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 №3

Тема работы:

Работа с одномерными массивами

Выполнил

студент: гр. 151004 Данилов Ф.А.

Проверил: Фадеева Е.П.

Минск 2021

Содержание

[1 Постановка задачи 3](#_Toc88223740)

[2 Ход решения 4](#_Toc88223741)

[3 Текстовый алгоритм решения задачи 5](#_Toc88223742)

[4 Структура данных 7](#_Toc88223743)

[5 Схема алгоритма решения задачи по ГОСТ 19.701-90 8](#_Toc88223744)

[6 Результаты расчетов 12](#_Toc88223745)

[Приложение А 13](#_Toc88223746)

[Приложение Б 16](#_Toc88223747)

[Приложение В 17](#_Toc88223748)

[Приложение Г 20](#_Toc88223749)

[Приложение Д 23](#_Toc88223750)

# Постановка задачи

Вводится произвольная последовательность чисел . Найти частоту повторения каждого числа. Результаты расчетов поместить в одномерный массив , где каждая пара элементов () соответственно равна:  - различные элементы последовательности , а  - число их повторений. Количество пар () соответствует числу различных чисел, входящих в последовательность .

# Ход решения

Перед началом подсчета элементов массива *A* нужно выполнить следующие действия:

1. Отсортировать введённый массив *A* в порядке возрастания для эффективного подсчета элементов.
2. Заполнить массив подсчитанных значений *B* нулями для последующего инкрементирования счетчика элементов из массива *A*.

После выполнения вышестоящих действий начинается алгоритм подсчета элементов массива *A*:

Перед началом цикла считывается элемент *A1*, его значение передается в *B1*, инкрементируется *B2*. Указатель *j* ставится на ячейку *B2*, *i* – на ячейку *A2*. После выполнения этих действий начинается цикл подсчета элементов массива *A*.

Каждую итерацию цикла считывается элемент *Ai* и сравниватся с элементом *Ai-1*. В случае, если элементы совпадают, значение *Bj* инкрементируется, иначе *Ai* добавляется в *Bj+1*, *j* инкрементируетсятся на 2, *Bj* – на 1.

Пример:

A = (4, 3, 5, 6, 4, 1, 3)

После сортировки массива получится:

A = (1, 3, 3, 4, 4, 5, 6)

1. Значение элемента *A1* добавляется в *B1*.Инкрементируется значение элемента *B2*.

A = (**1**, 3, 3, 4, 4, 5, 6)

B = (1, 1, …)

1. Сравнивается значение *A2* с *A1*. Так как они не равны, добавляется новый элемент *B3*, равный элементу *A2*. Инкрементируется значение элемента *B4*.

A = (1, **3**, 3, 4, 4, 5, 6)

B = (1, 1, 3, 1, …)

1. Сравнивается значение *A3* с *A2*. Так как они равны, инкрементируется *B4*

A = (1, 3, **3**, 4, 4, 5, 6)

B = (1, 1, 3, 2, …)

После выполнения всех операций получится массив

B = (1, 1, 3, 2, 4, 2, 5, 1, 6, 1), где *Bi*– значение элемента из массива A, а *Bi+1*– количество таких элементов

# Текстовый алгоритм решения задачи

Таблица 1 – Алгоритм решения

|  |  |
| --- | --- |
| Номер  шага | Назначение шага |
|  | i := 1 |
|  | Начало цикла А1. Проверка выполнения условия (i ≤ M). Если условие истинно, идти к шагу 3, иначе – к шагу 6 |
|  | Ввод элемента массива A[i] |
|  | i := i + 1 |
|  | Конец цикла А1. Идти к шагу 2 |
|  | i := 1 |
|  | Начало цикла А2. Проверка выполнения условия (i ≤ 2\*M).  Если условие истинно, идти к шагу 8, иначе – к шагу 11 |
|  | B[i] := 0 |
|  | i := i + 1 |
|  | Конец цикла А2. Идти к шагу 7 |
|  | i := 1 |
|  | Начало цикла А3. Проверка выполнения условия (i ≤ M).  Если условие истинно, идти к шагу 13, иначе – к шагу 23 |
|  | j := 1 |
|  | Начало цикла А4. Проверка выполнения условия (j ≤ M - i).  Если условие истинно, идти к шагу 15, иначе – к шагу 21 |
|  | Проверка выполнения условия A[j] > A[j+1]. Если условие истинно, идти к шагу 16, иначе – к шагу 19 |
|  | A[j] := A[j] xor A[j+1] |
|  | A[j+1] := A[j] xor A[j+1] |
|  | A[j] := A[j] xor A[j+1] |
|  | j := j + 1 |
|  | Конец цикла А4. Идти к шагу 14 |
|  | i := i + 1 |
|  | Конец цикла А3. Идти к шагу 12 |
|  | B[1] := A[1] |
|  | B[2] := 1 |
|  | N := 2 |
|  | i := 2 |
|  | j := 2 |
|  | Начало цикла А5. Проверка выполнения условия (i ≤ M). Если условие истинно, идти к шагу 29, иначе – к шагу 37 |
|  | Проверка выполнения условия A[i] = A[i-1]. Если условие истинно, идти к шагу 30, иначе – к шагу 31 |
|  | B[j] := B[j] + 1 |
|  | B[j+1] := A[i] |

