Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра программного обеспечения информационных технологий

Дисциплина: Основы алгоритмизации и программирования (ОАиП)

ОТЧЕТ

по разминке №4

Тема работы: Длинная арифметика: сложение

Выполнил

студент: гр. 251003 Панкратьев Е.С.

Проверил: Фадеева Е.П.

Минск 2022

**СОДЕРЖАНИЕ**

[1 Исходная постановка задачи 3](#_Toc120385808)

[2 Дополненная постановка задачи 4](#_Toc120385809)

[3 Методика решения 5](#_Toc120385810)

[3.1 Описание оператора try..except..end и функции Pos 5](#_Toc120385811)

[3.1.1 Оператор try..except..end 5](#_Toc120385812)

[3.1.2 Функция Pos и использование константной строки 5](#_Toc120385813)

[3.2 Условия ввода 5](#_Toc120385814)

[3.3 Проверка введенных данных и запись числа в массив 6](#_Toc120385815)

[3.3.1 Проверка ввода системы счисления и количества чисел для сложения 6](#_Toc120385816)

[3.3.2 Проверка корректности введенного числа и его запись в массив 6](#_Toc120385817)

[3.4 Сложение чисел 6](#_Toc120385818)

[3.5 Вывод суммы всех чисел 7](#_Toc120385819)

[4 Структура данных 8](#_Toc120385820)

[Приложение А 9](#_Toc120385821)

[Приложение Б 16](#_Toc120385822)

# Исходная постановка задачи

Даны два целых числа в десятичной системе счисления до 50 цифр, которые >=0. Необходимо найти их сумму.

# Дополненная постановка задачи

Вводится необходимая система счисления NS (максимальная система счисления – 20-ая) и количество чисел для сложения n (не больше 50). Числа должны быть целыми, не более 256 символов и >=0. Необходимо найти их сумму.

# [Методика решения](#_Toc83996305)

## Описание оператора try..except..end и функции Pos

### Оператор try..except..end

Оператор try..except..end имеет вид:

try  
 операторы;  
except  
  блок обработки исключений;  
end;

Выполнение блока начинается с секции try, при отсутствии исключительных ситуаций только она и выполняется. Секция except получает управление в случае возникновения исключительной ситуации. После обработки выполняются операторы, стоящие после end.

### Функция Pos и использование константной строки

Функция Pos имеет вид:

Pos(Substr, S)

Функция возвращает индекс символа, с которого начинается первое вхождение подстроки Substr в строку S. Eсли заданная подстрока не встречается в заданной строке, то функция вернет 0.

В данной задачи в роли подстроки Substr применяется символ из введенной строки; в роли строки S применяется константная строка, в которой индекс элемента уменьшенный на единицу соответствует численному значению символа в этом элементе:

NSAlphabet = ''

## Условия ввода

Условия ввода:

* система счисления NS должна быть целочисленной, больше либо равно 2 и меньше либо равно 20;
* количество чисел для сложения n должно быть целочисленным, больше 1 и меньше либо равно 50;
* при вводе чисел, длина должна быть не больше 256 символов и значение каждого символа должно соответствовать введенной системе счисления NS.

## Проверка введенных данных и запись числа в массив

### Проверка ввода системы счисления и количества чисел для сложения

Проверка ввода системы счисления NS и количества чисел для сложения n происходит с помощью цикла с предусловием repeat..until, чтобы при вводе некорректных данных пользователь заново заполнял их. В теле цикла с помощью оператора try..except..end (описание оператора см. [главу 3.1.1](#_Оператор_try..except..end)) проверяем целочисленность; оператором if проверяет принадлежность заданному промежутку.

### Проверка корректности введенного числа и его запись в массив

Проверка корректности введенного числа происходит с помощью цикла с предусловием repeat..until, чтобы при вводе некорректных данных пользователь заново заполнял их. В теле цикла с помощью оператора if проверяем условие: длина должна быть не больше 256 символов.

