УО «Белорусский государственный университет информатики и радиоэлектроники»

Кафедра ПОИТ

Отчет по лабораторной работе №2.4

по предмету «Основы алгоритмизации и программирования»

Вариант 20

Выполнил:

Машевский Д.В

Гр. 351003

Проверил:

Данилова Г. В.

Минск 2023

Задание:

Дана последовательность чисел a1, a2,..., an. Каждый член этой последовательности кроме крайних заменить суммой соседних с ним членов.

**Код на Delphi**:

Program Laba24;

Uses

SysUtils;

Type

Mas = Array Of Integer;

Procedure DelProga();

Begin

Writeln('Данная программа заменяет суммой соседних с ним элементов все члены

последовательности чисел a1, a2,..., an,');

Writeln('кроме крайних');

End;

Function InputChoice(): Integer;

Var

Vib: Integer;

IsCorrect: Boolean;

Begin

Vib:= 0;

Write('Введите 0, если хотите сделать ввод с консоли; 1, если хотите ввод из

файла.');

Repeat

IsCorrect := True;

Write('Выбор: ');

Try

Readln(Vib);

Except

IsCorrect := False;

Write('Ошибка ввода. ');

End;

If IsCorrect and (Vib <> 1) and (Vib <> 0) then

Begin

IsCorrect := False;

Write('Число должно быть либо 0, либо 1. ');

End;

Until IsCorrect;

InputChoice := Vib;

End;

Function PTF(): String;

Var

Path: String;

IsCorrect: Boolean;

Begin

Repeat

IsCorrect := True;

Write('Укажите путь к файлу: ');

Readln(Path);

If Not FileExists(Path) Then

Begin

Write('Такой файл не найден. ');

IsCorrect := False;

End;

If IsCorrect And (ExtractFileExt(Path) <> '.txt') Then

Begin

IsCorrect := False;

Write('У файла должно быть расширение txt. ');

End;

Until IsCorrect;

PTF := Path;

End;

Procedure Na();

Begin

Writeln('Введите размер массива');

End;

Function EnterN(): Integer;

Const

MIN = 2;

Var

N: Integer;

IsCorrect: Boolean;

Begin

N:= 0;

Repeat

IsCorrect := True;

Try

Readln(N);

If N < MIN Then

Begin

Writeln('Введено недопустимое число элементов. Повторите ввод.');

IsCorrect := False;

End;

Except

Writeln('Введены некорректные данные. Повторите ввод.');

IsCorrect := False;

End;

Until IsCorrect;

EnterN := N;

End;

Function ReadSizeFile(Path: String): Integer;

Var

N: Integer;

IsCorrect: Boolean;

InputFile: TextFile;

Begin

N:= 0;

AssignFile(InputFile, Path);

Reset(InputFile);

IsCorrect := True;

Try

Readln(InputFile, N);

Except

IsCorrect := False;

Writeln('Ошибка при считывании порядка матрицы. ');

End;

If IsCorrect And (N < 2) Then

Begin

IsCorrect := False;

Writeln('Значение порядка матрицы в файле должно быть больше 1. ');

End;

Close(InputFile);

If IsCorrect Then

ReadSizeFile := N

Else

Write('Введите порядок матрицы с консоли. ');

ReadSizeFile := (EnterN);

End;

Function MasConsol(A: Mas; N: Integer): Mas;

Var

I: Integer;

IsCorrect: Boolean;

Begin

SetLength(A, N);

For I := 0 To N - 1 Do

Begin

Repeat

IsCorrect := True;

Write('A[', I + 1, '] = ');

Try

Readln(A[I]);

Except

Writeln('Введены некорректные данные. Повторите ввод.');

IsCorrect := False;

End;

Until IsCorrect;

End;

MasConsol := A;

End;

Function OutputChoice(): Integer;

Const

MIN = 0;

MAX = 1;

Var

Vib: Integer;

IsCorrect: Boolean;

Begin

Vib:= 0;

