**УО «Белорусский государственный университет информатики и**

**радиоэлектроники»**

**Кафедра ПОИТ**

**Отчет по лабораторной работе №2.4**

**по предмету**

**Основы Алгоритмизации и Программирования**

**Вариант 1**

**Выполнил**

**Андросов И.С.**

**Проверила**

**Данилова Г.В.**

Группа:

**8**51001

**Минск 2018**

**Задание**

Даны два упорядоченных по возрастанию массива целых чисел: А, состоящего из n-элементов, и В, состоящего из m-элементов. Выполнить слияние этих двух массивов в один упорядоченный массив C (без повторений).

**Код программы**

**(Delphi)**

**program** laba\_2\_4;

{$APPTYPE CONSOLE}

**uses**

SysUtils;

**type**

TShortIntArray = **array of** ShortInt;

**const**

MinNumber: ShortInt = -128;

MaxNumber: ShortInt = 127;

MinSize: ShortInt = 2;

MaxSize: ShortInt = 127;

**function** CheckNumbersOfArray(MyArray: TShortIntArray): Boolean;

**var**

IsCorrect: Boolean;

i: ShortInt;

**begin**

IsCorrect := false;

i := 0;

**repeat**

**if** MyArray[i] = MyArray[i + 1] **then**

IsCorrect := true;

Inc(i);

**until** (IsCorrect) **or** (i = High(MyArray));

CheckNumbersOfArray := IsCorrect;

**end**;

**procedure** GetOutputToConsole(MyArray: TShortIntArray);

**var**

i: ShortInt;

**begin**

**for** i := 0 **to** High(MyArray) **do**

Write(MyArray[i],' ');

WriteLn;

**end**;

**function** GetAnswer(): Char;

**var**

Answer: Char;

IsCorrect: Boolean;

**begin**

**repeat**

ReadLn(Answer);

Answer := UpCase(Answer);

**if** (Answer = 'Y') **or** (Answer = 'N') **then**

IsCorrect := true

**else**

**begin**

IsCorrect := false;

WriteLn('Incorrect input. Enter Y(Yes) or N(No):');

**end**;

**until** IsCorrect;

GetAnswer := Answer;

**end**;

**procedure** GetOutputToFile(ArrayC: TShortIntArray);

**var**

IsCorrect: Boolean;

NewFile: TextFile;

NameOfFile: String;

i: ShortInt;

**begin**

WriteLn('Enter the name of file Name.txt:');

IsCorrect := false;

**repeat**

ReadLn(NameOfFile);

**if** (**not** FileExists(NameOfFile)) **then**

WriteLn('File does not exist. Try again:')

**else**

**begin**

IsCorrect := true;

Assign(NewFile, NameOfFile);

writeln('Would you like to rewrite the file? Press Y(Yes) or N(No):');

**if** GetAnswer = 'Y' **then**

**try**

Rewrite(NewFile);

**for** i := 0 **to** High(ArrayC) **do**

Write(NewFile, ArrayC[i],' ');

**except**

WriteLn('Access is not allowed. Try again:');

IsCorrect:= false;

**end**

**else**

**try**

Append(NewFile);

**for** i := 0 **to** High(ArrayC) **do**

Write(NewFile, ArrayC[i],' ');

**except**

WriteLn('Access is not allowed. Try again:');

IsCorrect:= false;

**end**;

**end**;

**until** IsCorrect;

Close(NewFile);

**end**;

**procedure** OrderArray(**var** MyArray: TShortIntArray);

**var**

i, j, Temp: ShortInt;

**begin**

**for** i := 0 **to** High(MyArray) **do**

**for** j := 0 **to** (High(MyArray) - i - 1) **do**

**begin**

**if** MyArray[j] > MyArray[j + 1] **then**

**begin**

temp := MyArray[j];

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

MyArray[j + 1] := temp;

**end**;

**end**;

**end**;

**function** CheckExistion(ArrayC: TShortIntArray; Num, MaxI: ShortInt): Boolean;

**var**

i: ShortInt;