Так в данной задаче необходимо работать с очень большими числами (которые не входят ни в один тип данных), необходимо записывать каждую цифру числа в отдельный элемент массива. Так как в результате сложения количество цифр суммы может быть больше, чем количества цифр изначальных значений, то будем записывать числа с конца для увеличения размера числа.

Перед началом записи числа в массив заполним все его элементы 0 с помощью цикла с параметром for. Далее для записи числа в массив используем цикл с предусловием while, в котором условие входа – i <= len и flag = false (i – счетчик цикла, len – длина введенной строки, flag – индикатор проверки правильности ввода). В теле цикла для перевода из символа в его численное значение используем константную строку NSAlphabet (в которой индекс элемента уменьшенный на единицу соответствует численному значению символа в этом элементе) и процедуру Pos (описание процедуры и константной строки см. [главу 3.1.2](#_Функция_Pos_и)). Далее проверяем корректность (символ должен соответствовать введенной системе счисления) с помощью оператора if. Если соответствует, то численное значение символа записываем в элемент массива.

## Сложение чисел

Сложение чисел происходит поэтапно: сначала складываются два числа, потом сумма этих двух чисел складывается с третьим числом, потом сумма предыдущих чисел со следующим числом и так далее.

Сложение двух чисел происходит с помощью метода сложения столбиком. Так как числа записаны с конца, складывать элементы начнем сначала. Для нахождения суммы и переноса для следующего разряда используем операторы div и mod. При операции Sum mod NS (где Sum – сумма элементов и переноса, NS – заданная система счисления) получится сумма двух элементов в заданной системе счисления; при операции Sum div NS (где Sum – сумма элементов и переноса, NS – заданная система счисления) получится перенос в следующий разряд.

## Вывод суммы всех чисел

Для перевода из численного значения символа в символ используем константную строку NSAlphabet, так как данная строка хранит символы, в которых индекс элемента уменьшенный на единицу соответствуют численному значению символа в этом элементе.

Сумма записана с конца, значит вывод элементов массива следует начинать с конца. Для вывода суммы используем цикл с параметром for. Выводить символы будем из константной строки NSAlphabet по индексу текущего значения элемента из массива суммы, увеличенного на единицу.

# Структура данных

Таблица 1 – Данные

|  |  |  |
| --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение |
| MaxSize | Integer | Максимальный допустимый размер числа |
| MaxAmount | Integer | Максимальное допустимое количество чисел для сложения |
| NSAlphabet | String | Набор символов для перевода символа в его численное значение и наоборот. |
| Num1 | Array [1..(MaxSize + (MaxAmount div 10) + 1)] of ShortInt | Массив первого числа |
| Num2 | Array[1..MaxSize] of ShortInt | Массив второго числа |
| str | String | Строка для записи числа в массив |
| MaxNS | ShortInt | Максимальная допустимая система счисления |
| NS | Integer | Введенная система счисления |
| Amount | Integer | Введенное количество чисел для сложения |
| j | ShortInt | Счетчик цикла |
| Sum | ShortInt | Сумма цифр текущего разряда |
| Carry | ShortInt | Перенос в следующий разряд |
| Len1 | Word | Длина первого числа |
| Len2 | Word | Длина второго числа |
| Size | Word | Длина суммы двух чисел |
| i | Word | Счетчик цикла |
| flag | Boolean | Индикатор проверки правильности ввода |

Приложение А

(обязательное)

Исходный код программы

Program WarmUp4;

{

The program should calculate the sum of n numbers in

the entered number system

}

{$APPTYPE CONSOLE}

Uses

System.SysUtils;

Const

MaxSize=256;

MaxAmount=50;

NSAlphabet = '0123456789ABCDEFGHIJ';

MaxNS = length(NSAlphabet);

//MaxSize - maximum amount of digits for entered

//numbers

//MaxAmount - maximum amount of numbers

//NSAlphabet - transfer between symbols and numbers

//MaxNS - maximum number system

Var

Num1 :array[1..(MaxSize + (MaxAmount div 10) + 1)] of

ShortInt;