Write('Введите 0, если хотите сделать вывод в консоль; 1, если хотите вывод в

файл.');

Repeat

IsCorrect := True;

Write('Выбор: ');

Try

Readln(Vib);

Except

IsCorrect := False;

Write('Ошибка ввода. ');

End;

If IsCorrect And (Vib <> MAX) And (Vib <> MIN) Then

Begin

IsCorrect := False;

Write('Число должно быть либо 0, либо 1. ');

End;

Until IsCorrect;

OutputChoice :=Vib;

End;

Function ReadWrongElement(I: Integer):Integer;

Var

Elem: Integer;

IsCorrect: Boolean;

Begin

Elem:= 0;

Repeat

IsCorrect := True;

Write('Введите а', (I + 1),' с консоли: ');

Try

Readln(Elem);

Except

IsCorrect := False;

Write('Ошибка ввода. ');

End;

Until IsCorrect;

ReadWrongElement := Elem;

End;

Function MasFile(N: Integer; Path: String): Mas;

Var

A: Mas;

IsCorrect: Boolean;

InputFile: TextFile;

I: Integer;

Begin

N:= 0;

AssignFile(InputFile, Path);

Reset(InputFile);

Readln(InputFile);

SetLength(A, N);

For I := 0 To N - 1 Do

Begin

IsCorrect := True;

Try

Read(InputFile, A[I]);

Except

IsCorrect := False;

Write('Ошибка при считывании a', (I + 1), ' с файла. ');

End;

If Not IsCorrect Then

A[I] := ReadWrongElement(I);

End;

Close(InputFile);

MasFile := A;

End;

Function MasRead(A: Mas; Vib: Integer): Mas;

Var

N: Integer;

InputFile: TextFile;

Path: String;

Begin

Na();

If Vib = 0 Then

Begin

N := EnterN();

A := MasConsol(A, N);

MasRead := A;

End

Else

Begin

Path := PTF();

AssignFile(InputFile, Path);

Reset(InputFile);

N := ReadSizeFile(Path);

MasRead := MasFile(N, Path);

End;

End;

Function ZamSum(const A: Mas): Mas;

Var

I: Integer;

Begin

SetLength(Result, Length(A));

Result[0] := A[0]; // Для крайнего левого элемента

Result[Length(A) - 1] := A[Length(A) - 1]; // Для крайнего правого элемента

For I := 1 To Length(A) - 2 Do

Begin

Result[I] := A[I - 1] + A[I + 1];

End;

ZamSum := Result;

End;

Procedure PrintZamMasSumKrChlConsol(Result: Mas);

Var

N, I: Integer;

Begin

N := Length(Result);

For I := 0 To N - 1 Do

Begin

Write(' ', Result[I], ' ');

End;

End;

Procedure PrintZamMasSumKrChlFile(Result: Mas);

Var

N, I: Integer;

IsCorrect: Boolean;

Path: String;

OutputFile: TextFile;

Begin

N:= 0;

Path := PTF();

AssignFile(OutputFile, Path);

Reset(OutputFile);

Rewrite(OutputFile);

For I := 0 To N - 1 Do

Begin

Write(OutputFile, Result[I], ' ')

End;

Close(OutputFile);

End;

Procedure VivodSum(Vib: Integer; Result: Mas);

Begin

If Vib = 0 Then

PrintZamMasSumKrChlConsol(Result)

Else

Begin

PrintZamMasSumKrChlFile(Result);

End;

End;

Var

Vib1, Vib2, N: Integer;

Massiv1, Mass, MasGl,A, Result: Mas;

Begin

N:= 0;

DelProga();

Vib1 := InputChoice();

A := MasRead(A, Vib1);

Result := ZamSum(A);

Vib2 := OutputChoice();

VivodSum(Vib2, Result);

End.

