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: Одномерные массивы

Выполнил

студент: гр. 151003 Барановский Р.А.

Проверил: Фадеева Е.П.

Минск 2021

содержание

[1 Постановка задачи 3](#_Toc88489868)

[2 Текстовый алгоритм решения задачи 4](#_Toc88489869)

[3 Структура данных 6](#_Toc88489870)

[4 Схема алгоритма решения задачи по Гост 19.701-90 7](#_Toc88489871)

[5 Результаты расчетов 11](#_Toc88489872)

[Приложение А1 12](#_Toc88489873)

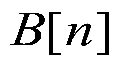
[Приложение А2 15](#_Toc88489874)

[Приложение А3 18](#_Toc88489875)

[Приложение А4 21](#_Toc88489876)

[Приложение Б 24](#_Toc88489877)

# Постановка задачи

 Из массива  исключить все элементы, у которых есть равные, оставив при этом последние вхождения элементов.

# Текстовый алгоритм решения задачи

Таблица 1 – Алгоритм решения

|  |  |
| --- | --- |
| Номер  Шага | Назначение шага |
| 1 | N := 10 |
| 2 | i := 1 |
| 3 | Начало цикла А1. Проверка выполнения условия i <= N. Если условие истинно, то идти к шагу 4, иначе к шагу 7 |
| 4 | Ввод B[i] |
| 5 | i := i + 1 |
| 6 | Конец цикла А1. Идти к шагу 3 |
| 7 | i := 1 |
| 8 | Начало цикла А2. Проверка выполнения условия i <= N. Если условие истинно, то идти к шагу 9, иначе к шагу 12 |
| 9 | Вывод B[i] |
| 10 | i := i + 1 |
| 11 | Конец цикла А2. Идти к шагу 8 |
| 12 | Size := N |
| 13 | i := Size + 1 |
| 14 | Rep := false |
| 15 | Начало цикла А3. Проверка выполнения условия i >= 2. Если условие истинно, то идти к шагу 16, иначе к шагу 34 |
| 16 | Проверка выполнения условия Rep = false. Если условие истинно, то идти к шагу 17, иначе к шагу 18 |
| 17 | dec(i) |
| 18 | Rep := false |
| 19 | j := i |
| 20 | Начало цикла А4. Проверка выполнения условия j >= 2. Если условие истинно, то идти к шагу 21, иначе к шагу 33 |
| 21 | dec(j) |
| 22 | Проверка выполнения условия B[i] = B[j]. Если условие истинно, то идти к шагу 23, иначе к шагу 32 |
| 23 | k := j |
| 24 | Начало цикла А5. Проверка выполнения условия k <= Size - 1. Если условие истинно, то идти к шагу 25, иначе к шагу 28 |
| 25 | B[k] := B[k + 1] |
| 26 | k := k + 1 |
| 27 | Конец цикла А5. Идти к шагу 24 |
| 28 | dec(Size) |

Продолжение таблицы 1

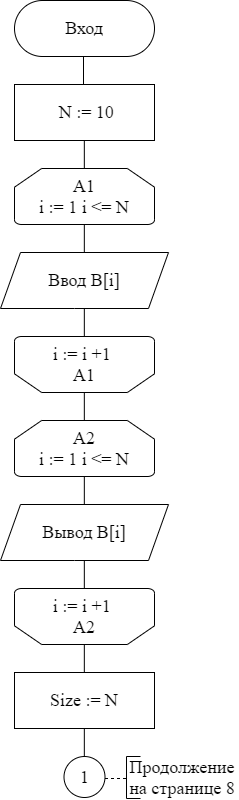
|  |  |
| --- | --- |
| 29 | dec(i) |
| 30 | Rep := true |
| 31 | j := i |
| 32 | Конец цикла А4. Идти к шагу 20 |
| 33 | Конец цикла А3. Идти к шагу 15 |
| 34 | i := 1 |
| 35 | Начало цикла А6. Проверка выполнения условия i <= Size. Если условие истинно, то идти к шагу 35, иначе к шагу 39 |
| 36 | Вывод B[i] |
| 37 | i := i + 1 |
| 38 | Конец цикла А6. Идти к шагу 35 |
| 39 | Останов. |

