Содержание

[1 Постановка задачи 3](#_Toc116916278)

[1.1 Исходное условие задачи 3](#_Toc116916279)

[1.2 Модифицированное условие задачи 3](#_Toc116916280)

[1.3 Исходные данные 3](#_Toc116916281)

[2 Структура данных 5](#_Toc116916282)

[3 Детали и методика решения задачи 6](#_Toc116916283)

[3.1 Детали задачи 6](#_Toc116916284)

[3.2 Используемые формулы 6](#_Toc116916285)

[3.3 Проверка исходных данных с помощью цикла с постусловием 7](#_Toc116916286)

[3.4 Использование Try … Except … End; 9](#_Toc116916287)

[4 Результаты расчетов 10](#_Toc116916288)

[Приложение А 11](#_Toc116916289)

[Приложение Б 17](#_Toc116916290)

# Постановка задачи

## Исходное условие задачи

Есть два целых натуральных числа, состоящих до 50 цифр. Вывести на экран их разность.

## Модифицированное условие задачи

Есть два целых числа (Num\_1 и Num\_2), состоящих до 255 цифр, в определенной системе счисления. Вывести на экран разность чисел ArrayOfSub.

## Исходные данные

Программа получает от пользователя :

* систему счисления;
* уменьшаемое;
* вычитаемое.

Вывод данных происходит в следующем виде:

Разность чисел Num\_1 и Num\_2 равна ArrayOfSub

# Структура данных

Таблица 1 – Данные

|  |  |  |
| --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение |
| Num\_1 | Shortstring | Уменьшаемое |
| Num\_2 | Shortstring | Вычитаемое |
| Check | Shortstring | Вспомогательная переменная для хранения знака разности |
| I | Smallint | Вспомогательная переменная для подсчетов |
| J | Smallint | Вспомогательная переменная для подсчетов |
| К | Smallint | Вспомогательная переменная для подсчетов |
| RunkOfNum | Smallint | Промежуточное значение вычитания n-х разрядов чисел |
| Border | Smallint | Вспомогательная переменная для ограничения цикла |
| Z1 | Smallint | Знак первого числа |
| Z2 | Smallint | Знак второго числа |
| N | Smallint | Система счисления |
| LengthA | Integer | Длина ArrayOfSub |
| ArrayOfSub | Array[1..257] of Smallnt | Массив для вычитания Num\_1 и Num\_2 |
| L | Boolean | Переменная для проверки на ошибки |

# Детали и методика решения задачи

## Детали задачи

Детали задачи таковы:

* числа должны быть введены без знака, если они положительные;
* числа должны быть целыми;
* числа могут быть отрицательными;
* система счисления должна быть от 2-ричной до 62-ричной, так как дальше в таблице ASCII используются другие символы;
* числа должны быть введены со знаком, если они отрицательные;
* количество цифр в числе должно быть не больше 255;
* второе число может быть больше, чем первое;
* буквы в системах счисления, предполагающих их, вводятся на латинице.

## Методика решения задачи

Проанализировав условие и детали задачи, можно сделать вывод, что для решения необходимы следующие пункты:

* ввод числа должен быть в тип Shortstring, т. к. тип Integer может содержать числа в промежутке от -231 до 231 - 1 (7 символов), a Shortstring – 256 символов;
* использования функции ord для получения значения кода цифры в таблице ASCII;
* «перевод в десятичные числа» (например, дана 16-тиричная система счисления, где для записи чисел используются как цифры, так и буквы; во время «перевода в десятичные числа» от кода, в зависимости буква это или цифра, будет отниматься разница кода и значения в десятичной системе счисления, а именно для букв – 55, а для цифр – 48):

Таблица 2 – Пример к «переводу в десятичные числа»

|  |  |  |
| --- | --- | --- |
| Значение цифры числа | Код в ASCII | Значение в десятичной системе |
| А | 65 | 10 |
| 2 | 50 | 2 |

* использование массива типа smallint для подсчётов;
* проверка на длины чисел и их знаки, так как при вычитании необходимо от большего вычитать меньшее, и изменение знака разности;
* наличие переменных z1 и z2, которые будут отвечать за знаки (если равна 1 – число положительное; -1 – отрицательное) при выводе начальных значений чисел и за знак их разности.

