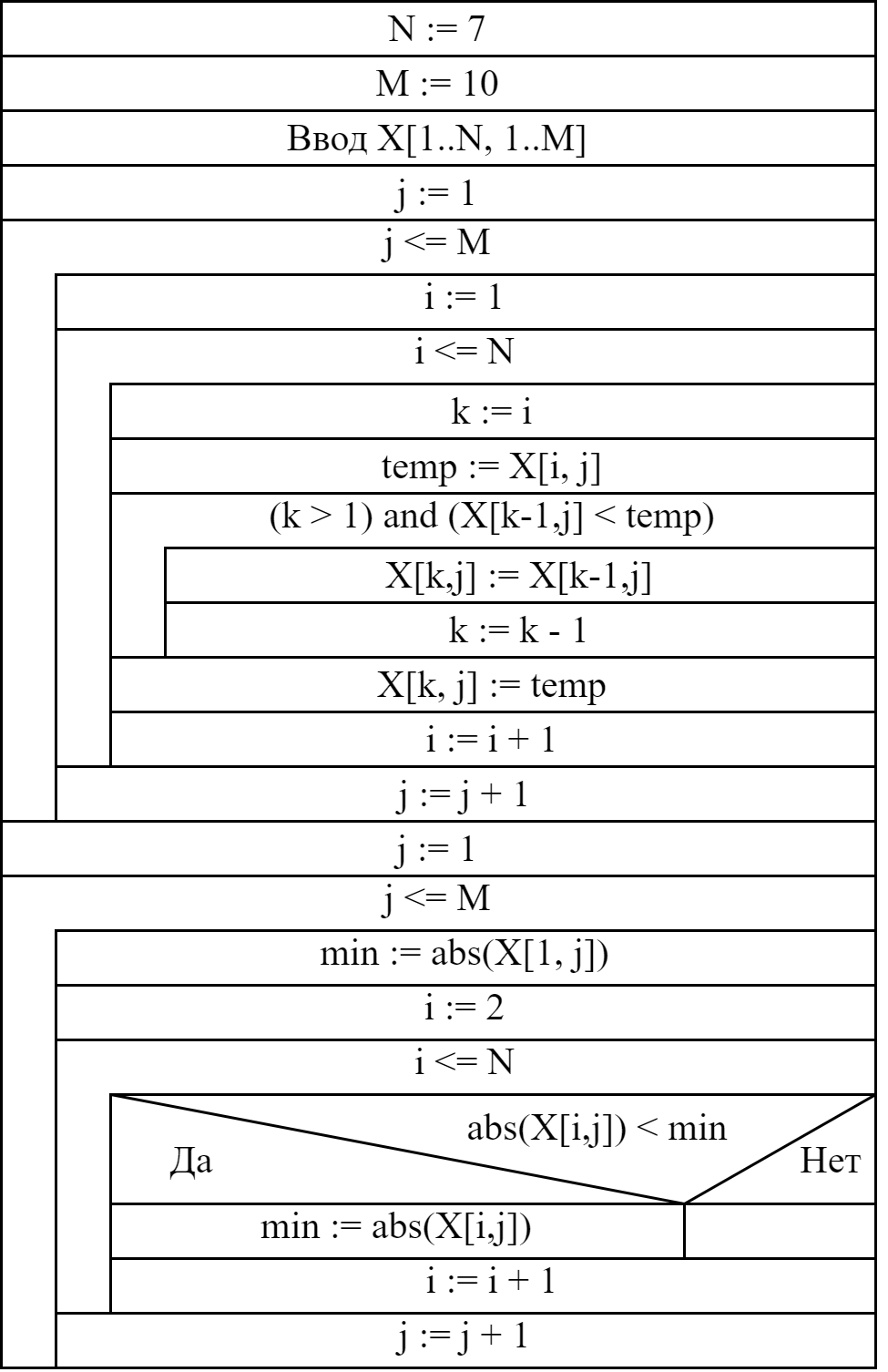


Рисунок 9 - Схема алгоритма решения задачи по методу Насси-Шнейдермана (часть 1)

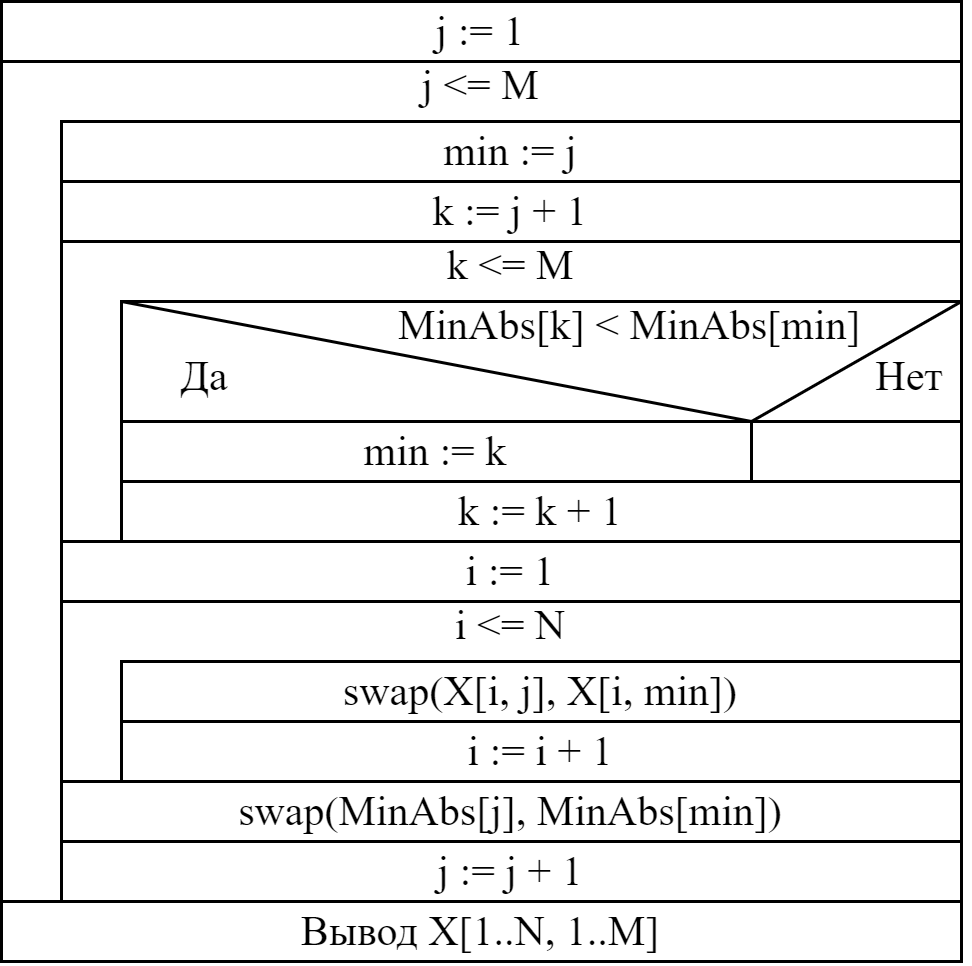


Рисунок 10 - Схема алгоритма решения задачи по методу

Насси-Шнейдермана (часть 2)