Продолжение таблицы 1

|  |  |
| --- | --- |
| 32 | j := j + 2 |
| 33 | B[j] := B[j] + 1 |
| 34 | N := N + 2 |
| 35 | i := i + 1 |
| 36 | Конец цикла А5. Идти к шагу 28 |
| 37 | i := 1 |
| 38 | Начало цикла А6. Проверка выполнения условия (i ≤ N). Если условие истинно, идти к шагу 39, иначе – к шагу 42 |
| 39 | Вывод элемента B[i] |
| 40 | i := i + 1 |
| 41 | Конец цикла А6. Идти к шагу 38 |
| 42 | Останов. |

# Структура данных

Таблица 3 – Данные

|  |  |  |
| --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение |
| M | Integer | Размер первоначального массива чисел *A* |
| N | Integer | Размер конечного массива чисел *B* |
| A | Array[1..M] of Integer | Первоначальный массив чисел |
| B | Array[1..N] of Integer | Конечный массив чисел |
| i, j | Integer | Счетчики циклов |

# Схема алгоритма решения задачи по ГОСТ 19.701-90

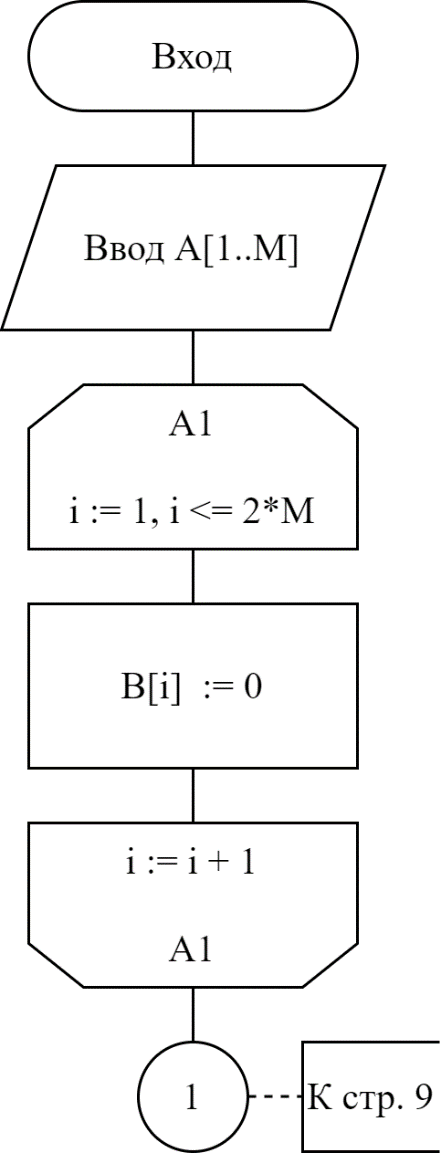


Рисунок 1 – Схема алгоритма решения задачи по ГОСТ 19.701-90 (часть 1)