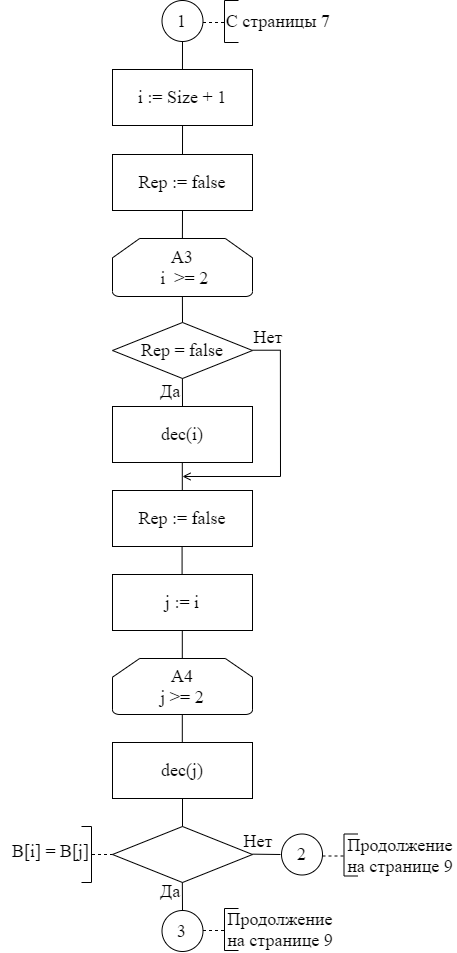
# Структура данных

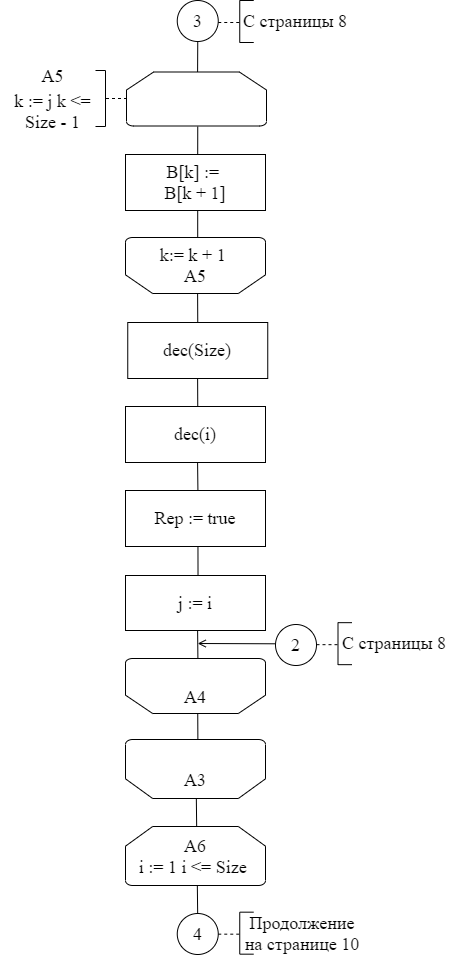
Таблица 2 - Данные

|  |  |  |
| --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение |
| N | Const | Размер массива |
| Mas | Array of real | Массив |
| MasStr | Array of String | Проверка ошибок ввода |
| B | Mas | Массив |
| BStr | MasStr | Проверка ошибок ввода |
| I | Integer | Счетчик для сравнивания элементов 1 |
| J | Integer | Счетчик для сравнивания элементов 2 |
| K | Integer | Счетчик для передвижения элементов в массиве |
| Error | Integer | Переменная, принимающая значение < > 0, если пользователь допустил ошибку при вводе. 0, если ошибок допущено не было |
| Size | Integer | Конечное количество выводимых элементов массива |
| Rep | Boolean | Переменная, принимающая значение true, если был найден повторяющийся элемент, false в противном случае |

