**УО «Белорусский государственный университет информатики и радиоэлектроники»**

**Кафедра ПОИТ**

**Отчёт по лабораторной работе №3.2**

**По предмету**

**Основы алгоритмизации и программирования**

**Вариант 6**

**Выполнил:**

**Гладкий М.Г.**

**Проверила:**

**Данилова Г.В.**

**Группа 851001**

**Минск 2018**

**Задание:**

Рассматривается массив символьных строк, состоящий не более чем из 10 строк. Длина строки не более 40. Прочитать массив и распечатать его. Для каждой строки массива построить множество из символов этой строки. Напечатать символы того множества, которое окажется подмножеством всех остальных множеств, и порядковый номер множества. Если таких множеств найдется несколько, то напечатать первое по порядку. Если такого множества не найдется, то напечатать сообщение "Такого множества нет."

**Код Delphi 10:**

**program** Project10;

{$APPTYPE CONSOLE}

**uses**

SysUtils;

**type**

TArr = **array**[1..10] **of** String[40];

TSet = **array**[1..10] **of Set of** Char;

**const**

MinLength = 1;

MaxLength = 40;

MinNumber = 2;

MaxNumber = 10;

**function** Calculation(Number: Byte; ArrayofStrings: TArr): String;

**var**

MySets: TSet;

i, j, Num, Res: Byte;

IsCorrect: Boolean;

Answer: String[40];

**begin**

**for** i := 1 **to** Number **do**

**begin**

MySets[i] := [];

**for** j := 1 **to** Length(ArrayofStrings[i]) **do**

MySets[i] := MySets[i] + [ArrayofStrings[i][j]];

**end**;

Num := 0;

Res := 0;

IsCorrect := true;

**while** (Num < Number)**and**(IsCorrect) **do**

**begin**

**for** i := 1 **to** Number **do**

**if** Res = 0 **then**

**begin**

**for** j := 1 **to** Number **do**

**if** i <> j **then**

**if** MySets[i] <= MySets[j] **then**

IsCorrect := false

**else**

IsCorrect := true;

**if not** IsCorrect **then**

Res := i;

**end**;

inc(Num);

**end**;

Str(Res, Answer);

Calculation := Answer + ArrayofStrings[Res];

**end**;

**function** ChoiceInput(): Char;

**var**

Input: Char;

IsCorrect: Boolean;

**begin**

**repeat**

Readln(Input);

Input := UpCase(Input);

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

IsCorrect := true

**else**

**begin**

IsCorrect := false;

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

**end**;

**until** IsCorrect;

ChoiceInput := Input;

**end**;

**procedure** OutputConsole(Answer: String);

**var**

i, j: Byte;

Code: Integer;

**begin**

val(Answer[1], j, Code);

**if** j <> 0 **then**

**begin**

Write('Sequence number is: ');

Writeln(Answer[1]);

Write('Set itself is: ');

**for** i := 2 **to** Length(Answer) **do**

Write(Answer[i]);

Writeln;

**end**

**else**

Writeln('There is no such set.');

**end**;

**procedure** ShowAnswerFile(**var** NewFile: TextFile; Answer: String);

**var**

i, j: Byte;

Code: Integer;

**begin**

val(Answer[1], j, Code);

**if** j <> 0 **then**

**begin**

Write(NewFile, 'Sequence number is: ');

Writeln(NewFile, Answer[1]);

Write(NewFile, 'Set itself is: ');

**for** i := 2 **to** Length(Answer) **do**

Write(NewFile, Answer[i]);

**end**

**else**

Writeln(NewFile, 'There is no such set.');

**end**;

**procedure** OutputFile(Answer: String);

**var**

IsCorrect: Boolean;

NewFile: TextFile;

NameOfFile: String;

**begin**

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

IsCorrect := false;

**repeat**

Readln(NameOfFile);

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

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

**else**

**begin**

IsCorrect := true;

Assign(NewFile, NameOfFile);

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

**if** ChoiceInput = 'Y' **then**

**try**

Rewrite(NewFile);

ShowAnswerFile(NewFile, Answer);

**except**

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

IsCorrect:= false;

**end**

**else**

**try**

Append(NewFile);

ShowAnswerFile(NewFile, Answer);

**except**

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

IsCorrect:= false;

**end**;

**end**;

**until** IsCorrect;

Close(NewFile);

**end**;

**function** CheckString(): String;

**var**

MyStr: String;

Len: Byte;

IsCorrect: Boolean;

**begin**

IsCorrect := false;

**repeat**

Readln(MyStr);

Len := Length(MyStr);

**if** (Len >= MinLength) **and** (Len <= MaxLength) **then**

IsCorrect := true

**else**

Writeln('Enter string with length from interval [', MinLength,

'..', MaxLength, ']: ');

**until** IsCorrect;

CheckString := MyStr;

**end**;

**function** CheckNumber(): Byte;

**var**

Num: Integer;

IsCorrect: Boolean;

**begin**

IsCorrect := false;

**repeat**

**try**

Readln(Num);

**if** (Num >= MinNumber) **and** (Num <= MaxNumber) **then**

IsCorrect := true

**else**

Writeln('Enter number from interval [', MinNumber, '..',

MaxNumber, ']: ');

**except**

Writeln('Check entered data. Enter number from interval [',

MinNumber, '..', MaxNumber, ']: ');

**end**;

**until** IsCorrect;

CheckNumber := Num;

**end**;

**procedure** GetDataConsole(**var** Number: Byte; **var** ArrayofStrings: TArr);

**var**

i: Byte;