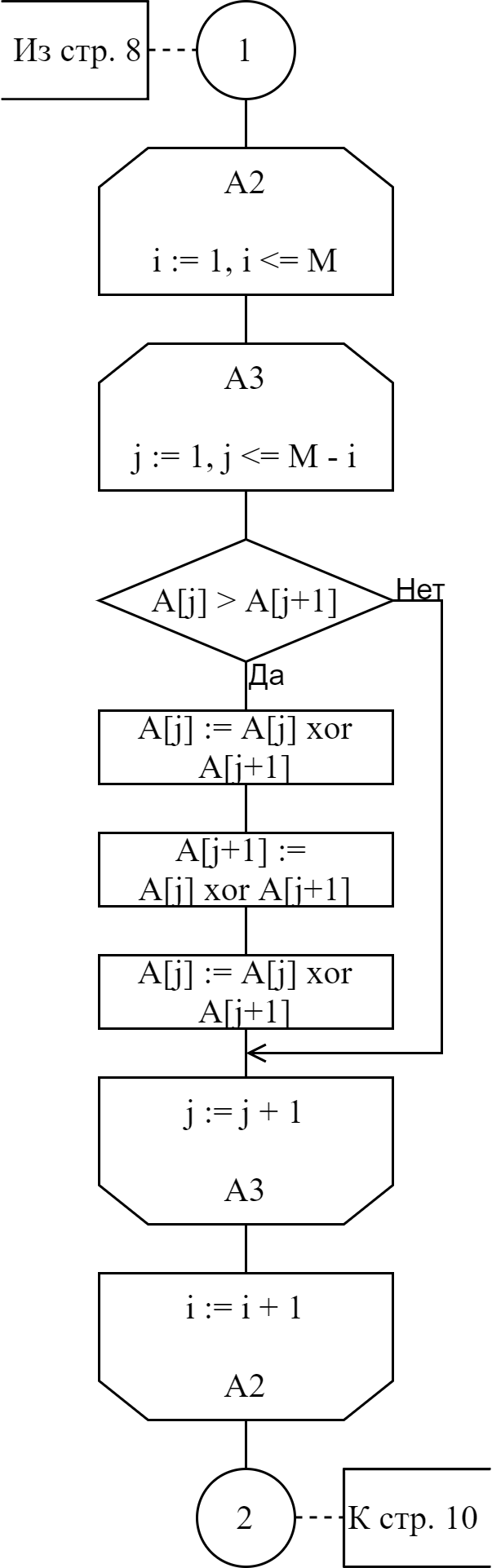


Рисунок 2 – Схема алгоритма решения задачи по ГОСТ 19.701-90 (часть 2)

