УО «Белорусский государственный университет информатики и радиоэлектроники»

Кафедра ПОИТ

Отчет по лабораторной работе №2.2

по предмету «Основы алгоритмизации и программирования»

Вариант 18

Выполнил:

Егоров А.С.

Гр. 351005

Проверил:

Данилова Г. В.

Минск 2023

**Задание:**

Найти все натуральные числа, которые в k раз больше суммы своих цифр.

**Код программы Delphi:**

Program Exercise2;

Uses

System.SysUtils;

Const

MaxK = 10000000;

MinK = 1;

MaxDigit = 9;

MaxDigits = 10;

Var

K: Integer;

Procedure Input(Var K: Integer);

Var

IsCorrect: Boolean;

Begin

IsCorrect := false;

Writeln('The program finds all natural numbers that are k times greater than the sum of their digits.');

Repeat

Try

Writeln('Enter the number from ', MinK, ' to ', MaxK, '.');

Readln(K);

Except

Begin

Writeln('Invalid type! Try again.');

End;

End;

If K < MinK Then

Writeln('The entered number cannot be less than ', MinK,

'! Try again.')

Else If MaxK < K Then

Writeln('The entered number cannot be more than ', MaxK,

'! Try again.')

Else

IsCorrect := True;

Until IsCorrect;

End;

Procedure Output(Var K: Integer);

Begin

Write(K, ' ');

End;

Function DigitsSum(NaturalNumber: Integer): Integer;

Var

Sum: Integer;

Begin

Sum := 0;

While NaturalNumber > 0 Do

Begin

Sum := Sum + NaturalNumber Mod 10;

NaturalNumber := NaturalNumber Div 10;

End;

Result := Sum;

End;

Function FindMaxNumber(Const K: Integer): Integer;

Var

MaxNumber: Integer;

CountOfDigits: Integer;

Begin

MaxNumber := 1;

CountOfDigits := 1;

While (CountOfDigits < MaxDigits) And

((K \* MaxDigit \* CountOfDigits) > MaxNumber) Do

Begin

MaxNumber := MaxNumber \* 10;

CountOfDigits := CountOFDigits + 1;

End;

MaxNumber := MaxNumber Div 10;

Result := MaxNumber;

End;

Procedure FindNumber(Var K: Integer);

Var

NaturalNumber: Integer;

MaxNumber: Integer;

Begin

NaturalNumber := K;

MaxNumber := FindMaxNumber(K);

While NaturalNumber < MaxNumber Do

Begin

If NaturalNumber = DigitsSum(NaturalNumber) \* K Then

Output(NaturalNumber);

NaturalNumber := NaturalNumber + K;

End;

End;

Begin

K := 0;

Input(K);

FindNumber(K);

//freeze console

Writeln(#13#10,'Press enter to exit...');

Readln;

End.

**Код программы С++:**

#include<iostream>

const int minRangeOfInteger32 = -2147483648;

const int maxRangeOfInteger32 = 2147483647;

const int maxK = 10000000;

const int minK = 1;

const int maxDigit = 9;

const int maxDigits = 10;

void input(int& k)

{

bool isIncorrect = true;

std::cout << "The program finds all natural numbers that are k times greater than

the sum of their digits.\n";

do

{

std::cout << "Enter the number from "<< minK << " to " << maxK << ".\n";

std::cin >> k;

if (std::cin.get() != '\n')

{

std::cin.clear();

std::cin.ignore(5323, '\n');

std::cout << ((k == maxRangeOfInteger32 || k == minRangeOfInteger32)

? "The entering number not in range of Integer32!" : "Invalid type!");

std::cout << " Try again.\n";

}

else if (k < minK)

{

std::cout << "The entered number cannot be less than " << minK

<< "! Try again.\n";

}

else if (maxK < k)

std::cout << "The entered number cannot be more than " << maxK

<< "! Try again.\n";

else

isIncorrect = false;

} while (isIncorrect);

}

inline void output(int& naturalNumber)

{

std::cout << naturalNumber << " ";

}

int digitsSum(int naturalNumber)

{

int sum = 0;

while (naturalNumber)

{

sum += naturalNumber % 10;

naturalNumber /= 10;

}

return sum;

}

int findMaxNumber(const int& k)

{

int maxNumber = 1;

int countOfDigits = 1;

while (countOfDigits < maxDigits && k \* maxDigit \* countOfDigits++ > maxNumber)

{

maxNumber \*= 10;

};

return maxNumber /= 10;

}

void findNumbers(int& k)

{

int naturalNumber = k;

int maxNumber = findMaxNumber(k);

while (naturalNumber < maxNumber)

{

if (naturalNumber == digitsSum(naturalNumber) \* k)

output(naturalNumber);

naturalNumber += k;

}

}

int main()

{

int k = 0;

input(k);

findNumbers(k);

return 0;