**Код на С++:**

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

void delProg() {

cout << "Данная программа заменяет суммой соседних с ним элементов все члены

последовательности чисел a1, a2,..., an, кроме крайних";

}

int inputChoice() {

int vib;

bool isIncorrect;

cout << "Введите 0, если хотите сделать ввод с консоли; 1, если хотите ввод из

файла. ";

do

{

isIncorrect = false;

cout << "Выбор: ";

cin >> vib;

if (cin.fail() || cin.peek() != '\n')

{

isIncorrect = true;

cout << "Ошибка ввода. ";

cin.clear();

while (cin.get() != '\n');

}

if (!(isIncorrect) && (vib != 1) && (vib != 0))

{

isIncorrect = true;

cout << "Число должно быть либо 0, либо 1. ";

cin.clear();

while (cin.get() != '\n');

}

} while (isIncorrect);

return vib;

}

int enterN() {

int n;

bool isIncorrect;

do

{

isIncorrect = false;

cout << "Введите размер массива: ";

cin >> n;

if (cin.fail() || cin.peek() != '\n')

{

isIncorrect = true;

cout << "Ошибка ввода. ";

cin.clear();

while (cin.get() != '\n');

}

if (!(isIncorrect) && (n < 2))

{

isIncorrect = true;

cout << "Введено недопустимое число элементов. Повторите ввод. ";

cin.clear();

while (cin.get() != '\n');

}

} while (isIncorrect);

return n;

}

int\* masConsol(int n) {

bool isIncorrect;

int i;

int\* a = new int[n];

for (i = 0; i < n; i++)

{

do

{

isIncorrect = false;

cout << "Введите a" << (i + 1) << ": ";

cin >> a[i];

if (cin.fail() || cin.peek() != '\n')

{

isIncorrect = true;

cout << "Ошибка ввода. ";

cin.clear();

while (cin.get() != '\n');

}

} while (isIncorrect);

}

return a;

}

void outputPathInputCondition()

{

cout << "Введите путь к файлу, содержащему массив:\n";

}

bool isFileCorrect(string path)

{

bool isIncorrect = false;

string line;

fstream f(path);

if (!f.is\_open())

{

isIncorrect = true;

cout << "Нет файла по заданному пути, либо же он отсутствует или недоступен

для открытия. ";

}

else if ((path[path.length() - 1] != 't') &&

(path[path.length() - 2] != 'x') &&

(path[path.length() - 3] != 't') &&

(path[path.length() - 4] != '.'))

{

isIncorrect = true;

cout << "Недопустимое расширение файла. ";

}

f.close();

return isIncorrect;

}

string pTF()

{

bool isIncorrect;

string path;

do

{

cout << "Укажите путь к файлу:\n";

isIncorrect = false;

cin >> path;

isIncorrect = isFileCorrect(path);

if (isIncorrect)

{

cout << "Повторите ввод пути к файлу:\n";

}

} while (isIncorrect);

return path;

}

int readSizeFile(string path) {

int n, i;

i = 0;

bool isIncorrect;

ifstream fin(path);

isIncorrect = false;

if (!fin.eof())

{

fin >> n;

if (fin.fail() || fin.peek() != '\n')

{

cout << "Ошибка считывания размера с файла. \n";

isIncorrect = true;

}

if (!isIncorrect && (n < 2))

{

cout << "Значение размера массива должно быть больше 1. \n";

isIncorrect = true;

}

}

fin.close();

if (isIncorrect)

{

n = enterN();

}

return n;

}

int\* matrixFile(string path, int n) {

int\* a = new int[n];

ifstream fin(path);

fin.seekg(2, ios::beg);

int i;

int j;

for (i = 0; i < n; i++)

{

fin >> a[i];

}

fin.close();

return a;

}

int\* matrixRead(int ch, int& n) {

int\* a;

string path;

if (ch == 0)

{

n = enterN();

a = masConsol(n);

}

else

{

path = pTF();

n = readSizeFile(path);

outputPathInputCondition();

a = matrixFile(path, n);

}

return a;