Сами расчёты будут проходить по следующим пунктам:

1. Проверка чисел на их длину (если они равны, то нахождение большего числа).
2. Добавление первого числа в обратном порядке в конец массива, где каждая цифра будет записана как её код в таблице ASCII, «переведенный в десятичные числа».
3. Вычитание второго числа в обратном порядке с элементом массива с конца, где каждая цифра будет записана как её код в таблице ASCII, «переведенный в десятичные числа», что будет записано в переменную RunkOfNum.
4. Если разряд первого числа меньше, чем RunkOfNum, то получение разряда от следующих (если следующий – 0, то происходит поиск ненулевого разряда, получая также разряды).
5. Если RunkOfNum равен N , то к следующему элементу массива прибавляется 1 (как переход к новому разряду, т.к. расчеты происходят в десятичной системе); если равен -N – прибавление -1.

## Проверка исходных данных с помощью цикла с постусловием

При проверке исходных данных использовался цикл с постусловием, так как в таком случае цикл будет запущен хотя бы один раз:

Repeat

Write('Введите первое число: ');

readln(Num\_1);

j := length(Num\_1);

for I := (LengthA-1) downto (LengthA-length(Num\_1)) do

Begin

if (ord(Num\_1[J]) < 48) or

((ord(Num\_1[J]) >= 58) and (ord(Num\_1[J]) <= 64))

then

Begin

Writeln('Вы ввели неверный символ! Попробуйте ещё

раз');

L := False;

End

Else

L := True;

if ord(Num\_1[J]) <= 57 then

ArrayOfSum[I] := ord(Num\_1[J]) - 48

Else

ArrayOfSum[I] := ord(Num\_1[J]) - 55;

j := j - 1;

End;

Until L;

С помощью этого цикла мы повторяем вывод фразы «Введите первое число: », получение данных от пользователя, если было правильно введено значение, иначе фразу «Вы ввели неверный символ! Попробуйте еще раз!» до тех пор, пока полученные значения не будет выполнять всем требованиям. Так же происходит и с вводом второго числа, и системы счисления.

# Результаты расчетов

Вследствие выполнения программы на экран выводятся следующие результаты расчетов:

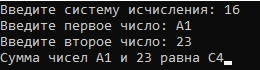


Рисунок 1– Результаты расчётов

Приложение А

(обязательное)

Исходный код программы

program Вычитание\_СИ;

{

Substracting two numbers (around 255 digits in each)

}

uses

System.SysUtils;

var

Num\_1, Num\_2, Check: shortstring;

N, I, j, Z1, Z2, K, RankOfNum: smallint;

L: boolean;

ArrayOfSub: array [1 .. 255] of smallint;

// Num\_1, Num\_2 - numbers, which must be added

// I, J, K - auxiliary variables for counting

// Z1, Z2 - variable of sign of numbers

// Check - variable for keeping the

// value of the first number

// ArrayOfSub - array to substract Num\_1 and Num\_2

// L - variable to check for errors

// N - the numbering system

const

LengthA = length(ArrayOfSub) + 1;

// LengthA - Length ArrayOfSum

begin

// Requiring and checking the Numbering system

Repeat

Write('Введите систему счисления: ');

readln(N);

if (N < 2) or (N > 62) then

Begin

Writeln('');

L := False;

End

Else

L := True;

Until L;

// Requiring numbers

Repeat

Write('Введите первое число: ');

readln(Num\_1);

// Checking for sign of the first number

if Num\_1[1] = '-' then

Begin

Delete(Num\_1, 1, 1);

Z1 := -1;

End

else

Z1 := 1;

Write('Введите второе число: ');

readln(Num\_2);

// Checking for sign of the second number

if Num\_2[1] = '-' then

Begin

Delete(Num\_2, 1, 1);

Z2 := -1;

End

else

Z2 := 1;

if (length(Num\_1)<length(Num\_2)) and (Z1=1) and (Z2=1)

then

Begin

Check := Num\_2;

Num\_2 := Num\_1;

Num\_1 := Check;

// As we swap, then the final sign will be negative

Check := '-';

End;

if (length(Num\_1)>length(Num\_2)) and (Z1=-1) and (Z2=-1) then

// As we swap, then the final sign will be negative

Check := '-';

j := length(Num\_1);

// Trying convert string of the first

// number to array of it

begin

// Checking for the biggest number

// and adding it to array

if (length(Num\_1)=length(Num\_2)) and (Z1=1) and (Z2=1)

then

Begin

Repeat

I := I + 1;

// Cheking for right symbols

if (ord(Num\_1[j])<48) or ((ord(Num\_1[j])>=58) and

(ord(Num\_1[j])<=64)) then

Begin

Writeln('Вы ввели неверный символ! Попробуйте ещё

раз');

L := False;

End

Else

L := True;

Until (ord(Num\_1[I])<ord(Num\_2[I])) or

(I=length(Num\_1));

if I < length(Num\_1) then

Begin

Check := Num\_1;

Num\_1 := Num\_2;

