УО «Белорусский государственный университет информатики и радиоэлектроники»

Кафедра ПОИТ

Отчет по лабораторной работе №2.4

по предмету «Основы алгоритмизации и программирования»

Вариант 20

Выполнил:

Захвей И.В.

Гр. 351005

Проверил:

Данилова Г. В.

Минск 2023

**Задание:**

Перевести число в 16-ю систему счисления.

**Код программы Delphi:**

Program Project2;

Uses

System.SysUtils;

Type

TArray = Array Of Char;

Const

MIN\_ELEM = -MaxInt - 1;

MAX\_ELEM = MaxInt;

ERRORS: Array [0 .. 9] Of String = ('Successfull',

'Data is not correct, or number is too large',

'Enter the number within the borders',

'This is not a .txt file',

'This file is not exist',

'Data in file is not correct, or number is too

large',

'There is only one number in file should be

(without whitespace)',

'File is can not be opened', 'File is can not

be opened',

'File is not exist');

Procedure PrintInf();

Begin

Writeln('Program converts decimal to hexadecimal');

End;

Function GetLenOfNum(Num: Integer): Integer;

Var

Len: Integer;

Begin

Len := 1;

While Num > 9 Do

Begin

Inc(Len);

Num := Num Div 10;

End;

GetLenOfNum := Len;

End;

Procedure FillWithZero(Var Arr: Array Of Char);

Var

I: Integer;

Begin

For I := 0 To High(Arr) Do

Arr[I] := '0';

End;

Function IntToHexArr(Num: Integer; IsNumNegative: Boolean): TArray;

Const

HEX\_ELEM: Array [0 .. 16] Of Char = ('0', '1', '2', '3', '4', '5', '6', '7',

'8', '9', 'A', 'B', 'C', 'D', 'E', 'F', '-');

Var

Index, Size: Integer;

Arr: TArray;

Begin

If IsNumNegative Then

Num := -Num;

Index := 0;

Size := GetLenOfNum(Num);

SetLength(Arr, Size);

FillWithZero(Arr);

If Num > 15 Then

While Num > 1 Do

Begin

Arr[Index] := HEX\_ELEM[Num Mod 16];

Num := Num Div 16;

Inc(Index);

End

Else

Begin

Arr[Index] := HEX\_ELEM[Num];

Inc(Index);

End;

If IsNumNegative Then

Arr[Index] := HEX\_ELEM[16];

IntToHexArr := Arr;

End;

Function ReversArr(Arr: TArray): TArray;

Var

ReversedArr: TArray;

Index, I: Integer;

Begin

Index := 0;

I := High(Arr);

SetLength(ReversedArr, Length(Arr));

While Index < Length(Arr) Do

Begin

ReversedArr[Index] := Arr[I];

Dec(I);

Inc(Index);

End;

ReversArr := ReversedArr;

End;

Function InpValidNum(Var Num: Integer; Const MIN, MAX: Integer): Integer;

Var

IsCorrect: Boolean;

Err: Integer;

Begin

Err := 0;

Try

Readln(Num);

IsCorrect := True;

Except

Err := 1;

IsCorrect := False;

End;

If IsCorrect And ((Num < MIN) Or (Num > MAX)) Then

Begin

Err := 2;

End;

InpValidNum := Err;

End;

Function UserChoice(): Integer;

Var

Choice, Err: Integer;

Begin

Writeln('Choose a way of input/output of data', #13#10, '1 -- Console',

#13#10, '2 -- File');

Err := InpValidNum(Choice, 1, 2);

While (Err > 0) Do

Begin

Writeln(ERRORS[Err]);

Writeln('Please, enter again');

Err := InpValidNum(Choice, 1, 2);

End;

UserChoice := Choice;

End;

Procedure InputFromConsole(Var Num: Integer);

Var

Err: Integer;

Begin

Writeln('Enter the number from ', MIN\_ELEM, ' to ', MAX\_ELEM);

Err := InpValidNum(Num, MIN\_ELEM, MAX\_ELEM);