}

int outputChoice() {

int vib;

bool isIncorrect;

cout << "Введите 0, если хотите сделать вывод в консоль; 1, если хотите вывод в

файл.";

do {

isIncorrect = false;

cout << "Выбор: ";

cin >> vib;

if (cin.fail() || cin.peek() != '\n')

{

isIncorrect = true;

cout << "\nОшибка ввода. ";

cin.clear();

while (cin.get() != '\n');

}

if (!(isIncorrect) && (vib != 1) && (vib != 0))

{

isIncorrect = true;

cout << "\nЧисло должно быть либо 0, либо 1. ";

cin.clear();

while (cin.get() != '\n');

}

} while (isIncorrect);

return vib;

}

void printZamMasSumKrChlFile(int\* result, int n) {

bool positiveInCol;

string path;

path = pTF();

int j;

int i;

i = 0;

ofstream fout(path);

for (i = 0; i < n; i++)

{

fout << result[i] << " ";

}

fout.close();

}

int\* zamSum(int\* a, int n) {

int\* result = new int [n];

int i;

result[0] = a[0];

result[n - 1] = a[n - 1];

for (i = 1; i < n - 1 ; i++)

{

result[i] = a[i - 1] + a[i + 1];

}

return result;

}

void printZamMasSumKrChlConsol(int\* result, int n) {

int i;

for (i = 0; i < n; i++)

{

cout << result[i] << " ";

}

cout << endl;

}

void vivodSum(int vib, int n, int\* result) {

if (vib == 0)

{

printZamMasSumKrChlConsol(result, n);

}

else

{

printZamMasSumKrChlFile(result, n);

}

}

int main() {

setlocale(LC\_ALL, "");

delProg();

int vib1;

int vib2;

int n;

int\* mas;

int\* a;

int\* result;

vib1 = inputChoice();

a = matrixRead(vib1, n);

result = zamSum(a, n);

vib2 = outputChoice();

vivodSum(vib2, n, result);

delete[] a;

delete[] result;

return 0;

}

**Код на Java:**

import java.util.Scanner;

import java.io.\*;

public class Main {

static Scanner scanner = new Scanner(System.in);

public static int inputChoice() {

int vib;

boolean isIncorrect;

vib = 0;

System.out.println("Введите 0, если хотите сделать ввод с консоли; 1, если

хотите ввод из файла. ");

do {

isIncorrect = false;

System.out.print("Выбор: ");

try {

vib = Integer.parseInt(scanner.nextLine());

} catch (Exception e) {

isIncorrect = true;

System.out.println("Ошибка ввода. ");

}

if (!isIncorrect && (vib != 1) && (vib != 0)) {

isIncorrect = true;

System.out.println("Число должно быть либо 0, либо 1. ");

}

} while (isIncorrect);

return vib;

}

public static int enterN() {

int n;

boolean isIncorrect;

n = 0;

do {

isIncorrect = false;

System.out.print("Введите размер массива: ");

try {

n = Integer.parseInt(scanner.nextLine());

} catch (Exception e) {

isIncorrect = true;

System.out.println("Ошибка ввода. ");

}

if (!isIncorrect && (n < 2)) {

isIncorrect = true;

System.out.println("Введено недопустимое число элементов. Повторите

ввод. ");

}

} while (isIncorrect);

return n;

}

public static int[] masConsol(int n) {

int[] a = new int[n];

boolean isIncorrect;

int i;

int j;

for (i = 0; i < n; i++) {

do {

isIncorrect = false;

System.out.println("Enter a[" + (i + 1) + "]: ");

try {

a[i] = Integer.parseInt(scanner.nextLine());

} catch (Exception e) {

isIncorrect = true;

System.out.println("Input error");

}

} while (isIncorrect);

}

return a;

}

public static String pTF() {

String path;

boolean isIncorrect;

do {

isIncorrect = false;

System.out.println("Укажите путь к файлу: ");

path = scanner.nextLine();

File file = new File(path);

if (!file.exists()) {

isIncorrect = true;