**begin**

Write('Enter number of strings: ');

Number := CheckNumber();

**for** i := 1 **to** Number **do**

**begin**

Write('Enter string[', i,']: ');

ArrayofStrings[i] := CheckString();

**end**;

**end**;

**function** CheckStringFile(**var** MyFile: TextFile; **var** Num: Byte): String;

**var**

MyStr: String;

Len: Byte;

**begin**

Readln(MyFile, MyStr);

Len := Length(MyStr);

**if** (Len < MinLength) **or** (Len > MaxLength) **then**

inc(Num);

CheckStringFile := MyStr;

**end**;

**procedure** ReadFromFile(**var** MyFile: TextFile; **var** Number: Byte; **var**

ArrayofStrings: TArr);

**var**

i, Num: Byte;

**begin**

Reset(MyFile);

Number := 0;

Num := 0;

**while** (Number < 10)**and**(**not** Eof(MyFile)) **do**

**begin**

inc(Number);

ArrayofStrings[Number] := CheckStringFile(MyFile, Num);

Readln(MyFile);

**end**;

**if** Num <> 0 **then**

**begin**

Number := 0;

Writeln('Enter string with length from interval [', MinLength, '..',

MaxLength, '].');

**end**;

**end**;

**procedure** GetDataFile(**var** Number: Byte; **var** ArrayofStrings: TArr);

**var**

IsCorrect: Boolean;

NameOfFile: String;

MyFile: TextFile;

**begin**

Write('Enter file 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**

**begin**

IsCorrect := true;

ReadFromFile(MyFile, Number, ArrayofStrings);

**end**;

**end**;

**until** IsCorrect;

CloseFile(MyFile);

**end**;

**procedure** Main();

**var**

ArrayofStrings: TArr;

Number: Byte;

Answer: String[40];

**begin**

Writeln('This program finds a set that is a subset of all other sets.');

Write('Would you like to use File input instead of Console input? Enter

Y(Yes) or N(No): ');

**if** ChoiceInput = 'Y' **then**

GetDataFile(Number, ArrayofStrings)

**else**

GetDataConsole(Number, ArrayofStrings);

Answer := Calculation(Number, ArrayofStrings);

Write('Would you like to write down the answer to File instead of

Console? Enter Y(Yes) or N(No): ');

**if** ChoiceInput = 'Y' **then**

OutputFile(Answer)

**else**

OutputConsole(Answer);

Writeln('Press "Enter" to exit.');

Readln;

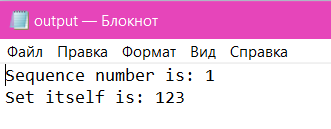
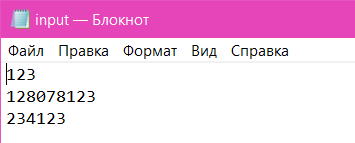
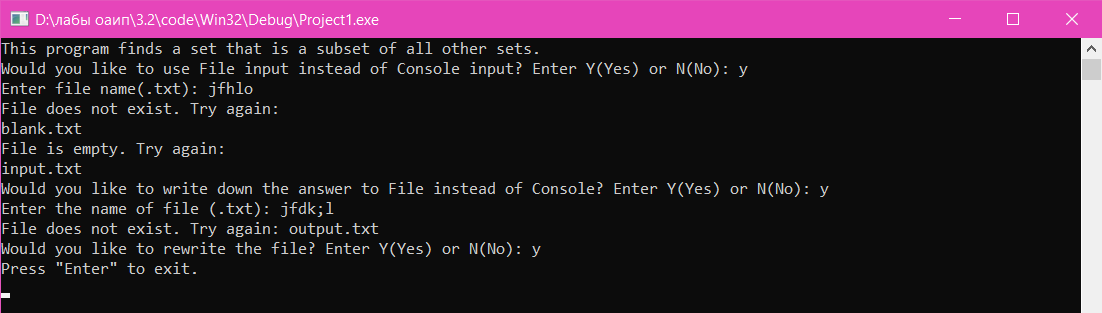
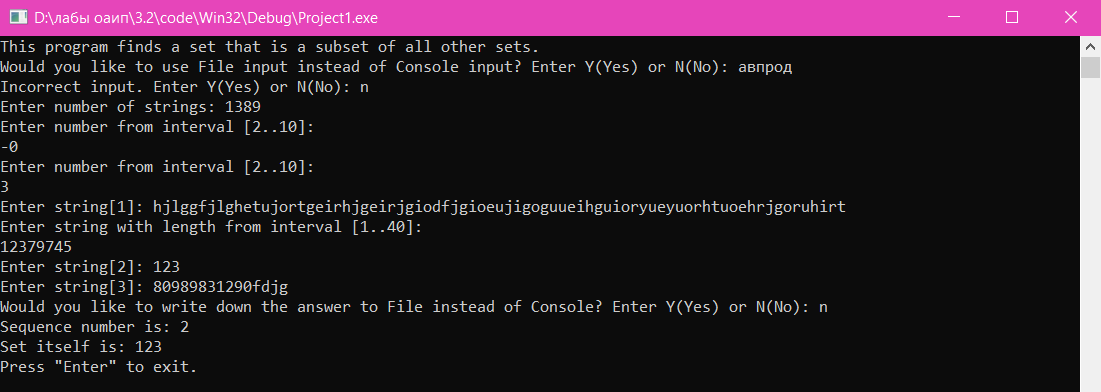
**end**;

**begin**

Main();

**end**.

**Скриншоты:**



**Блок-схема:**