IsCorrect: Boolean;

**begin**

IsCorrect := true;

i := 0;

**repeat**

**if** Num = ArrayC[i] **then**

IsCorrect := false

**else**

Inc(i);

**until** (**not** IsCorrect) **or** (i = MaxI);

CheckExistion := IsCorrect;

**end**;

**function** GetSizeOfArrayC(MyArray, ThatArray: TShortIntArray): ShortInt;

**var**

i, j, Size: ShortInt;

IsCorrect: Boolean;

**begin**

Size := 0;

**for** i := 0 **to** High(MyArray) **do**

**begin**

IsCorrect := true;

**for** j := 0 **to** High(ThatArray) **do**

**if** MyArray[i] = ThatArray[j] **then**

IsCorrect := false;

**if** IsCorrect **then**

Inc(Size, 2)

**else**

Inc(Size);

**end**;

GetSizeOfArrayC := Size + (High(ThatArray) - High(MyArray));

**end**;

**function** GetArrayC(ArrayA, ArrayB: TShortIntArray): TShortIntArray;

**var**

i, j, Size: ShortInt;

MyArray: TShortIntArray;

**begin**

Size := GetSizeOfArrayC(ArrayA, ArrayB);

SetLength(MyArray, Size);

**for** i := 0 **to** High(ArrayA) **do**

MyArray[i] := ArrayA[i];

j := 0;

**repeat**

**if** CheckExistion(MyArray, ArrayB[j], i) **then**

**begin**

MyArray[i] := ArrayB[j];

Inc(i);

**end**;

Inc(j);

**until** (i = High(MyArray) + 1);

OrderArray(MyArray);

GetArrayC := MyArray;

**end**;

**function** CheckInput(Min, Max: ShortInt): ShortInt;

**var**

IsCorrect: Boolean;

Number: ShortInt;

**begin**

IsCorrect := false;

**repeat**

**try**

ReadLn(Number);

**if** (Number >= min) **and** (Number <= max) **then**

IsCorrect := true

**else**

WriteLn('Enter number from interval ', Min, '..', Max, ':');

**except**

WriteLn('Check entered data. Enter number from interval ', Min, '..', Max,

':');

**end**;

**until** IsCorrect;

CheckInput := Number;

**end**;

**function** GetArrayConsole(): TShortIntArray;

**var**

i, Size: ShortInt;

MyArray: TShortIntArray;

**begin**

WriteLn('Enter size of array ', MinSize, '..', MaxSize, ':');

Size := CheckInput(MinSize, MaxSize);

SetLength(MyArray, Size);

Dec(Size);

**for** i := 0 **to** Size **do**

**begin**

WriteLn('Enter [', i, '] number of array ', MinNumber, '..', MaxNumber,

':');

MyArray[i] := CheckInput(MinNumber, MaxNumber);

**end**;

GetArrayConsole := MyArray;

**end**;

**function** CheckArray(MyArray: TShortIntArray): Boolean;

**var**

i, MaxI: ShortInt;

IsCorrect: Boolean;

**begin**

IsCorrect := true;

MaxI := High(MyArray) - 1;

**for** i := 0 **to** MaxI **do**

**if** IsCorrect **then**

**if** MyArray[i] > MyArray[i + 1] **then**

**begin**

IsCorrect := false;

**end**;

CheckArray := IsCorrect;

**end**;

**function** GetArray(**var** MyFile: TextFile; MyArray: TShortIntArray; **const** Size: ShortInt): TShortIntArray;

**var**

i: ShortInt;

**begin**

SetLength(MyArray, Size);

**for** i := 0 **to** High(MyArray) **do**

Read(MyFile, MyArray[i]);

GetArray := MyArray;

**end**;

**function** GetSizeOfArray(**var** MyFile: TextFile): ShortInt;

**var**

Size, Num: ShortInt;

**begin**

Size := 0;

**repeat**

Read(MyFile, Num);

Inc(Size);

**until** EoLn(MyFile);

GetSizeOfArray := Size;