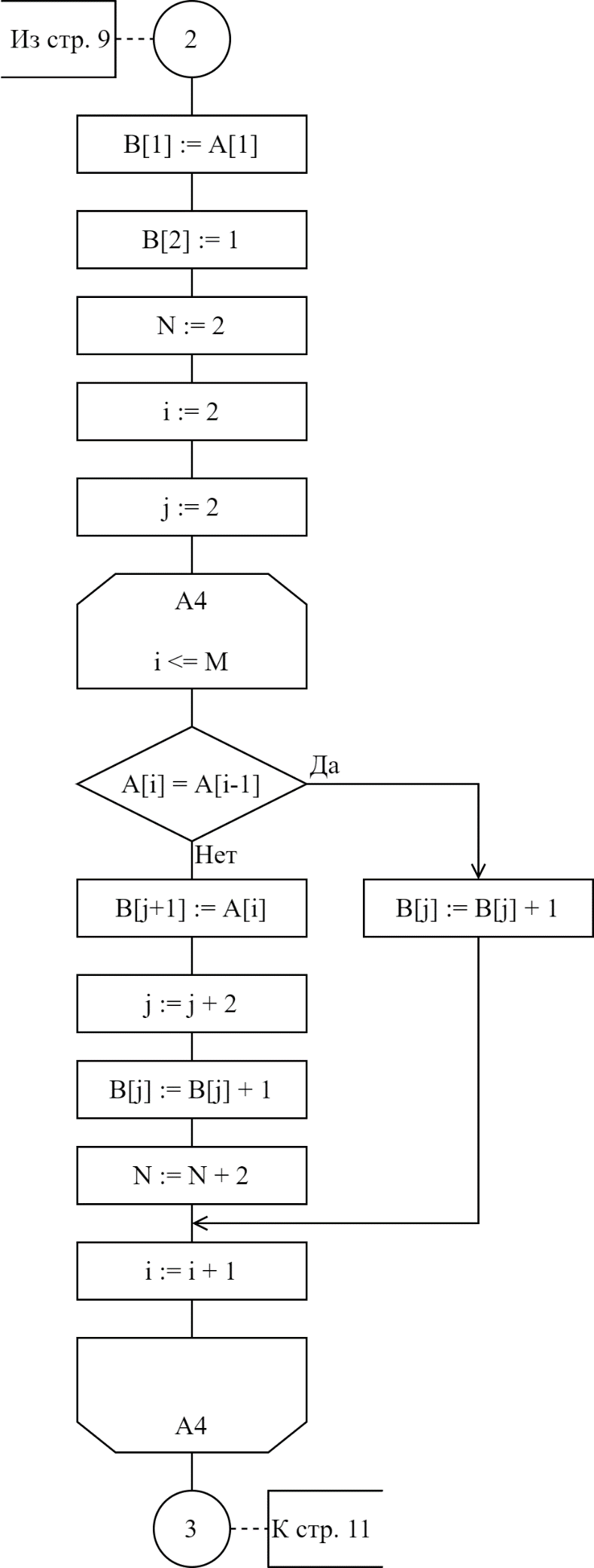


Рисунок 3 – Схема алгоритма решения задачи по ГОСТ 19.701-90 (часть 3)

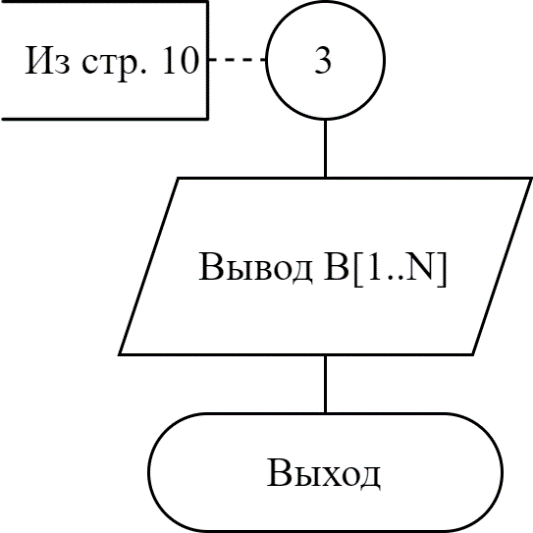


Рисунок 4 – Схема алгоритма решения задачи по ГОСТ 19.701-90 (часть 4)

# Результаты расчетов

Вследствие выполнения программы на экран выводятся следующие результаты:

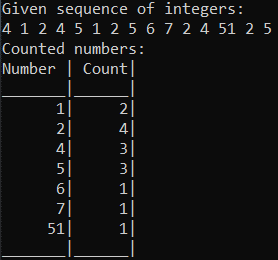


Рисунок 5 – Результаты расчётов

Приложение А

(обязательное)

Исходный код программы

program Lab3\_1;

{

Given array A, count all of it's elements.

Input: sequence of elements

Output: number same elements in given sequence

}

{$APPTYPE CONSOLE}

{$R \*.res}

uses

System.SysUtils;

// Size of array A

const

M = 15;

// i, j - counts for array

// Error - error value for val method

// InputStr - string to input values

// N - number of elements in array B

// A - array of numbers

// B - array of counted numbers from array A

var

i, j, N, Error: Integer;

InputStr: String;

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

B: Array [1..2\*M] of Integer;

begin

// Input array A size M using val method as such:

// In a loop from 1 to M do following:

// Input a string named InputStr

// Convert that string to integer A[i] using method val

// If conversion fails, repeat

writeln('Enter a sequence of numbers length ', M, ':');

for i := 1 to M do

repeat

readln(InputStr);

Val(InputStr, A[i], Error);

until Error = 0;

writeln;

// Output given array

writeln('Given sequence of integers:');

for i := 1 to M do

write(A[i], ' ');

writeln;

// Fill array B with 0

for i := 1 to 2\*M do

B[i] := 0;

// Sort Array using Bubble Sort algorithm

for i := 1 to M do

begin

for j := 1 to M - i do

begin

if A[j] > A[j+1] then

begin

// Swap A[j] and A[j+1] using swap xor algorithm

A[j] := A[j] xor A[j+1];

A[j+1] := A[j] xor A[j+1];

A[j] := A[j] xor A[j+1];

end;

end;

end;

// The main algorithm

// Initialize first elements of B and set the indexes

// to needed positions to start for loop

B[1] := A[1];

B[2] := 1;

N := 2;

i := 2;

j := 2;

while i <= M do

begin

if A[i] = A[i-1] then

begin

// Increment count of A[i] element in B

B[j] := B[j] + 1;

end

else

begin

// Add new element to array B

B[j+1] := A[i];