Num2 :array[1..MaxSize] of ShortInt;

str :string;

NS, Amount : Integer;

j, Sum, Carry : ShortInt;

Len1, Len2, Size, i: Word;

flag: boolean;

//Num1 - array of digits of the first number (further

//the sum of the two previous numbers)

//Num2 - array of digits of the second numbers (with

//which need to add the first number)

//str - string variable for writing numbers

//NS - number system

//Amount - amount of numbers

//Sum - sum of digits

//Carry - сarry 1 (if there is) to the next element

//Len1 - the length of the first entered number

//Len2 - the length of the second entered number

//Size - the size of the sum

//i,j - cycle counter

//flag - flag to confirm the correctness of entering

//numbers

Begin

Writeln('Enter number system (maximum number system is

',MaxNS,', minimal is 2). The program will

calculate their sum.');

//Cycle with postcondition for entering correct data.

Repeat

//Initialize the flag

flag:= False;

//Validating the correct input data type

Try

Readln(NS);

Except

Writeln('Wrong input of number system! It must be

an integer');

flag:= True;

End;

//Validate Range

if ((NS > MaxNS) or (NS < 2)) and (flag = False) then

begin

Writeln('Wrong input of number system! It must be

>=2 and <=',MaxNS);

flag:= True;

end;

Until flag = False;

Writeln;

Writeln('Enter amount of numbers (no more than ',

MaxAmount,' and more than 1)');

//Cycle with postcondition for entering correct data.

Repeat

//Initialize the flag

flag:= False;

//Validating the correct input data type

Try

Readln(Amount);

Except

Writeln('Wrong input of amount of numbers! It must

be an integer');

flag:= True;

End;

//Validate Range

if ((Amount > MaxAmount) or (Amount <= 1)) and (flag

= False) then

begin

Writeln('Wrong input of amount of numbers! It must

be >1 and <=',MaxAmount);

flag:= True;

end;

Until flag = False;

//Declaring available symbols and their value

Writeln;

Writeln('Available symbols on the ',NS,'th number

system and their number system:');

for i := 0 to (NS - 1) do

Writeln('Symbol ', NSAlphabet[i+1],' Value = ',i);

Writeln;

Writeln('Enter numbers (no more than ',MaxSize,'

digits).');

//Cycle with postcondition for entering correct data.

Repeat

//Initialize the flag

flag:= False;

//Read the first entered number and check for

//correctness.

Readln(str);

//Find length of the first number

Len1:= length(str);

//Checking the correct length

if Len1 > MaxSize then

begin

Writeln('Wrong input of number! The length of the

number must be no more than ',MaxSize,'

digits.');

flag:= True;

end

//Else if length >1, the first digit cannot be 0

//(in the mirrored view it is last)

else if (Len1 > 1) and (str[1] = '0') then

begin

Writeln('Wrong input of number! The first digit of

a number cannot be 0');

flag:= True;

end

//Else writing a number to an array and checking for

//valid symbols

else

begin

//Reset the first number for the input

for i := 1 to length(Num1) do

Num1[i]:= 0;

//Write the first entered number in mirrored view

//to an array

i:= 1;

while (i <= Len1) and (flag = False) do

begin

//Transfer to numerical value (-1 because

//numbering in delphi starts from 1)

Num1[i]:= Pos(str[Len1-i+1], NSAlphabet) - 1;

//Checking for correct input in the NS. Num1[i]

//will be <0 if the symbol is not in NSAlphabet

if (Num1[i] < 0) or (Num1[i] >= NS) then

begin

Writeln('Wrong input of number! Namely, wrong

input of symbols! See available symbols

above!');

flag:= True;

end;

//Modernize i

i:= i + 1;

end;

end;

Until flag = False;

//The cycle go (Amount - 1) times to add all the numbers

for j := 1 to (Amount - 1) do

begin

Writeln('+');

//Cycle with postcondition for entering correct

//data.

Repeat

//Initialize the flag

flag:= False;

//Read the second entered number and check for

//correctness.