**end**;

**function** CheckInputFile(**const** MinNumber, MaxNumber: ShortInt; **var** MyFile: TextFile): Boolean;

**var**

IsCorrect: boolean;

Number: ShortInt;

**begin**

IsCorrect := true;

**while** (**not** SeekEof(MyFile)) **and** (IsCorrect) **do**

**try**

ReadLn(MyFile, Number);

**if** (Number <= MinNumber) **or** (Number >= MaxNumber) **then**

IsCorrect := false;

**except**

IsCorrect := false;

**end**;

CheckInputFile := IsCorrect;

**end**;

**function** ReadFromFile(**var** MyFile: TextFile; **var** ArrayA, ArrayB: TShortIntArray): Boolean;

**var**

SizeA, SizeB: ShortInt;

**begin**

**if** CheckInputFile(MinNumber, MaxNumber, MyFile) **then**

**begin**

reset(MyFile);

SizeA := GetSizeOfArray(MyFile);

ReadLn(MyFile);

SizeB := GetSizeOfArray(MyFile);

**if** Eof(MyFile) **then**

**begin**

reset(MyFile);

ArrayA := GetArray(MyFile, ArrayA, SizeA);

ReadLn(MyFile);

ArrayB := GetArray(MyFile, ArrayB, SizeB);

reset(MyFile);

ReadFromFile := true;

**end**

**else**

**begin**

WriteLn('There must be only 2 arrays');

ReadFromFile := false;

**end**;

**end**

**else**

**begin**

CloseFile(MyFile);

ReadFromFile := false;

WriteLn('Check entered data. Enter number from interval ', MinNumber, '..',

MaxNumber, '. Try Again:');

**end**;

**end**;

**procedure** GetArraysFile(**var** ArrayA, ArrayB: TShortIntArray);

**var**

IsCorrect: Boolean;

NameOfFile: String;

MyFile: TextFile;

**begin**

WriteLn('Enter file name(Name.txt):');

**repeat**

ReadLn(NameOfFile);

**if** (**not** FileExists(NameOfFile)) **then**

**begin**

WriteLn('File does not exist. Try again:');

IsCorrect := false;

**end**

**else**

**begin**

AssignFile(MyFile, NameOfFile);

reset(MyFile);

**if** SeekEof(MyFile) **then**

**begin**

WriteLn('File is empty. Try again:');

IsCorrect := false;

**end**

**else**

IsCorrect := ReadFromFile(MyFile, ArrayA, ArrayB);

**end**;

**until** IsCorrect;

CloseFile(MyFile);

**end**;

**procedure** Main();

**var**

ArrayA, ArrayB, ArrayC: TShortIntArray;

**begin**

WriteLn('This program merges two ordered by ascending arrays A and B into one

ordered array C without repetitions.', #10, 'Would you like to open the

file or work with console? Press Y(Yes) or N(No):');

**if** GetAnswer = 'Y' **then**

GetArraysFile(ArrayA, ArrayB)

**else**

**begin**

Write('First array. ');

ArrayA := GetArrayConsole();

Write('Second array. ');

ArrayB := GetArrayConsole();

**end**;

**if** (**not** CheckArray(ArrayA)) **then**

**begin**

OrderArray(ArrayA);

WriteLn('Array A has been sorted by ascending:');

GetOutputToConsole(ArrayA);

**end**;

**if** (**not** CheckArray(ArrayB)) **then**

**begin**

OrderArray(ArrayB);

WriteLn('Array B has been sorted by ascending:');

GetOutputToConsole(ArrayB);

**end**;

**if** CheckNumbersOfArray(ArrayA) **or** CheckNumbersOfArray(ArrayB) **then**

WriteLn('Error. There should not be the same numbers.')