System.out.println("Такой файл не найден. ");

}

if (!path.endsWith(".txt")) {

isIncorrect = true;

System.out.println("У файла должно быть расширение txt. ");

}

} while (isIncorrect);

return path;

}

public static int readSizeFile(String path) {

int n;

boolean isIncorrect;

n = 0;

isIncorrect = false;

try {

Scanner scannerFile = new Scanner(new File(path));

n = scannerFile.nextInt();

scannerFile.close();

} catch (Exception e) {

isIncorrect = true;

System.out.println("Ошибка считывания размера с файла. ");

}

if (!isIncorrect && n < 2) {

isIncorrect = true;

System.out.println("Значение размера массива в файле должно быть больше

1. ");

}

if (isIncorrect) {

System.out.println("Введите размер массива с консоли. ");

n = enterN();

}

return n;

}

public static int[] matrixFile(String path, int n) {

int a[];

a = new int[n];

int i;

int j;

try {

Scanner scannerFile = new Scanner(new File(path));

for (i = 0; i < n; i++) {

a[i] = scannerFile.nextInt();

}

scannerFile.close();

} catch (Exception e) {

System.out.println("Ошибка при считывании файла. ");

}

return a;

}

public static int[] matrixRead(int vib) {

int[] a;

int n;

String path;

if (vib == 0) {

n = enterN();

a = masConsol(n);

} else {

path = pTF();

n = readSizeFile(path);

a = matrixFile(path, n);

}

return a;

}

public static int outputChoice() {

int vib;

boolean isIncorrect;

vib = 0;

System.out.println("Введите 0, если хотите сделать вывод в консоль; 1, если

хотите вывод в файл. ");

do {

isIncorrect = false;

System.out.print("Выбор: ");

try {

vib = Integer.parseInt(scanner.nextLine());

} catch (Exception e) {

isIncorrect = true;

System.out.println("\nОшибка ввода. ");

}

if (!isIncorrect && (vib != 1) && (vib != 0)) {

isIncorrect = true;

System.out.println("Число должно быть либо 0, либо 1. ");

}

} while (isIncorrect);

return vib;

}

public static int[] zamSum(int[] a) {

int n = a.length;

int[] result = new int[n];

int i;

result[0] = a[0];

result[n - 1] = a[n - 1];

for (i = 1; i < n - 1; i++) {

result[i] = a[i - 1] + a[i + 1];

}

return result;

}

public static void printResultMatrix(int[] result) {

int n = result.length;

int i;

for (i = 0; i < n; i++) {

System.out.print(result[i] + " ");

System.out.println();

}

}

public static void printResultMatrixFile(int[] result) {

int i;

int n = result.length;

String path;

path = pTF();

try (PrintWriter zapis = new PrintWriter(new FileWriter(path))) {

for (i = 0; i < n; i++) {

zapis.write(result[i] + " ");

}

} catch (Exception e) {

System.out.println("Не удалось записать данные в файл. ");

}

}

public static void vivodSum(int vib, int[] result) {

if (vib == 0) {

printResultMatrix(result);

}

else {

printResultMatrixFile(result);

}

}

public static void main(String[] args) {

int vib1;

int vib2;

int[] a;

int[] result;

vib1 = inputChoice();

a = matrixRead(vib1);

result = zamSum(a);

vib2 = outputChoice();