# Схема алгоритма решения задачи по Гост 19.701-90







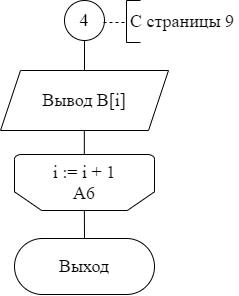


Рисунок 1 - Схема алгоритма решения задачи по ГОСТ 19.701-90

# Результаты расчетов

Вследствие результатов программы на экран выводятся следующие результаты расчетов:

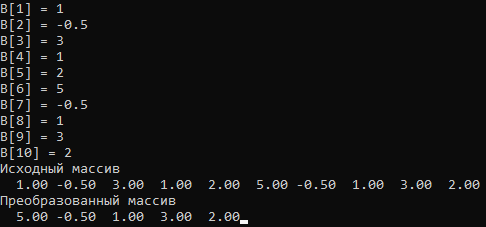


Рисунок 2 – Результаты расчетов 1 программы

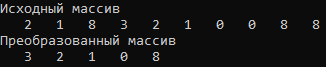


Рисунок 3 – Результаты расчетов 2 программы

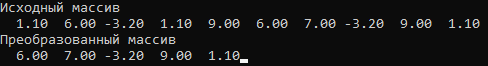


Рисунок 4 – Результаты расчетов 3 программы

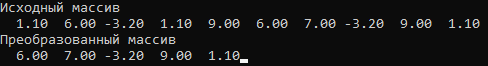


Рисунок 5 - Результаты расчетов 4 программы

Приложение А1

(обязательное)

Исходный код программы

Program Lab3Exp1;

//Program exclude repeating elements

//Console app

{$APPTYPE CONSOLE}

//Modules declaration

Uses

System.SysUtils;

//Constants declaration

Const

N = 10;

//Types declaration

Type

Mas = array[1..N] of real;

MasStr = array [1..N] of String;

//Variables declaration

Var

B:Mas;

BStr:MasStr;

i, j, k, Size, Error: integer;

Rep: boolean;

{ B - array

BStr - Mistakes checker

i - counter for compare 1

j - counter for compare 2

k - counter fot moving elements of array

Size - current size of array

Error - variable which is <> 0 if user made a mistake

and 0 if not

Rep – vatiable which is used to avoid double decrement }

//Start the program

Begin

//Set Error to the original value

Error := 1;

//Inputting array elements

for i:=1 to N do

begin

//Protection from mistakes

repeat

write('B[',i,'] = ');

readln(BStr[i]);

val(BStr[i], B[i], Error);

//If mistake happened warn about it

if Error <> 0 then

writeln('Пожалуйста, введите вещественное число');

//If Error = <> 0 repeat the cycle

until Error = 0;

end;

//Output original array

writeln('Исходный массив');

for i:=1 to N do

write(B[i]:6:2);

//Remember the original size of array

Size:=N;

//Set i to the original value

i:=Size+1;

//Set Rep to the original value

Rep:=false;

//Entering the cycle

while i>=2 do

begin

//In order to avoid double decrement

if Rep = false then

dec(i); (\*Decrement i\*)

//Set Rep to false

Rep:=false

//Make j equals to i

j:=i;

//Entering the cycle

while j>=2 do

begin

//Decrement of j

dec(j);

//If repeated element was found

if B[i]=B[j] then

begin

for k:=j to Size - 1 do

//Move elements to the right

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

//Decrease size of array

dec(Size);

//Position of checking element has changed

dec(i);

//In order to avoid double decrement of i

Rep := true;

//If number repeated more than once

j := i;

end;

end;

end;

writeln;

//Output new array

writeln('Преобразованный массив');

for i:=1 to Size do

write(B[i]:6:2);

//Final operations

readln;

End.