While Err > 0 Do

Begin

Writeln(ERRORS[Err]);

Writeln('Please, enter again');

Err := InpValidNum(Num, MIN\_ELEM, MAX\_ELEM);

End;

End;

Function FileAvailable(Name: String; ForReset: Boolean): Integer;

Var

Err: Integer;

MyFile: TextFile;

Begin

Err := 0;

AssignFile(MyFile, Name);

If ForReset Then

Try

Try

Reset(MyFile);

Finally

Close(MyFile);

End;

Except

Err := 7;

End

Else

Try

Try

Rewrite(MyFile);

Finally

Close(MyFile);

End;

Except

Err := 8;

End;

FileAvailable := Err;

End;

Function FileTxt(Name: String): Integer;

Var

Err: Integer;

Begin

Err := 0;

If ExtractFileExt(Name) <> '.txt' Then

Err := 3;

FileTxt := Err;

End;

Function FileExist(Name: String): Integer;

Var

Err: Integer;

Begin

Err := 0;

If Not FileExists(Name) Then

Err := 9;

FileExist := Err;

End;

Function GetFileName(ForReset: Boolean): String;

Var

IsCorrect: Boolean;

ErrExist, ErrTxt, ErrAvailable: Integer;

FileName: String;

Begin

Repeat

IsCorrect := True;

Readln(FileName);

ErrExist := FileExist(FileName);

ErrTxt := FileTxt(FileName);

If (ErrExist > 0) Then

Begin

Writeln(ERRORS[9]);

IsCorrect := False;

End

Else If (ErrTxt > 0) Then

Begin

Writeln(ERRORS[ErrTxt]);

IsCorrect := False;

End;

if ((ErrExist = 0) And (ErrTxt = 0)) then

Begin

ErrAvailable := FileAvailable(FileName, ForReset);

If (ErrAvailable > 0) Then

Begin

Writeln(ERRORS[ErrAvailable]);

IsCorrect := False;

End;

End;

Until IsCorrect;

GetFileName := FileName;

End;

Function ReadFile(Var Num: Integer; Name: String): Integer;

Var

Err: Integer;

IsCorrect: Boolean;

InfFile: TextFile;

Begin

AssignFile(InfFile, Name);

Reset(InfFile);

IsCorrect := True;

Err := 0;

Try

Read(InfFile, Num);

Except

Err := 5;

IsCorrect := False;

End;

If IsCorrect And ((Num < MIN\_ELEM) Or (Num > MAX\_ELEM)) Then

Begin

IsCorrect := False;

Err := 2;

End;

If IsCorrect And Not EoF(InfFile) Then

Begin

IsCorrect := False;

Err := 6;

End;

CloseFile(InfFile);

ReadFile := Err;

End;

Procedure InputFromFile(Var Num: Integer);

Var

Err: Integer;

FileName: String;

Begin

Writeln('Enter full path to file');

FileName := GetFileName(True);

Err := ReadFile(Num, FileName);

While (Err > 0) Do

Begin

Writeln(ERRORS[Err]);

Writeln('Please, enter full path again');

FileName := GetFileName(True);

Err := ReadFile(Num, FileName);

End;

Writeln('Reading is successfull');

End;

Function InputInf(): Integer;

Var

Num, Choice: Integer;

Begin

Choice := UserChoice();

If (Choice = 1) Then

InputFromConsole(Num)

Else

InputFromFile(Num);

InputInf := Num;

End;

Function GetArrOfHexDigit(Num: Integer): TArray;

Var

Size: Integer;

IsNumNegative: Boolean;

Arr: TArray;

Begin

IsNumNegative := Num < 0;

Arr := IntToHexArr(Num, IsNumNegative);

GetArrOfHexDigit := ReversArr(Arr);

End;

Procedure OutputInConsole(Num: Integer; Arr: TArray);

Var

Index, I: Integer;

Begin

Index := 0;

Writeln('Decimal number:');

Writeln(Num);

Writeln('Hexadecimal number:');

If (High(Arr) > 0) Then

Begin

While (Arr[Index] = '0') Do

Inc(Index);