// Move cursor j to position for A[i] to

// count and increment B[j]

j := j + 2;

B[j] := B[j] + 1;

// Change size of array B by 2

N := N + 2;

end;

i := i + 1;

end;

// Output B

writeln('Counted numbers:');

writeln('Number | Count|');

writeln('\_\_\_\_\_\_\_|\_\_\_\_\_\_|');

for i := 1 to N do

begin

if i mod 2 <> 0 then

begin

write(B[i]:7, '|');

end

else

begin

writeln(B[i]:6, '|');

end;

end;

writeln('\_\_\_\_\_\_\_|\_\_\_\_\_\_|');

// Stop console from closing

readln;

end.

Приложение Б

(обязательное)

Тестовые наборы

Тестовые наборы для проверки вычислений

Тест 1

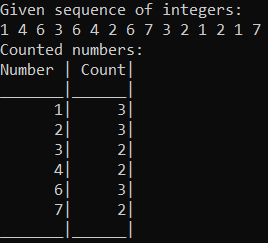
Исходные данные:

A = (1, 4, 6, 3, 6, 4, 2, 6, 7, 3, 2, 1, 2, 1, 7)

Ожидаемый результат:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Num | 1 | 2 | 3 | 4 | 6 | 7 |
| Count | 3 | 3 | 2 | 2 | 3 | 2 |

Полученный результат:



Тест 2

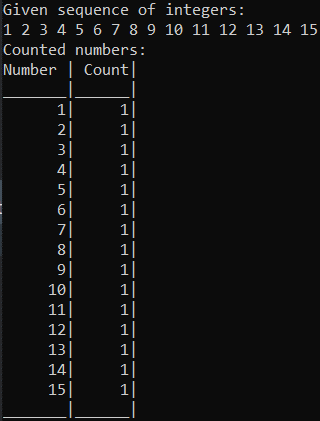
Исходные данные:

A = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15)

Ожидаемый результат:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Num | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| Count | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Полученный результат:



Приложение В

(обязательное)

Дополнительный код программы №1

program Lab3\_2;

{

Given random generated array A, count all of it's elements.

Input: sequence of elements

Output: number same elements in given sequence

}

{$APPTYPE CONSOLE}

{$R \*.res}

uses

System.SysUtils;

// Size of array A

const

M = 15;

// i, j - counts for array

// N - number of elements in array B

// A - array of numbers

// B - array of counted numbers from array A

var

i, j, N: Integer;

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

B: Array [1..2\*M] of Integer;

begin

// Generate array A using following methods:

// Firstly randomize seed for random method

// Then run a loop from 1 to M and do following:

// Randomize A[i] in scope [1..10]

// Output A[i] for user to see

randomize;

writeln('Random generated array:');

for i := 1 to M do

begin

A[i] := random(10) + 1;

write(A[i], ' ');

end;

writeln;

// Fill array B with 0

for i := 1 to 2\*M do

B[i] := 0;

// Sort Array using Bubble sort algorithm

for i := 1 to M do

begin

for j := 1 to M - i do

begin

if A[j] > A[j+1] then

begin

// Swap A[j] and A[j+1] using swap xor algorithm

A[j] := A[j] xor A[j+1];

A[j+1] := A[j] xor A[j+1];

A[j] := A[j] xor A[j+1];

end;

end;

end;

// The main algorithm

// Initialize first elements of B and set the indexes

// to needed positions to start for loop

B[1] := A[1];

B[2] := 1;

N := 2;

i := 2;

j := 2;

while i <= M do

begin

if A[i] = A[i-1] then

begin

// Increment count of A[i] element in B

B[j] := B[j] + 1;

end

else

begin

// Add new element to array B

B[j+1] := A[i];

// Move cursor j to position for A[i] to count

// and increment B[j]

j := j + 2;

B[j] := B[j] + 1;

// Change size of array B by 2

N := N + 2;

end;

i := i + 1;

end;

// Output B

writeln('Counted numbers:');

writeln('Number | Count|');

writeln('\_\_\_\_\_\_\_|\_\_\_\_\_\_|');

for i := 1 to N do

begin

if i mod 2 <> 0 then

begin

write(B[i]:7, '|');

end

else

begin

writeln(B[i]:6, '|');

end;

end;

writeln('\_\_\_\_\_\_\_|\_\_\_\_\_\_|');

// Stop console from closing

readln;

end.