Приложение А2

(обязательное)

Исходный код программы

Program Lab3Exp2;

//Program exclude repeating elements

//Console app

{$APPTYPE CONSOLE}

//Modules declaration

Uses

System.SysUtils;

//Constants declaration

Const

N = 10;

//Types declaration

Type

Mas = array[1..N] of integer;

//Variables declaration

Var

B: Mas;

i, j, k, Size: integer;

Rep: boolean;

{ B - array

i - counter for compare 1

j - counter for compare 2

k - counter for moving elements of array

Size - current size of array

Rep – variable which is used to avoid double decrement }

//Start the program

Begin

//In order to create new random numbers

Randomize;

//Generate random elements of array

for i := 1 to N do

B[i]:=Random(10);

//Output original array

writeln('Исходный массив');

for i:=1 to N do

write(B[i]:4);

//Remember the original size of array

Size:=N;

//Set i to the original value

i:=Size+1;

//Set Rep to the original value

Rep := false;

//Entering the cycle

while i>=2 do

begin

//In order to avoid double decrement

if Rep = false then

dec(i); (\*Decrement i\*)

//Set Rep to false

Rep:=false

//Make j equals to i

j:=i;

//Entering the cycle

while j>=2 do

begin

//Increment of j

dec(j);

//If repeated element was found

if B[i]=B[j] then

begin

for k:=j to Size - 1 do

//Move elements to the right

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

//Decrease size of array

dec(Size);

//Position of checking element has changed

dec(i);

//In order to avoid double decrement of i

Rep := true;

//If number repeated more than once

j := i;

end;

end;

end;

writeln;

//Output new array

writeln('Преобразованный массив');

for i:=1 to Size do

write(B[i]:6:2);

//Final operations

readln;

End.

Приложение А3

(обязательное)

Исходный код программы

Program Lab3Exp3;

//Program exclude repeating elements

//Console app

{$APPTYPE CONSOLE}

//Modules declaration

Uses

System.SysUtils;

//Constants declaration

Const

N = 10;

Mas:array[1..N] of real = (1.1,6,-3.2,1.1,9,6,7,

-3.2,9,1.1);

//Variables declaration

Var

B:array[1..N] of real;

i, j, k, Size: integer;

Rep: boolean;

{ B - array

i – counter for compare 1

j – counter for compare 2

k - counter for moving elements of array

Size - current size of array

Rep – variable which is used to avoid double decrement }

//Start the program

Begin

//Output original array

writeln('Исходный массив');

for i:=1 to N do

write(Mas[i]:6:2);

//Fill array B

for i:=1 to N do

B[i] := Mas[i];

//Remember the original size of array

Size:=N;

//Set i to the original value

i:=Size+1;

//Set Rep to the original value

Rep := false;

//Entering the cycle

while i>=2 do

begin

//In order to avoid double decrement

if Rep = false then

dec(i); (\*Decrement i\*)

//Set Rep to false

Rep:=false

//Make j equals to i

j:=i;

//Entering the cycle

while j>=2 do

begin

//Increment of j

dec(j);

//If repeated element was found

if B[i]=B[j] then

begin

for k:=j to Size - 1 do

//Move elements to the right

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

//Decrease size of array

dec(Size);

//Position of checking element has changed

dec(i);

//In order to avoid double decrement of i

Rep := true;

//If number repeated more than once

j := i;

end;

end;

end;

writeln;

//Output new array

writeln('Преобразованный массив');

for i:=1 to Size do

write(B[i]:6:2);

//Final operations

readln;

End.

Приложение А4

(обязательное)

Исходный код программы

Program Lab3Exp4;

//Program exclude repeating elements

//Console app

{$APPTYPE CONSOLE}

//Modules declaration

Uses

System.SysUtils;

//Constants declaration

Const

N = 10;

//Variables declaration

Var

B:array[1..N] of real = (1.1,6,-3.2,1.1,9,6,7,

- 3.2,9,1.1);

i, j, k, Size: integer;

Rep: Boolean;

{ B - array

i – counter for compare 1

j – counter for compare 2

k – counter for moving elements of array

Size - current size of array

Rep – variable which is used to avoid }

//Start the program

Begin

//Output original array

writeln('Исходный массив');

for i:=1 to N do

write(B[i]:6:2);