For I := Index To High(Arr) Do

Write(Arr[I]);

End

Else

Writeln(Arr[Index]);

End;

Procedure OutputInFile(Num: Integer; Arr: TArray);

Var

Index, I: Integer;

FileName: String;

MyFile: TextFile;

Begin

Writeln('Enter full path to file');

FileName := GetFileName(False);

AssignFile(MyFile, FileName);

Rewrite(MyFile);

Index := 0;

Writeln(MyFile, 'Decimal number:');

Writeln(MyFile, Num);

Writeln(MyFile, 'Hexadecimal number:');

If (High(Arr) > 0) Then

Begin

While (Arr[Index] = '0') Do

Inc(Index);

For I := Index To High(Arr) Do

Write(MyFile, Arr[I]);

End

Else

Writeln(MyFile, Arr[Index]);

CloseFile(MyFile);

Writeln('Writing is successfull');

End;

Procedure OutputInf(Num: Integer; ArrOfDigit: TArray);

Var

Choice: Integer;

Begin

Choice := UserChoice();

If (Choice = 1) Then

OutputInConsole(Num, ArrOfDigit)

Else

OutputInFile(Num, ArrOfDigit);

End;

Var

Num: Integer;

ArrOfDigit: TArray;

Begin

PrintInf();

Num := InputInf();

ArrOfDigit := GetArrOfHexDigit(Num);

OutputInf(Num, ArrOfDigit);

Readln;

End.

**Код программы С++:**

#include <iostream>

#include <fstream>

#include <string>

using std::string;

using std::cin;

using std::cout;

using std::endl;

using std::ifstream;

using std::ofstream;

const int MIN\_ELEM = -2147483647;

const int MAX\_ELEM = 2147483647;

const string ERRORS[] = {"Successfull",

"Data is not correct, or number is too large\n",

"Enter the number within the borders\n",

"This is not a .txt file\n",

"This file is not exist\n",

"Data in file is not correct, or number is too large\n",

"There is only one number in file should be (without

whitespace)\n"};

void printInf()

{

cout << "Program converts decimal to hexadecimal\n";

}

int getLenOfNum(int num)

{

int len = 1;

while (num > 9) {

len++;

num /= 10;

}

return len;

}

void fillWithZero(char \*arr, int size) {

int i;

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

arr[i] = '0';

}

}

char \*intToHexArr(int num, int size, bool isNumNegative)

{

const char HEX\_ELEM[] = { '0', '1', '2', '3', '4', '5', '6', '7', '8',

'9', 'A', 'B', 'C', 'D', 'E', 'F', '-' };

if (isNumNegative)

num = -num;

int index = 0;

char \*hexNumArr = new char[size];

fillWithZero(hexNumArr, size);

if (num > 15)

{

while (num > 1)

{

hexNumArr[index++] = HEX\_ELEM[num % 16];

num /= 16;

}

}

else

hexNumArr[index++] = HEX\_ELEM[num];

if (isNumNegative)

hexNumArr[index] = HEX\_ELEM[16];

return hexNumArr;

}

void freeArr(char \*arr)

{

delete[] arr;

}

char \*reversArr(char \*arr, int size) {

char \*reversedArr = new char[size];

int index = 0;

int i = size - 1;

while (index < size) {

reversedArr[index++] = arr[i--];

}

freeArr(arr);

return reversedArr;

}

int inputNum(int &number, const int MIN, const int MAX)

{

int err = 0;

bool isIncorrect = false;

cin >> number;

if (cin.fail())

{

err = 1;

cin.clear();

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

isIncorrect = true;

}

if (!isIncorrect && cin.get() != '\n')

{

err = 1;

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

isIncorrect = true;

}

if (!isIncorrect && (number > MAX || number < MIN))

{

err = 2;

isIncorrect = true;

}

return err;

}

int userChoice()

{

int choice;

cout << "Choose a way of input/output of data\n"

<< "1 -- Console\n"

<< "2 -- File\n";

int err = inputNum(choice, 1, 2);

while (err != 0)