}

**Код программы Java:**

import java.util.Scanner;  
  
  
public class Main {  
 static final int maxK = 10000000;  
 static final int minK = 1;  
 static final int maxDigit = 9;  
 static final int maxDigits = 10;  
 public static int input(int k) {  
 boolean isIncorrect = true;  
 boolean isNumber = false;  
 Scanner in = new Scanner(System.in);  
 System.out.println("The program finds all natural numbers that are k times

greater than the sum of their digits.");  
 do {  
 isNumber = false;  
 try {  
 k = Integer.parseInt(in.nextLine());  
 isNumber = true;  
 }  
 catch (NumberFormatException exception)  
 {  
 System.err.println("Invalid type! Try again.");  
 }  
 if (isNumber && k < minK)  
 System.err.printf("The entered number cannot be less than %d! Try

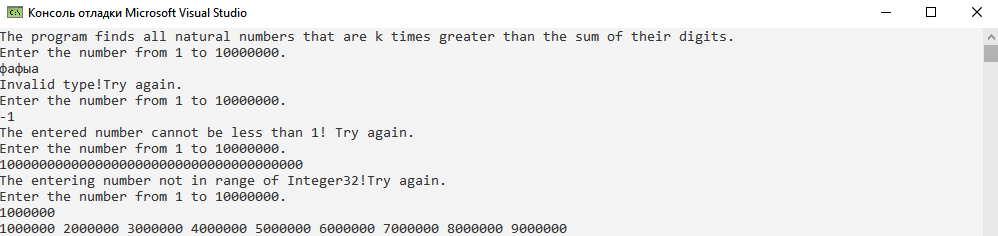
again.\n",minK);  
 else if (isNumber && k > maxK )  
 System.err.printf("The entered number cannot be more than %d! Try

again.\n",maxK);  
 else if (isNumber)  
 isIncorrect = false;  
 } while(isIncorrect);  
 return k;  
 }  
 public static void output(int naturalNumber)  
 {  
 System.out.print(naturalNumber + " ");  
 }  
  
 public static int digitsSum(int naturalNumber)  
 {  
 int sum = 0;  
 while (naturalNumber != 0)  
 {  
 sum += naturalNumber % 10;  
 naturalNumber /= 10;  
 }  
 return sum;  
 }  
  
 public static int findMaxNumber(int k)  
 {  
 int maxNumber = 1;  
 int countOfDigits = 1;  
 while (countOfDigits < maxDigits && k \* maxDigit \* countOfDigits++ >=

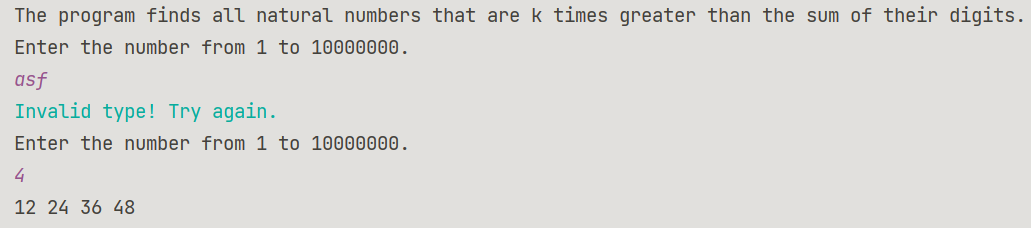
maxNumber)  
 {  
 maxNumber \*= 10;  
 };  
 return maxNumber /= 10;  
 }  
  
 public static void findNumbers(int k)  
 {  
 int naturalNumber = k;  
 int maxNumber = findMaxNumber(k);  
 while (naturalNumber < maxNumber)  
 {  
 if (naturalNumber == digitsSum(naturalNumber) \* k)  
 output(naturalNumber);  
 naturalNumber += k;  
 }  
 }  
  
 public static void main(String[] args) {  
 int k = 0;  
 k = input(k);  
 findNumbers(k);  
 }  
}

**Скриншоты:**

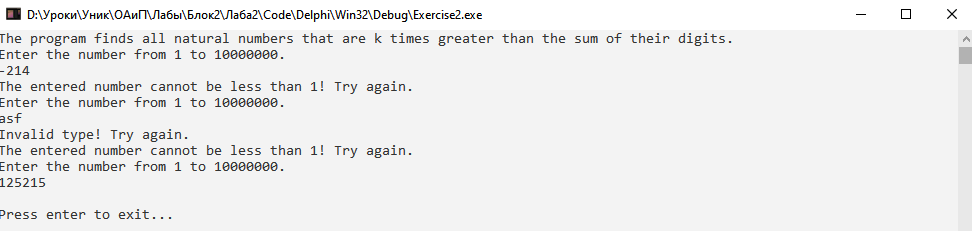
**C++:**

****

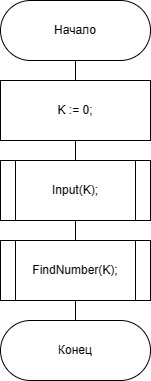
**Delphi:**

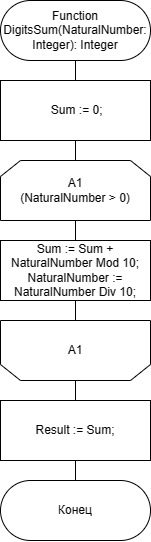


**Java:**

****

**Блок-схема:**



****