Приложение Г

(обязательное)

Дополнительный код программы №2

program Lab3\_3;

{

Given array A from const Arr, count all of it's elements.

Output: number of same elements in given sequence

}

{$APPTYPE CONSOLE}

{$R \*.res}

uses

System.SysUtils;

// Size of array A

const

M = 15;

Arr: Array [1..M] of Integer = (1,4,5,1,4,6,7,8,3,1,5,6,1,1,4);

// i, j - counts for array

// N - number of elements in array B

// A - array of numbers

// B - array of counted numbers from array A

var

i, j, N: Integer;

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

B: Array [1..2\*M] of Integer;

begin

// Input A from const Arr

for i := 1 to M do

A[i] := Arr[i];

// Fill array B with 0

for i := 1 to 2\*M do

B[i] := 0;

// Sort Array using Bubble sort algorithm

for i := 1 to M do

begin

for j := 1 to M - i do

begin

if A[j] > A[j+1] then

begin

// Swap A[j] and A[j+1] using swap xor algorithm

A[j] := A[j] xor A[j+1];

A[j+1] := A[j] xor A[j+1];

A[j] := A[j] xor A[j+1];

end;

end;

end;

// The main algorithm

// Initialize first elements of B and set the indexes

// to needed positions to start for loop

B[1] := A[1];

B[2] := 1;

N := 2;

i := 2;

j := 2;

while i <= M do

begin

if A[i] = A[i-1] then

begin

// Increment count of A[i] element in B

B[j] := B[j] + 1;

end

else

begin

// Add new element to array B

B[j+1] := A[i];

// Move cursor j to position for A[i] to

// count and increment B[j]

j := j + 2;

B[j] := B[j] + 1;

// Change size of array B by 2

N := N + 2;

end;

i := i + 1;

end;

// Output B

writeln('Counted numbers:');

writeln('Number | Count|');

writeln('\_\_\_\_\_\_\_|\_\_\_\_\_\_|');

for i := 1 to N do

begin

if i mod 2 <> 0 then

begin

write(B[i]:7, '|');

end

else

begin

writeln(B[i]:6, '|');

end;

end;

writeln('\_\_\_\_\_\_\_|\_\_\_\_\_\_|');

// Stop console from closing

readln;

end.

Приложение Д

(обязательное)

Дополнительный код программы №3

program Lab3\_4;

{

Given array A, count all of it's elements.

Output: number same elements in given sequence

}

{$APPTYPE CONSOLE}

{$R \*.res}

uses

System.SysUtils;

// M - Size of array A

const

M = 15;

// i, j - counts for array

// N - number of elements in array B

// A - array of numbers

// B - array of counted numbers from array A

var

i, j, N: Integer;

A: Array [1..M] of Integer = (1,5,2,6,7,1,2,5,6,7,2,2,6,4,2);

B: Array [1..2\*M] of Integer;

begin

// Fill array B with 0

for i := 1 to 2\*M do

B[i] := 0;

// Sort Array using Bubble sort algorithm

for i := 1 to M do

begin

for j := 1 to M - i do

begin

if A[j] > A[j+1] then

begin

// Swap A[j] and A[j+1] using swap xor algorithm

A[j] := A[j] xor A[j+1];

A[j+1] := A[j] xor A[j+1];

A[j] := A[j] xor A[j+1];

end;

end;

end;

// The main algorithm

// Initialize first elements of B and set the indexes

// to needed positions to start for loop

B[1] := A[1];

B[2] := 1;

N := 2;

i := 2;

j := 2;

while i <= M do

begin

if A[i] = A[i-1] then

begin

// Increment count of A[i] element in B

B[j] := B[j] + 1;

end

else

begin

// Add new element to array B

B[j+1] := A[i];

// Move cursor j to position for A[i] to

// count and increment B[j]

j := j + 2;

B[j] := B[j] + 1;

// Change size of array B by 2

N := N + 2;

end;

i := i + 1;

end;

// Output B

writeln('Counted numbers:');

writeln('Number | Count|');

writeln('\_\_\_\_\_\_\_|\_\_\_\_\_\_|');

for i := 1 to N do

begin

if i mod 2 <> 0 then

begin

write(B[i]:7, '|');

end

else

begin

writeln(B[i]:6, '|');

end;

end;

writeln('\_\_\_\_\_\_\_|\_\_\_\_\_\_|');

// Stop console from closing

readln;

end.