{

cout << ERRORS[err];

cout << "Please, enter again\n";

err = inputNum(choice, 1, 2);

}

return choice;

}

void inputFromConsole(int& num)

{

cout << "Enter the number from " << MIN\_ELEM << " to " << MAX\_ELEM << "\n";

int err = inputNum(num, MIN\_ELEM, MAX\_ELEM);

while (err != 0)

{

cout << ERRORS[err];

cout << "Please, enter again\n";

err = inputNum(num, MIN\_ELEM, MAX\_ELEM);

}

}

int readFile(int& num, string fileName)

{

int err = 0;

bool isCorrect = true;

ifstream file(fileName);

file >> num;

if (file.fail())

{

err = 5;

file.clear();

while (!file.eof())

file.get();

isCorrect = false;

}

if (file.eof()) {

err = 0;

isCorrect = false;

}

if (isCorrect && (file.get() != '\n'))

{

err = 6;

while (!file.eof())

file.get();

isCorrect = false;

}

if (isCorrect && (num < MIN\_ELEM || num > MAX\_ELEM)){

err = 2;

}

file.close();

return err;

}

int isFileExist(string nameOfFile)

{

int err = 0;

ifstream file(nameOfFile);

if (!file.is\_open())

err = 4;

file.close();

return err;

}

int thisIsTxtFile(string &fileName)

{

int err = 0;

if (fileName.length() > 4)

{

string lastFourChar = fileName.substr(fileName.length() - 4);

if (lastFourChar != ".txt")

err = 3;

}

else

err = 3;

return err;

}

string getFileName()

{

bool isIncorrect;

string name;

int errExist = 0;

int errTxt = 0;

do

{

cin >> name;

errExist = isFileExist(name);

errTxt = thisIsTxtFile(name);

isIncorrect = false;

if (errTxt != 0)

{

cout << ERRORS[errTxt];

isIncorrect = true;

}

else if (errExist != 0)

{

cout << ERRORS[errExist];

isIncorrect = true;

}

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

} while (isIncorrect);

return name;

}

void inputFromFile(int& num)

{

cout << "Enter full path to file\n";

string fileName = getFileName();

int err = readFile(num, fileName);

while (err != 0)

{

cout << ERRORS[err];

cout << "Please, enter full path again\n";

fileName = getFileName();

err = readFile(num, fileName);

}

cout << "Reading is successfull\n";

}

int inputInf()

{

int num;

int choice = userChoice();

if (choice == 1)

inputFromConsole(num);

else

inputFromFile(num);

return num;

}

char \*getArrOfHexDigit(int num, int &size)

{

bool isNumNegative = num < 0;

size = getLenOfNum(num);

char \*arr = intToHexArr(num, size, isNumNegative);

return reversArr(arr, size);

}

void outputInConsole(int num, char\* arr, int size)

{

int index = 0;

int i;

cout << "Decimal number:\n";

cout << num << endl;

cout << "Hexadecimal number:\n";

if (size > 1)

{

while (arr[index] == '0')

index++;

for (i = index; i < size; i++)

cout << arr[i];

}

else

cout << arr[index];

freeArr(arr);

}

void outputInFile(int num, char\* arr, int size)

{

int index = 0;

int i;

cout << "Enter full path to file\n";

string fileName = getFileName();

ofstream file(fileName);

file << "Decimal number:" << endl;

file << num << endl;

file << "Hexadecimal number:" << endl;

if (size > 1)

{

while (arr[index] == '0')

index++;

for (i = index; i < size; i++)

file<< arr[i];

}

else

file << arr[index];

cout << "Writing is successfull\n";

freeArr(arr);

file.close();

}

void outputInf(int num, char\* arrOfDigit, int size)

{

int choice = userChoice();

if (choice == 1)

outputInConsole(num, arrOfDigit, size);

else

outputInFile(num, arrOfDigit, size);

}

int main()

{

int num, size;

printInf();

num = inputInf();

char \*arrOfDigit = getArrOfHexDigit(num, size);

outputInf(num, arrOfDigit, size);

return 0;