Readln(str);

//Find length of the second number

Len2:= length(str);

//Checking the correct length

if Len2 > MaxSize then

begin

Writeln('Wrong input of number! The length of

the number must be no more than

',MaxSize,' digits.');

flag:= True;

end

//Else if length >1, the first digit cannot be 0

//(in the mirrored view it is last)

else if (Len2 > 1) and (str[1] = '0') then

begin

Writeln('Wrong input of number! The first digit

of a number cannot be 0');

flag:= True;

end

//Else writing a number to an array and checking

//for valid symbols

else

begin

//Reset the second number for the input

for i := 1 to length(Num2) do

Num2[i]:= 0;

//Write the second number in mirrored view to an

//array

i:= 1;

while (i <= Len2) and (flag = False) do

begin

//Transfer to numerical value (-1 because

//numbering in delphi starts from 1)

Num2[i]:= Pos(str[Len2-i+1], NSAlphabet) - 1;

//Checking for correct input in the NS. Num2[i]

//will be <0 if the symbol is not in NSAlphabet

if (Num2[i] < 0) or (Num2[i] >= NS) then

begin

Writeln('Wrong input of number! Namely,

wrong input of symbols! See available

symbols above!');

flag:= True;

end;

//Modernize i

i:= i + 1;

end;

end;

Until flag = False;

//The larger size assign in a Size (this is the size

//of the sum of the numbers)

if Len1 > Len2 then

Size:= Len1

else

Size:= Len2;

//Start adding digits separately in the cycle

for i := 1 to Size do

begin

//Starting to add the last digits of the numbers

//(in the mirrored view it is first)

//and add the carry (if there is).

Sum:= Carry + Num1[i] + Num2[i];

//The modulo of the Sum by NS is the last (in the

//mirrored it is first) digit of sums (array Num1).

Num1[i]:= Sum mod NS;

//The integer part of dividing by number system is

//the carry that will go to the next element

Carry:= Sum div NS;

//If there is a carry on the last digit, then

//increase the size of sum by 1 and add a carry to

//the next element

if (i = Size) and (Carry = 1) then

begin

Size:= Size + 1;

Num1[i+1]:= Carry;

end;

end;

//Update the length of the first number

Len1:= Size;

//Reset the carry for the following operations

Carry:= 0;

end;

//Write the answer, mirroring the back

//Transfer to symbolic value

Writeln('The sum of the numbers is:');

for i := Size downto 1 do

Write(NSAlphabet[Num1[i]+1]);

Readln;

End.

Приложение Б

(обязательное)

Тестовые наборы

**Тестовая ситуация: некорректный ввод данных**

Тест 1

Исходные данные: Некорректный ввод системый счисления

Ожидаемый результат: Повторная попытка

Полученный результат:

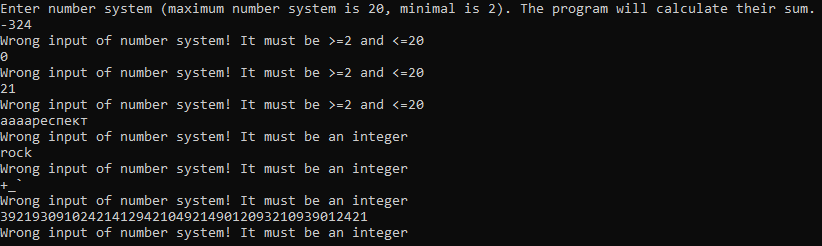


Рисунок 1 – Результаты расчетов

Тест 2

Исходные данные: Некорректный ввод количества чисел для сложения

Ожидаемый результат: Повторная попытка

Полученный результат:

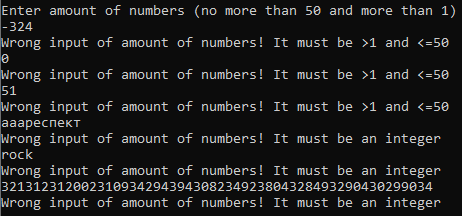


Рисунок 2 – Результаты расчетов

Тест 3

Исходные данные: Некорректный ввод первого числа

Ожидаемый результат: Повторная попытка

Полученный результат:

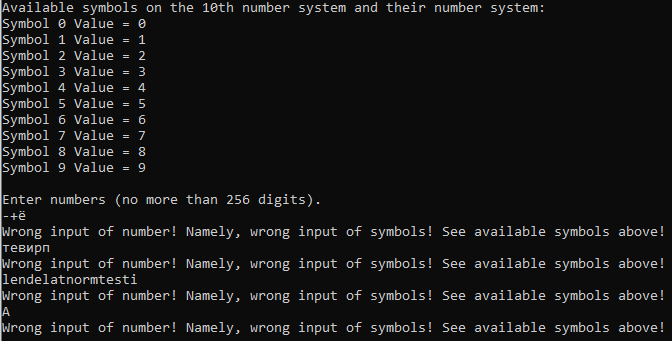


Рисунок 3 – Результаты расчетов

Тест 4

Исходные данные: Некорректный ввод второго числа

Ожидаемый результат: Повторная попытка

Полученный результат:

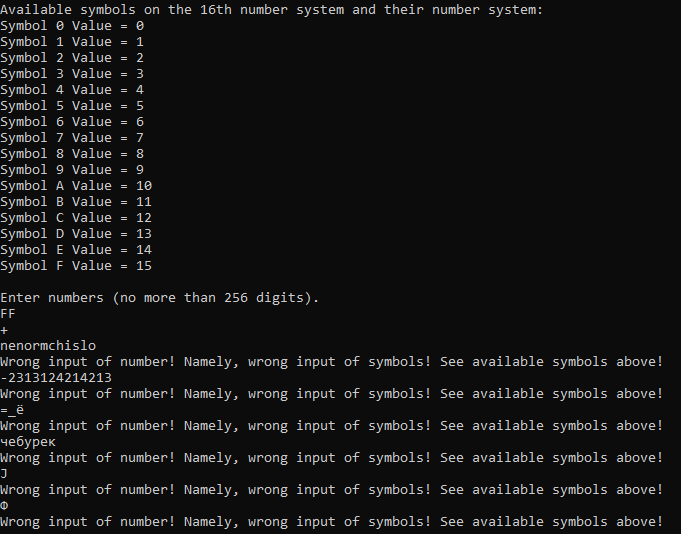


Рисунок 4 – Результаты расчетов

**Тестовая ситуация: корректный ввод данных**

Тест 5

Исходные данные: Система счисления: 16; количество чисел для сложения: 2.

Ожидаемый результат:

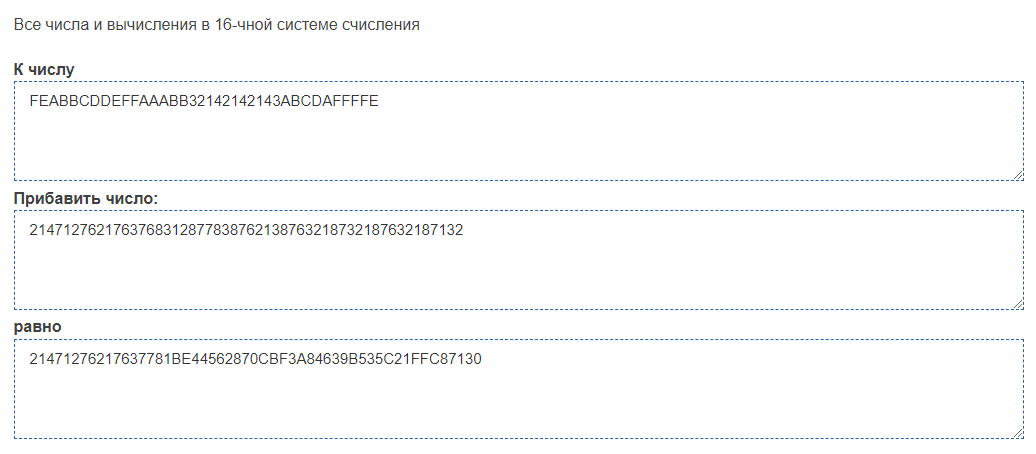


Рисунок 5 – Ожидаемый результат (ilovecalc)