**else**

**begin**

ArrayC := GetArrayC(ArrayA, ArrayB);

WriteLn('Would you like to write down the answer to file? Press Y(Yes) or

N(No):');

**if** GetAnswer = 'Y' **then**

GetOutputToFile(ArrayC);

GetOutputToConsole(ArrayC);

**end**;

WriteLn('Press "Enter" to exit the console.');

ReadLn;

**end**;

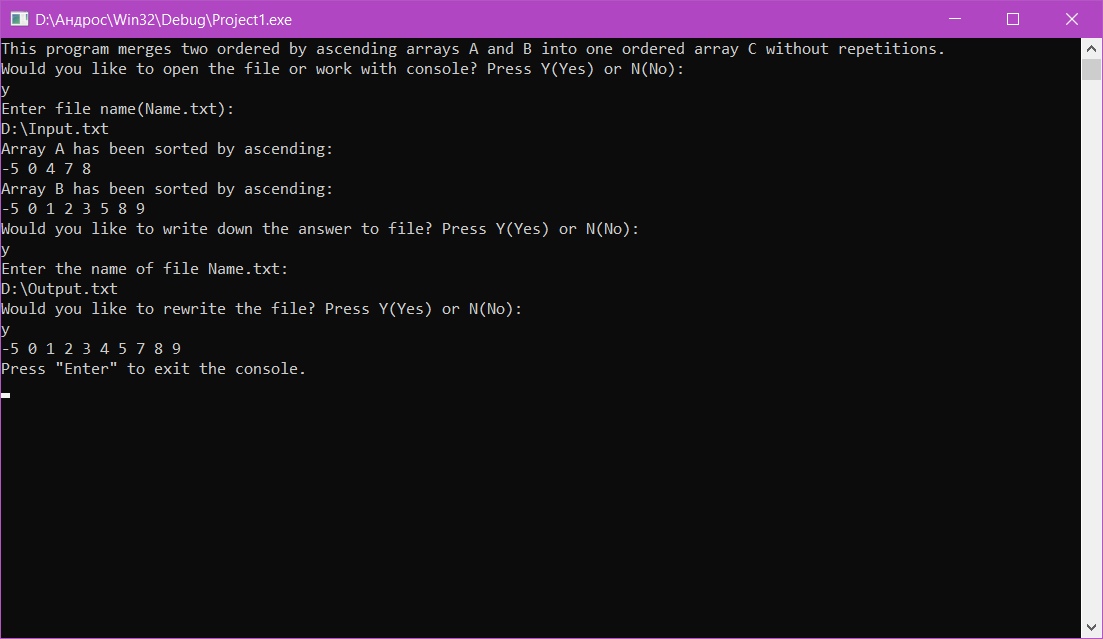
**begin**

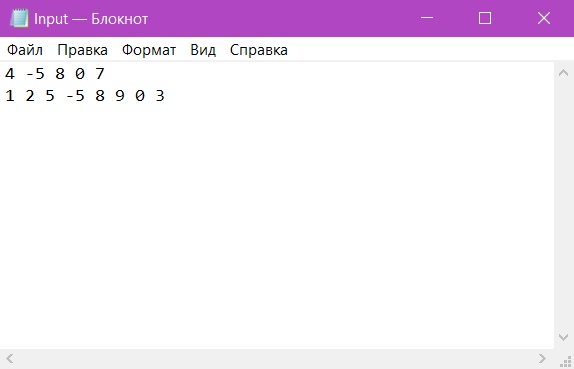
Main();

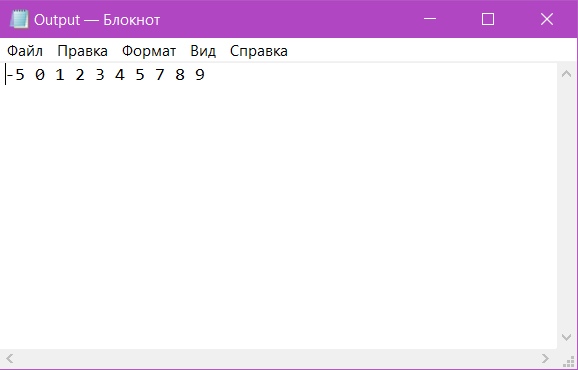
**end**.

**Скриншоты**

**(Delphi)**







**Схема алгоритма**