}

**Код программы Java:**

import java.io.File;

import java.io.IOException;

import java.io.PrintWriter;

import java.nio.file.Path;

import java.nio.file.Paths;

import java.util.Scanner;

public class Main {

static final String[] ERRORS ={"Successfull",

"Data is not correct, or number is too large",

"Enter the number within the borders",

"This is not a .txt file",

"This file is not exist"};

static final int MIN\_ELEM = -2147483647;

static final int MAX\_ELEM = 2147483647;

static void printInf() {

System.out.println("Program converts decimal to hexadecimal");

}

static int getLenOfNum(int num) {

int len = 1;

while (num > 9) {

len++;

num /= 10;

}

return len;

}

static void fillWithZero(char[] arr) {

int i;

for (i = 0; i < arr.length; i++) {

arr[i] = '0';

}

}

static char[] intToHexArr(int num, boolean isNumNegative) {

final char[] HEX\_ELEM = {'0', '1', '2', '3', '4', '5', '6', '7', '8',

'9', 'A', 'B', 'C', 'D', 'E', 'F', '-'};

if (isNumNegative)

num = -num;

int lenNum = getLenOfNum(num);

int index = 0;

char[] hexNumArr = new char[lenNum];

fillWithZero(hexNumArr);

if (num > 15) {

while (num > 1) {

hexNumArr[index++] = HEX\_ELEM[num % 16];

num /= 16;

}

} else {

hexNumArr[index] = HEX\_ELEM[num];

index++;

}

if (isNumNegative)

hexNumArr[index] = HEX\_ELEM[16];

return hexNumArr;

}

static char[] reversArr(char[] arr) {

char[] reversedArr = new char[arr.length];

int index = 0;

int i = arr.length - 1;

while (index < reversedArr.length) {

reversedArr[index++] = arr[i--];

}

return reversedArr;

}

static int inputNum(Scanner input,int[] number, final int MIN, final int MAX){

int errorCode = 0;

boolean isIncorrect = false;

try {

number[0] = Integer.parseInt(input.next());

} catch (NumberFormatException e) {

isIncorrect = true;

errorCode = 1;

}

if (!isIncorrect && (number[0] < MIN || number[0] > MAX)) {

errorCode = 2;

}

return errorCode;

}

static int userChoice(Scanner input) {

int[] choice = {0};

System.out.println("Choose a way of input/output of data\n"

+ "1 -- Console\n"

+ "2 -- File");

int err = inputNum(input, choice, 1, 2);

while (err != 0) {

System.err.println(ERRORS[err]);

System.out.println("Please, enter again");

err = inputNum(input, choice, 1, 2);

}

return choice[0];

}

static int readFile(String fileName) throws IOException, NumberFormatException{

Path path = Paths.get(fileName);

Scanner file = new Scanner(path);

int num = 0;

try {

num = Integer.parseInt(file.nextLine());

} catch (NumberFormatException e) {

throw new NumberFormatException("Data in file is not correct");

}

System.out.println("Reading is successfull");

return num;

}

static int isFileExist(String fileName) {

File file = new File(fileName);

int err = 0;

if (!file.exists())

err = 4;

return err;

}

static int thisIsTxtFile(String fileName) {

int err = 0;

if (!fileName.endsWith(".txt"))

err = 3;

return err;

}

static String getFileName(Scanner input) {

boolean isIncorrect;

String fileName;

fileName = input.nextLine();

System.out.println("Enter full path to file");

do {

isIncorrect = false;

fileName = input.nextLine();

int errTxt = thisIsTxtFile(fileName);

int errExist = isFileExist(fileName);

if (errTxt != 0) {

isIncorrect = true;

System.err.println(ERRORS[errTxt]);

}

else if (errExist != 0) {

isIncorrect = true;

System.err.println(ERRORS[errExist]);

}

} while (isIncorrect);

return fileName;

}

static void inputFromFile(Scanner input, int[] num) throws NumberFormatException,

IOException {

String fileName = getFileName(input);

try {

num[0] = readFile(fileName);