Полученный результат:

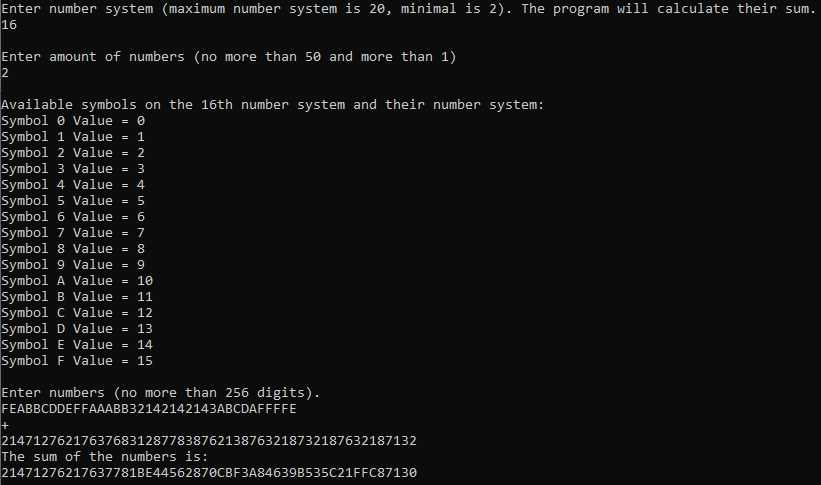


Рисунок 6 – Результаты расчетов

Тест 6

Исходные данные: Система счисления: 2; количество чисел для сложения: 2.

Ожидаемый результат:

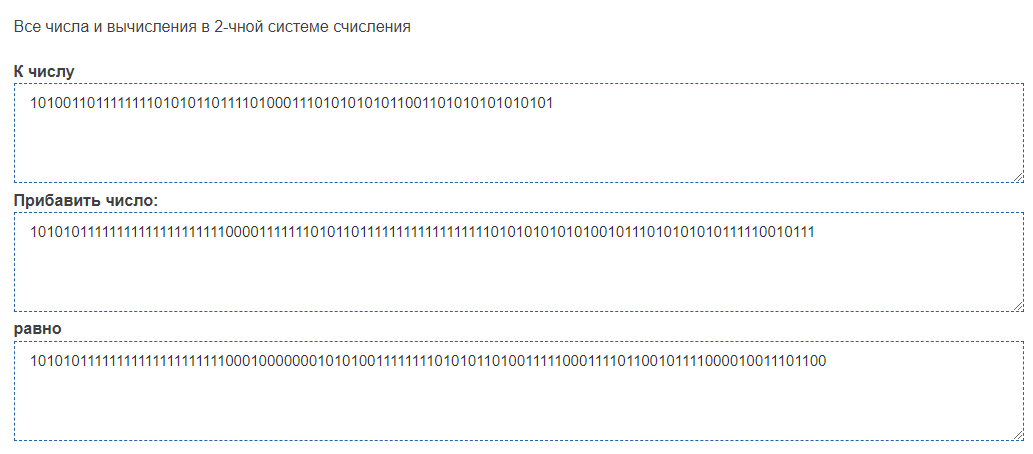


Рисунок 7 – Ожидаемый результат (ilovecalc)

Полученный результат:

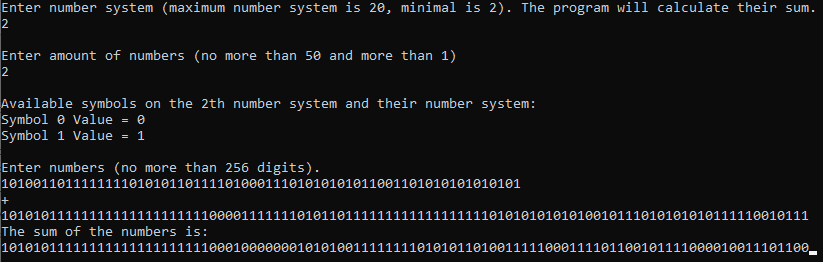


Рисунок 8 – Результаты расчетов

Тест 7

Исходные данные: Система счисления: 20; количество чисел для сложения: 2.