Num\_2 := Check;

// As we swap, then the final sign will be negative

Check := '-';

End;

End;

j := length(Num\_1);

for I := (LengthA-1) downto (LengthA-length(Num\_1)) do

Begin

// Adding digits of the first number

// as like they are in 10th numbering system

if ord(Num\_1[j]) <= 57 then

ArrayOfSub[I] := ord(Num\_1[j]) - 48

Else

ArrayOfSub[I] := ord(Num\_1[j]) - 55;

j := j - 1;

End;

L := True;

end;

Until L;

j := length(Num\_2);

I := (LengthA - 1);

While I >= (LengthA - length(Num\_2)) do

Begin

// A new rank of the sub

If ArrayOfSub[I] = -N then

Begin

ArrayOfSub[I] := 0;

ArrayOfSub[I-1] := ArrayOfSub[I-1] - 1;

End;

If ArrayOfSub[I] = N then

Begin

ArrayOfSub[I] := 0;

ArrayOfSub[I-1] := ArrayOfSub[I-1] + 1;

End;

// If the second number isn't over

If j <> 0 then

begin

// as like digits of the second number are

// in 10th numbering system

if ord(Num\_2[j]) <= 57 then

RankOfNum := ord(Num\_2[j]) - 48

Else

RankOfNum := ord(Num\_2[j]) - 55;

// If we need to get a rank from the next one

If (ArrayOfSub[I]<RankOfNum) and (Z2=1) and (Z1=1) then

Begin

// if the next on is zero and we again need a rank

if ArrayOfSub[I-1] = 0 then

Begin

// Trying to find the rank, which isn't zero

while (ArrayOfSub[I-K] <= 0) and

((LengthA-I+K+1) <= length(Num\_1)) do

Begin

K := K + 1;

ArrayOfSub[I-K] := ArrayOfSub[I-K] - 1;

ArrayOfSub[I+1-K] := ArrayOfSub[I+1-K] + N;

End;

End

Else

Begin

// Getting a rank from th next one and

// substracting it

ArrayOfSub[I-1] := ArrayOfSub[I-1] - 1;

ArrayOfSub[I] := ArrayOfSub[I] + N;

End;

End;

ArrayOfSub[I] := Z1\*ArrayOfSub[I] –

Z2\*StrtoInt(Num\_2[j]);

// If the digit is negative to deleting the sign

if (I = (LengthA-length(Num\_2))) and (ArrayOfSub[I]<0)

then

Check := '-';

j := j - 1;

end;

I := I - 1;

End;

// Recreating the sign of the numbers

if Z1 = -1 then

Insert('-', Num\_1, 1);

if Z2 = -1 then

Insert('-', Num\_2, 1);

Write('Разность чисел ', Num\_1, ' и ', Num\_2, ' равна ',

Check);

// Finding a border of length of the sub

If length(Num\_2) > length(Num\_1) then

j := (LengthA - length(Num\_2))

Else

j := (LengthA - length(Num\_1));

if ArrayOfSub[j] = 0 then

While ArrayOfSub[j] = 0 do

j := j + 1;

// Outputting the sub in the required numbering system

for I := j to (LengthA - 1) do

if abs(ArrayOfSub[I]) <= 9 then

Write(char((abs(ArrayOfSub[I]) + 48)))

Else

Write(char((abs(ArrayOfSub[I]) + 55)));

readln;

end.

Приложение Б

(обязательное)

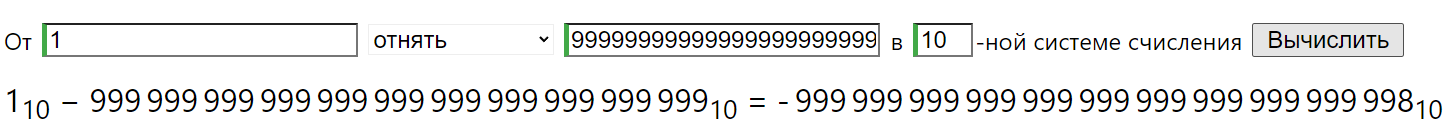
Тестовые наборы

Тест 1

Исходные данные:

* система счисления – 10;
* первое число – 1;
* второе число – 999999999999999999999999999999999.

Ожидаемый результат:



Полученный результат:

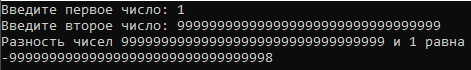


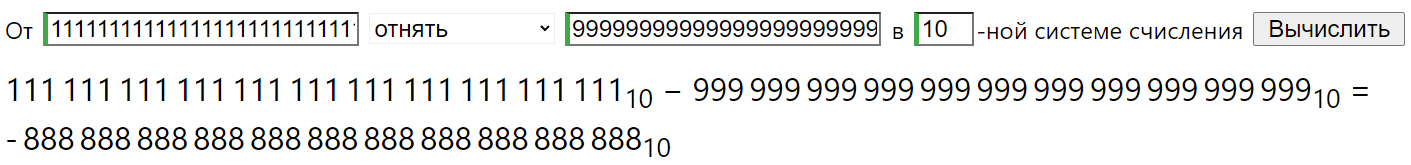
Рисунок 2 – Тест 1

Тест 2

Исходные данные:

* система счисления – 10;
* первое число – 111111111111111111111111111111111;
* второе число – 999999999999999999999999999999999.

Ожидаемый результат:



Полученный результат:

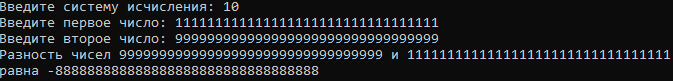


Рисунок 3 – Тест 2

Тест 3

Исходные данные:

* система счисления – 10;
* первое число – 999999999999999999999999999999999;
* второе число – 999999999999999999999999999999999.

Ожидаемый результат:

0

Полученный результат:

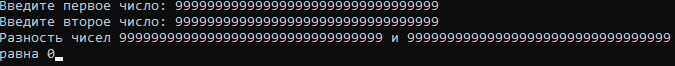


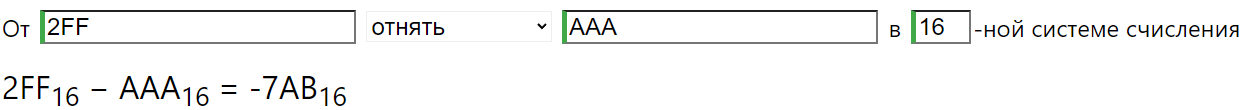
Рисунок 4 – Тест 3

Тест 4

Исходные данные:

* система счисления – 16;
* первое число – 2FF;
* второе число – AAA.

Ожидаемый результат:



Полученный результат:

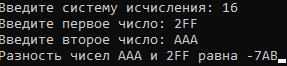


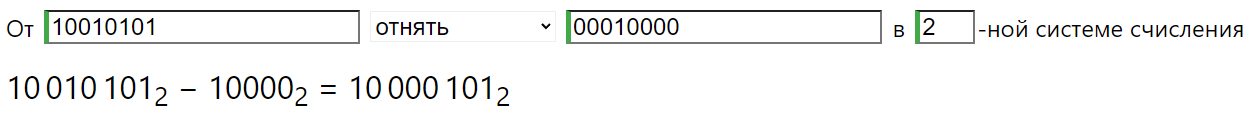
Рисунок 5 – Тест 4

Тест 5

Исходные данные:

* система счисления – 2;
* первое число – 1001 0101;
* второе число – 0001 0000.

Ожидаемый результат:



Полученный результат:

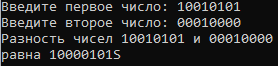


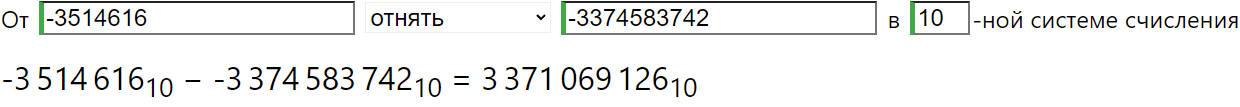
Рисунок 6 – Тест 5

Тест 6

Исходные данные:

* система счисления – 10;
* первое число – -3514616;
* второе число – -3374583742.

Ожидаемый результат:



Полученный результат:

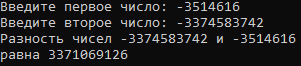


Рисунок 7 – Тест 6

Тест 7

Таблица 3 – Тестовые ситуации

|  |  |  |
| --- | --- | --- |
| Тест | Тестовые ситуации (вместе с расчётами) | Проверка полученного результата |
| 1. |  | [См. Детали задачи](#_Детали_задачи) |
| 2. |  | [См. Детали задачи](#_Детали_задачи) |
| 3. |  | [См. Детали задачи](#_Детали_задачи) |
| 4. |  | [См. Детали задачи](#_Детали_задачи) |
| 5. |  | [См. Детали задачи](#_Детали_задачи) |

Продолжение таблицы 5

|  |  |  |
| --- | --- | --- |
| Тест | Тестовые ситуации (вместе с расчётами) | Проверка полученного результата |
| 6. |  | [См. Детали задачи](#_Детали_задачи) |