} catch (IOException e) {

throw new IOException("Exception with reading from the file");

}

}

static void inputFromConsole(Scanner input, int[] num) {

System.out.println("Enter the number from " + MIN\_ELEM + " to " + MAX\_ELEM);

int err = inputNum(input, num, MIN\_ELEM, MAX\_ELEM);

while(err != 0) {

System.err.println(ERRORS[err]);

System.out.println("Please, enter again");

err = inputNum(input, num, MIN\_ELEM, MAX\_ELEM);

}

}

static int inputInf(Scanner input) throws NumberFormatException, IOException{

int choice = userChoice(input);

int[] num = {0};

if (choice == 1) {

inputFromConsole(input, num);

} else {

inputFromFile(input, num);

}

return num[0];

}

static char[] getArrOfHexDigit(int num) {

boolean isNumNegative = num < 0;

char[] arr = intToHexArr(num, isNumNegative);

return arr = reversArr(arr);

}

static void writeInConsole(int num, char[] arr) {

int i;

int index = 0;

System.out.println("Decimal number:");

System.out.println(num);

System.out.println("hexadecimal number");

if (arr.length > 1) {

while (arr[index] == '0') {

index++;

}

for (i = index; i < arr.length; i++) {

System.out.print(arr[i]);

}

} else {

System.out.println(arr[index]);

}

}

static void writeInFile(Scanner input,int num, char[] arr) throws IOException {

int i;

int index = 0;

String fileName = getFileName(input);

PrintWriter file = new PrintWriter(fileName);

file.println("Decimal number:");

file.println(num);

file.println("Hexadecimal number:");

while (arr[index] == '0') {

index++;

}

for (i = index; i < arr.length; i++)

file.print(arr[i]);

file.close();

System.out.println("Writing is successfull");

}

static void outputInf(char[] arr, int num, Scanner input) throws IOException{

int choice = userChoice(input);

if (choice == 1) {

writeInConsole(num, arr);

} else {

try {

writeInFile(input, num, arr);

} catch (IOException e) {

throw new IOException("Exception with writing in the file");

}

}

}

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

printInf();

try {

int num = inputInf(input);

char[] arrOfDigit = getArrOfHexDigit(num);

outputInf(arrOfDigit, num, input);

} catch (Exception e) {

System.err.println(e.getMessage());

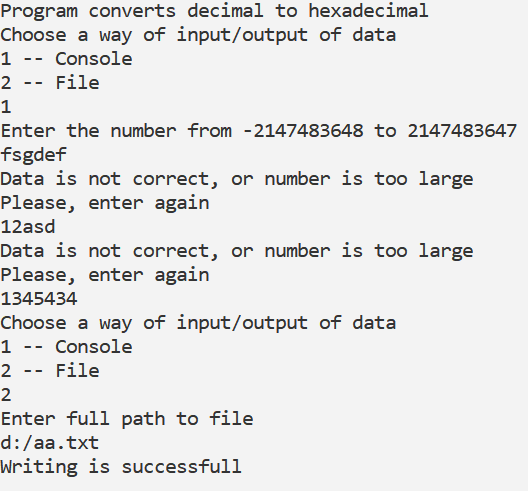
}

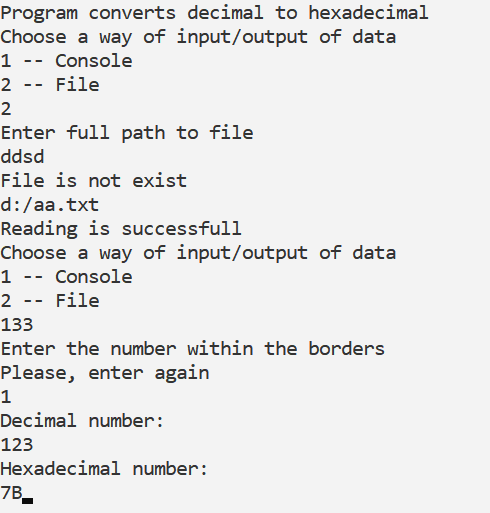
input.close();