Ожидаемый результат:



Рисунок 9 – Ожидаемый результат (ilovecalc)

Полученный результат:

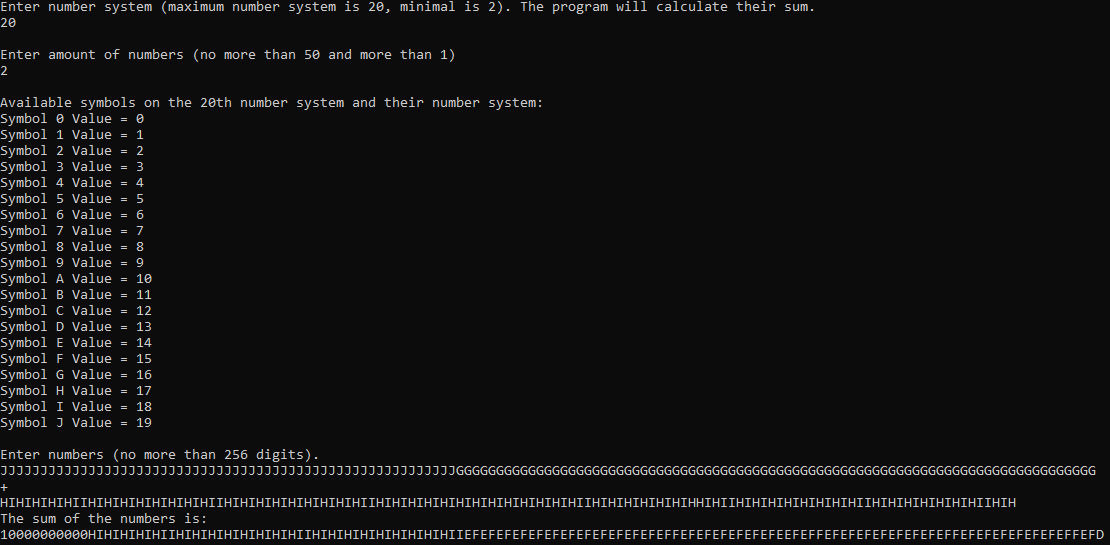


Рисунок 10 – Результаты расчетов

Тест 8

Исходные данные: Система счисления: 10; количество чисел для сложения: 2.

Ожидаемый результат:

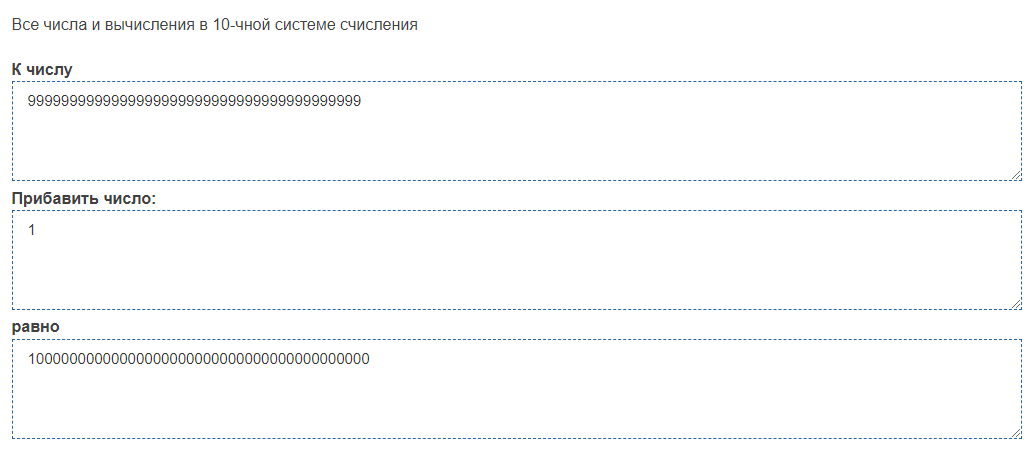


Рисунок 11 – Ожидаемый результат (ilovecalc)

Полученный результат:

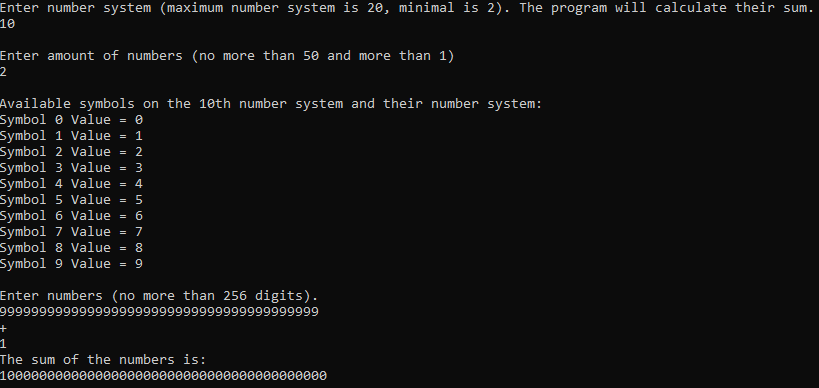


Рисунок 12 – Результаты расчетов

Тест 9

Исходные данные: Система счисления: 10; количество чисел для сложения: 6.

Ожидаемый результат:

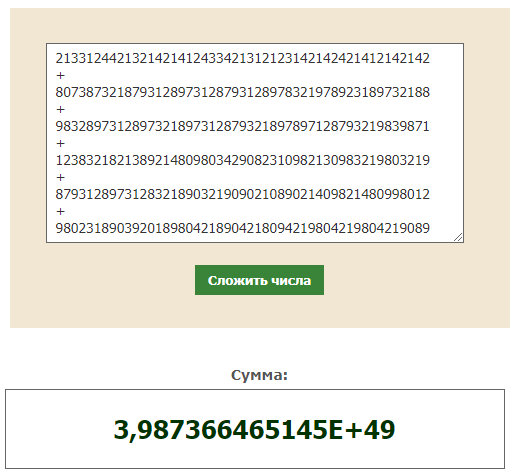


Рисунок 13 – Ожидаемый результат (tools.mega8)

Полученный результат:

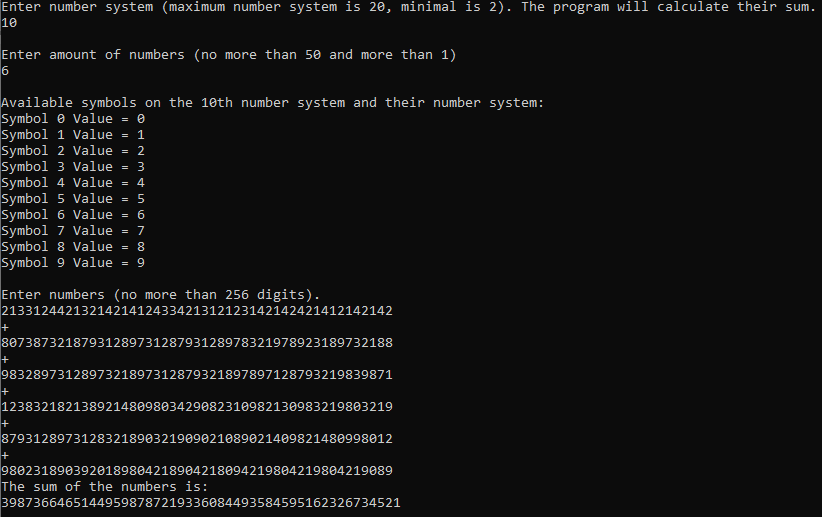


Рисунок 14 – Результаты расчетов