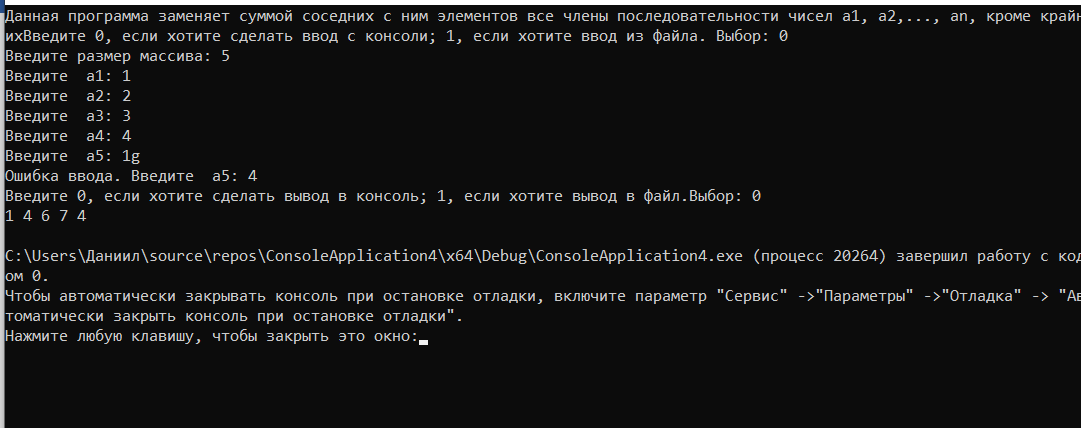
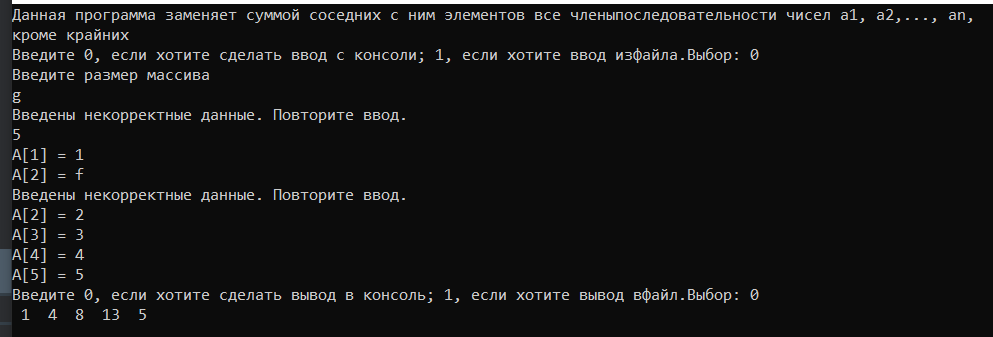
scanner.close();

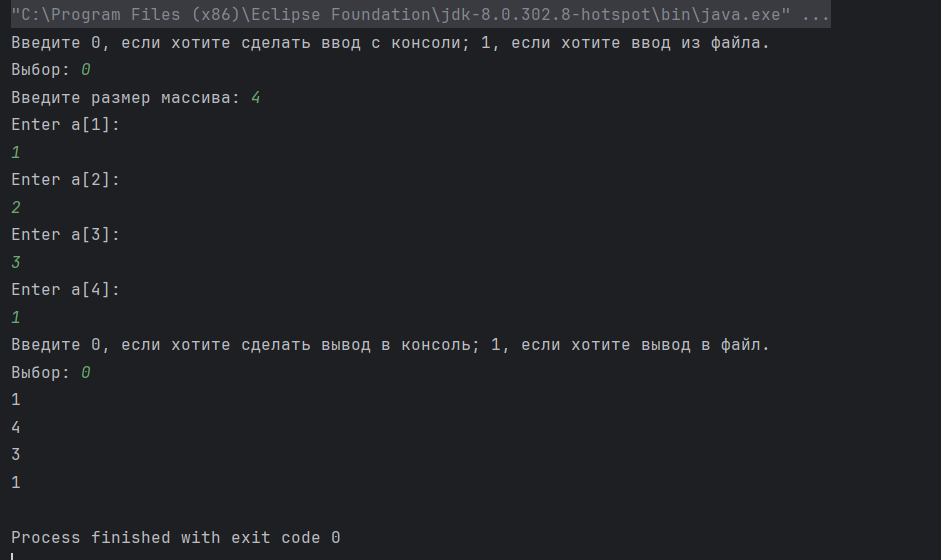
vivodSum(vib2, result);

}

}

**Скриншоты:**



****

**Блок-схема:**