}

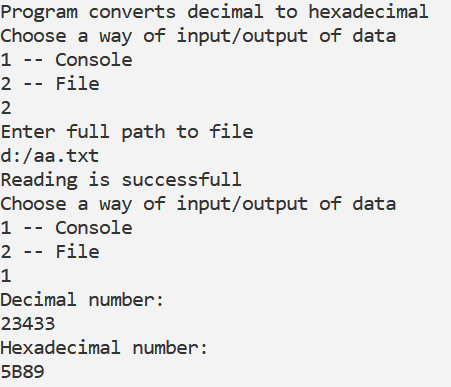
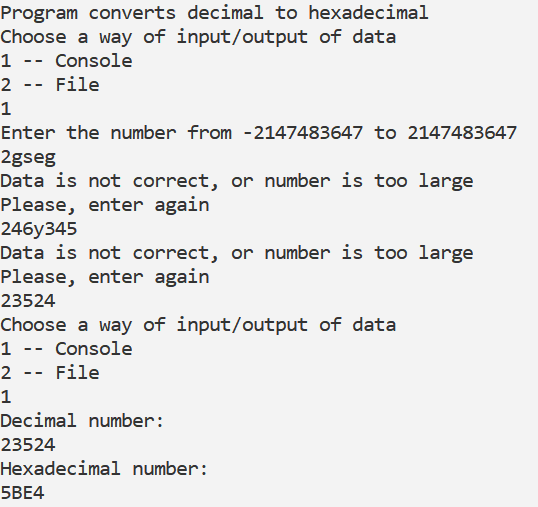
}

**Скриншоты:**

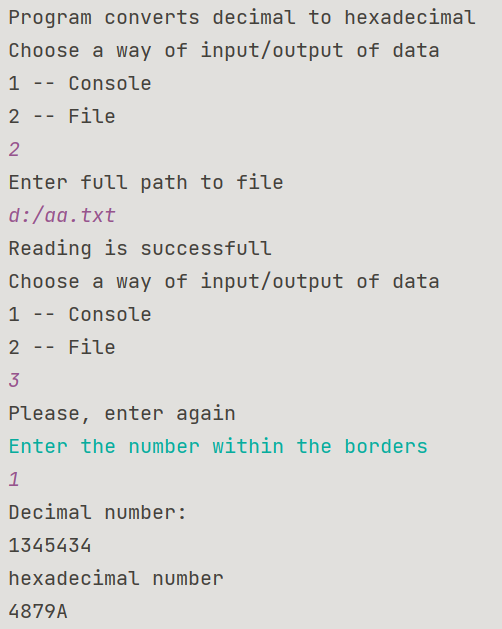
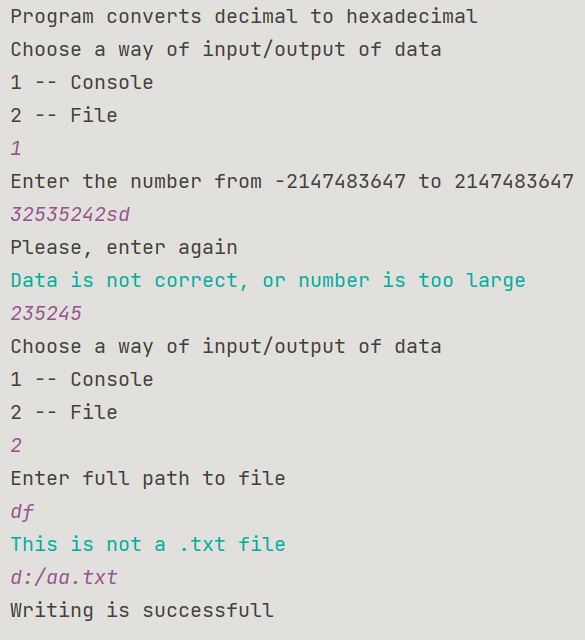
**Delphi:**

****

**C++:**

****

**Java:**

****

**Блок-схема:**