//Remember the original size of array

Size:=N;

//Set i to the original value

i:=Size+1;

//Set Rep to the original value

Rep := false;

//Entering the cycle

while i>=2 do

begin

//In order to avoid double decrement

if Rep = false then

dec(i); (\*Decrement i\*)

//Set Rep to false

Rep:=false

//Make j equals to i

j:=i;

//Entering the cycle

while j>=2 do

begin

//Increment of j

dec(j);

//If repeated element was found

if B[i]=B[j] then

begin

for k:=j to Size - 1 do

//Move elements to the right

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

//Decrease size of array

dec(Size);

//Position of checking element has changed

dec(i);

//In order to avoid double decrement of i

Rep := true;

//If number repeated more than once

j := i;

end;

end;

end;

writeln;

//Output new array

writeln('Преобразованный массив');

for i:=1 to Size do

write(B[i]:6:2);

//Final operations

readln;

End.

Приложение Б

(обязательное)

Тестовые наборы(расчеты)

Тест 1

Тестовая ситуация: Элементы массива – одно и то же число

Исходные данные: B[1] = 1

B[2] = 1

B[3] = 1

B[4] = 1

B[5] = 1

B[6] = 1

B[7] = 1

B[8] = 1

B[9] = 1

B[10] = 1

Ожидаемый результат:

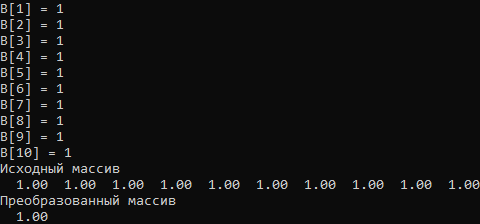
Исходный массив

1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00

Преобразованный массив

1.00

Полученный результат:



Тест 2

Тестовая ситуация: Элементы массива – разные числа

Исходные данные: B[1] = 1

B[2] = 1.11

B[3] = 2

B[4] = 2.11

B[5] = 3

B[6] = 3.11

B[7] = 4

B[8] = 4.44

B[9] = 5

B[10] = 5.55

Ожидаемый результат:

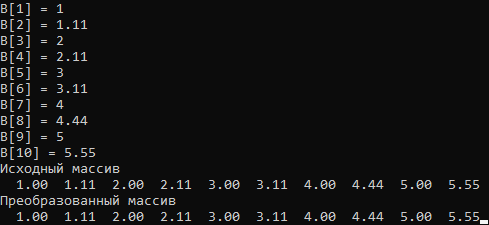
Исходный массив

1.00 1.11 2.00 2.11 3.00 3.11 4.00 4.44 5.00 5.55

Преобразованный массив

1.00 1.11 2.00 2.11 3.00 3.11 4.00 4.44 5.00 5.55

Полученный результат:



Тест 3

Тестовая ситуация: Среди элементов массива есть одинаковые и

неодинаковые числа

Исходные данные: B[1] = 1

B[2] = 9

B[3] = 4.44

B[4] = 1

B[5] = 1

B[6] = 9

B[7] = 5

B[8] = 1

B[9] = 1

B[10] = 6

Ожидаемый результат:

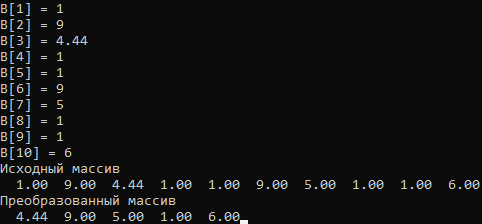
Исходный массив

1.00 9.00 4.44 1.00 1.00 9.00 5.00 1.00 1.00 6.00

Преобразованный массив

4.44 9.00 5.00 1.00 6.00

Полученный результат:



Тестовые наборы(Проверка ввода)

Тест 4

Тестовая ситуация: Ввод букв в поле для чисел

Исходные данные: B[1] = ‘hooray’

Ожидаемый результат:

Пожалуйста, введите вещественное число

Полученный результат:

