Содержание

[1 Постановка задачи 3](#_Toc119330995)

[1.1 Исходное условие задачи 3](#_Toc119330996)

[1.2 Модифицированное условие задачи 3](#_Toc119330997)

[1.3 Исходные данные 3](#_Toc119330998)

[2 Структура данных 4](#_Toc119330999)

[3 Детали и методика решения задачи 5](#_Toc119331000)

[3.1 Детали задачи 5](#_Toc119331001)

[3.2 Методика решения задачи 5](#_Toc119331002)

[3.3 Проверка исходных данных с помощью цикла с постусловием 6](#_Toc119331003)

[4 Результаты расчетов 8](#_Toc119331004)

[Приложение А 9](#_Toc119331005)

[Приложение Б 14](#_Toc119331006)

# Постановка задачи

## Исходное условие задачи

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

## Модифицированное условие задачи

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

## Исходные данные

Программа получает от пользователя:

* систему счисления;
* первое слагаемое;
* второе слагаемое.

Вывод данных происходит в следующем виде:

Сумма чисел Num\_1 и Num\_2 равна ArrayOfSum

# Структура данных

Таблица 1 – Данные

|  |  |  |
| --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение |
| Num\_1 | Shortstring | Число 1, которое нужно сложить со 2-м |
| Num\_2 | Shortstring | Число 2, которое нужно сложить со 1-м |
| I | Smallint | Вспомогательная переменная для подсчетов |
| J | Smallint | Вспомогательная переменная для подсчетов |
| RunkOfNum | Smallint | Промежуточное значение сложения 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 ArrayOfSum |
| ArrayOfSum | Array[1..257] of Smallnt | Массив для сложения Number\_1 и Number\_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 могут быть отрицательными, лишь при условии, что они вместе отрицательные;
* 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 исчисления, 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 исчисления, 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;
* 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 суммы.

Сами расчёты будут проходить по следующим пунктам:

1. Добавление первого числа в обратном порядке в конец массива, где каждая цифра будет записана как её код в таблице ASCII, «переведенный в десятичные числа».
2. Сложение 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.
3. Если RunkOfNum больше N-1, то к следующему элементу массива прибавляется 1 (как переход к новому разряду, т.к. расчеты происходят в десятичной системе); в противном случае – присваивание текущему элементу массива значение RunkOfNum.

## 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:

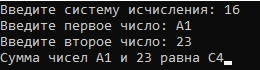


Рисунок 1– Результаты расчётов

Приложение А

(обязательное)

Исходный код программы

program Сложение\_СИ;

{

Adding two numbers (around 255 digit in each) in different

}

uses

System.SysUtils;

var

Num\_1, Num\_2: shortstring;

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

L: boolean;

ArrayOfSum: array [1 .. 257] of smallint;

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

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

// RunkOfNum - intermediate value of n-digit

// addition of numbers

// Z1, Z2 - variable of sign of number

// ArrayOfSum - array to add Num\_1 and Num\_2

// L - variable to check for errors

// N - the numbering system

const

LengthA = length(ArrayOfSum) + 1;

// LengthA - Length ArrayOfSum

begin

// Requiring and checking the Numbering system

Repeat

Write('Введите систему исчисления: ');

readln(N);

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

Begin

Writeln('');

L := False;

End

Else

L := True;

Until L;

// Requiring the first number

Repeat

Write('Введите первое число: ');

readln(Num\_1);

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

Begin

Z1 := -1;

Delete(Num\_1, 1, 1)

End

Else

Z1 := 1;

j := length(Num\_1);

// Trying convert string of the first number

// to array of it

// Inicialization from last digit to the first

// to array element from the end of it

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

Begin

// 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;

// Adding digits of the first number

// as like they are in 10th numbering system

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;

Repeat

Repeat

// Requiring the second number

Write('Введите второе число: ');

readln(Num\_2);

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

Begin

if Z1 = 1 then

Begin

Writeln('Числа должны быть одного знака!');

L := False;

End

Else

L := True;

Z2 := -1;

Delete(Num\_2, 1, 1);

End

Else

Begin

if Z1 = -1 then

Begin

Writeln('Числа должны быть одного знака!');

L := False;

End

Else

L := True;

Z2 := 1;

End;

Until L;

j := length(Num\_2);

// Seting the border of adding digits

Border := LengthA - length(Num\_2);

I := (LengthA - 1);

While I >= Border do

Begin

// Cheking for right symbols

if (ord(Num\_2[j])<48) or

((ord(Num\_2[j])>=58) and (ord(Num\_2[j])<=64)) then

Begin

Writeln('Вы ввели неверный символ! Попробуйте ещё

раз');

L := False;

End

Else

L := True;

// A new rank of the sum

If ArrayOfSum[I] = N then

Begin

ArrayOfSum[I] := 0;

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

// Increasing border and length of the sum

// as we have a new rank of the sum

if I = Border then

begin

Border := Border - 1;

end;

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 := ArrayOfSum[I] + (ord(Num\_2[j]) - 48)

Else

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

// A new rank of the sum

if RankOfNum > (N - 1) then

Begin

// Increasing border as we have a

// new rank of the sum

if I = Border then

Border := Border - 1;

// Changing the rank

ArrayOfSum[I] := RankOfNum - N;

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

End

Else

ArrayOfSum[I] := RankOfNum;

j := j - 1;

end;

I := I - 1;

End;

Until L;

if (Z1 = -1) and (Z2 = -1) then

Begin

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

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

Write('Сумма чисел ', Num\_1, ' и ', Num\_2, ' равна -')

End

Else

Write('Сумма чисел ', Num\_1, ' и ', Num\_2, ' равна ');

// Finding a border of length of the sum

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

j := (LengthA - length(Num\_2)) - 1

Else

j := (LengthA - length(Num\_1)) - 1;

if ArrayOfSum[j] = 0 then

While ArrayOfSum[j] = 0 do

j := j + 1;

// Outputting the sum in the required numbering system

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

if ArrayOfSum[I] <= 9 then

Write(char((ArrayOfSum[I] + 48)))

Else

Write(char((ArrayOfSum[I] + 55)));

readln;

end.

Приложение Б

(обязательное)

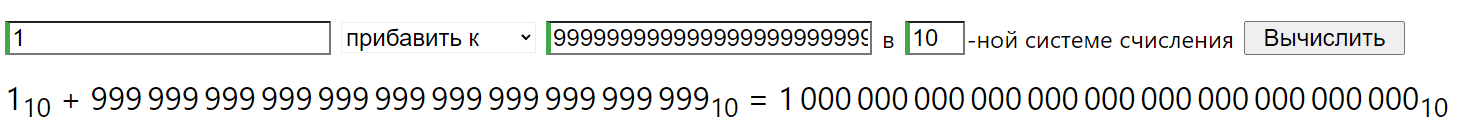
Тестовые наборы

Тест 1

Исходные данные:

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

Ожидаемый результат:



Полученный результат:

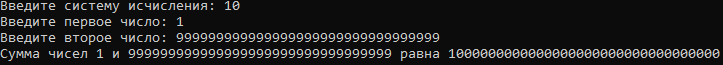


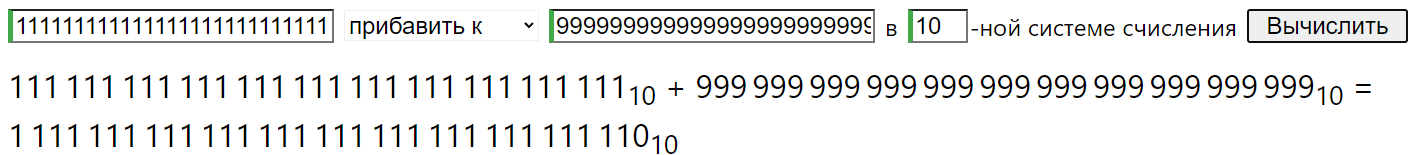
Рисунок 2 – Тест 1

Тест 2

Исходные данные:

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

Ожидаемый результат:



Полученный результат:

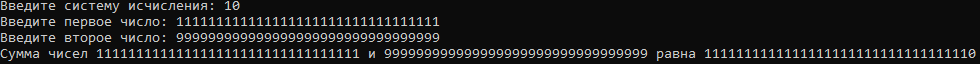


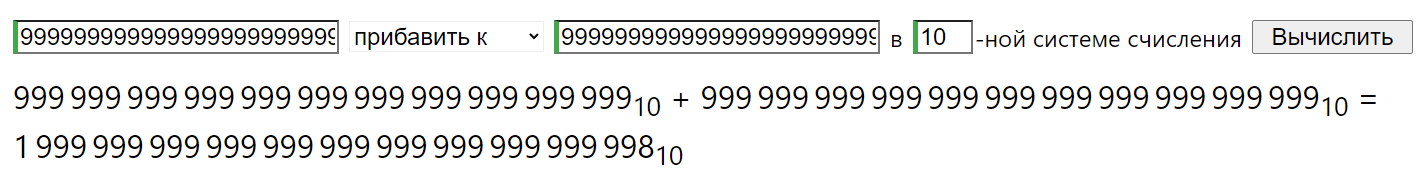
Рисунок 3 – Тест 2

Тест 3

Исходные данные:

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

Ожидаемый результат:



Полученный результат:

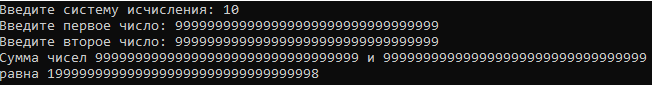


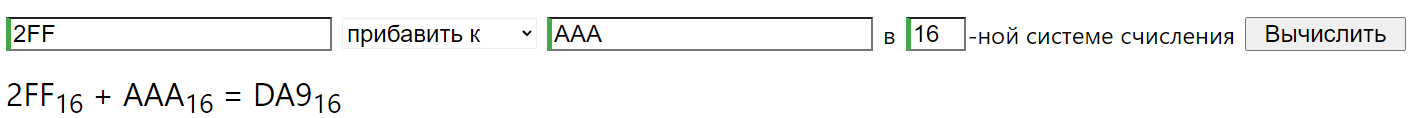
Рисунок 4 – Тест 3

Тест 4

Исходные данные:

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

Ожидаемый результат:



Полученный результат:

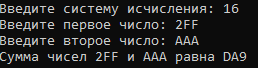


Рисунок 5 – Тест 4

Тест 5

Исходные данные:

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

Ожидаемый результат:

1010 0101

Полученный результат:

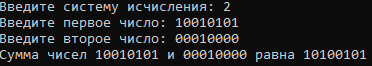


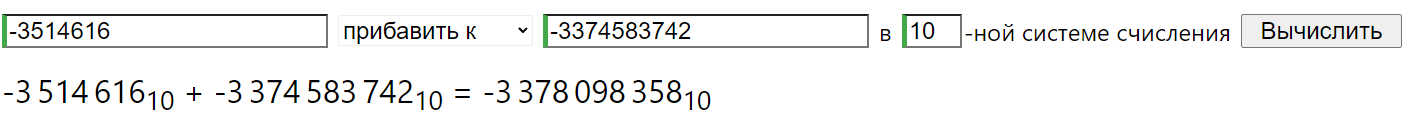
Рисунок 6 – Тест 5

Тест 6

Исходные данные:

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

Ожидаемый результат:



Полученный результат:

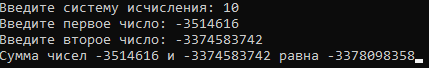


Рисунок 7 – Тест 6

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) |