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 разминочной задаче № 4

Тема работы: Длинная арифметика(сложение)

Выполнил

студент: гр. 151003 Барановский Р.А.

Проверил: Фадеева Е.П.

Минск 2021

содержание

[1 Постановка задачи 3](#_Toc84882004)

[2 Структура данных 4](#_Toc84882005)

[Приложение А 5](#_Toc84882006)

[Приложение Б 9](#_Toc84882007)

# Постановка задачи

Имеются два числа длиной до 50 символов, необходимо сложить их и вывести на экран результат подсчетов

# Структура данных

Таблица 1 - Данные

|  |  |  |
| --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение |
| Num1Str | Array of integer | Массив, содержащий в себе первое число в строковом виде |
| Num2Str | Array of integer | Массив, содержащий в себе первое число в строковом виде |
| I | Array of integer | Счетчик массивов |
| BigLength | Integer | Длина большего числа |
| Temp | Integer | Переменная, использующаяся для переноса из одного разряда в другой |
| NewSize | Integer | Размер результата вычислений |
| K | Integer | Счетчик, использующийся в цикле, который выполняет функцию отбрасывания 0 в начале получившегося числа |
| Num1 | Integer | Число 1 |
| Num2 | Integer | Число 2 |
| SumNum | Integer | Сумма чисел |

Приложение А

(обязательное)

Исходный код программы

Program Exp1New;

//Program pluses 2 numbers

//Console app

{$APPTYPE CONSOLE}

//Modules declaration

Uses

System.SysUtils;

//Variables declaration

Var

Num1Str, Num2Str: String;

Checker: boolean;

I, BigLength, Temp, NewSize, K: integer;

Num1: array of integer;

Num2: array of integer;

SumNum: array of integer;

{ Num1Str - Num1 in String

Num2Str - Num2 in String

Checker - mistakes checker

I - counter for arrays

BigLength - length of biggest inputted number

Temp - auxilliary variable used to move 1 to the next

rank if necessary

NewSize - size of final number after removing 0 in the

front

K - counter to output result

Num1 - 1-st number

Num2 - 2-nd number

SumNum - calculations result }

//Start the program

Begin

//Input 1-st number

repeat

Checker := true;

write('Пожалуйста, введите первое число: ');

readln(Num1Str);

if length(Num1Str) = 0 then

Checker := false;

i := 1;

for i := 1 to length(Num1Str) do

if (ord(Num1Str[i]) < 48) or (ord(Num1Str[i]) > 58)

then

Checker := false;

if not(Checker) then

writeln('Ошибка ввода!');

until Checker;

writeln('+':35);

//Input 2-nd number

repeat

Checker := true;

write('Пожалуйста, введите второе число: ');

readln(Num2Str);

if length(Num2Str) = 0 then

Checker := false;

i := 1;

for i := 1 to length(Num2Str) do

if (ord(Num2Str[i]) < 48) or (ord(Num2Str[i]) > 58)

then

Checker := false;

if not(Checker) then

writeln('Ошибка ввода!');

until Checker;

//Find out which variable is bigger

if length(Num1Str) > length(Num2Str) then

BigLength := length(Num1Str)

else

BigLength := length(Num2Str);

//Set lengths of arrays

setlength(Num1, BigLength);

setlength(Num2, BigLength);

setlength(SumNum, BigLength + 1);

//In order to place number 1 at the end of array num1

for i := 0 to length(Num1Str) - 1 do

Num1[BigLength - length(Num1Str) + i] :=

StrToInt(Num1Str[i + 1]);

//In order to place number 2 at the end of array num2

for i := 0 to length(Num2Str) - 1 do

Num2[BigLength - length(Num2Str) + i] :=

StrToInt(Num2Str[i + 1]);

//Calculate the sum

for i := BigLength - 1 downto 0 do

begin

//Pluses two columns

SumNum[i+1] := (SumNum[i+1] + (Num1[i] + Num2[i]) mod

10) mod 10 ;

//To stay within arrays border

if i < BigLength - 1 then

//Find value of Temp

Temp := Temp + Num1[i+1] + Num2[i + 1];

//If we need to move 1 to the next rank then Temp = 1

if Temp >= 10 then

Temp := 1

//If we dont need to move 1 to the next rank then

//Temp = 0

else

Temp := 0;

//If we need to move 1 to the next rank

if Num1[i] + Num2[i] + Temp >= 10 then

SumNum[i] := 1;

end;

writeln('=':35);

//Remove 0 before SumNumbers

i:=0;

NewSize := BigLength + 1;

while SumNum[i] = 0 do

begin

for k := 0 to BigLength - 1 do

SumNum[k] := Sumnum[k+1];

NewSize := NewSize - 1;

end;

//To make space in front of SumNum

write('':34);

//Output SumNumbers

for i := 0 to NewSize - 1 do

write(SumNum[i]);

readln;

End.

Приложение Б

(обязательное)

Тестовые наборы(Расчеты)

Тест 1

Тестовая ситуация:

Num1 := 32741009733780499453459097479817698344322403210836

Num2 := 86110622288602330612111242571019368673702820068831

Исходные данные:

Num1 := 32741009733780499453459097479817698344322403210836

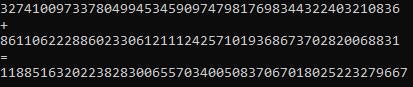
Num1 := 86110622288602330612111242571019368673702820068831

Ожидаемый результат:

SumNum :=

118851632022382830065570340050837067018025223279667

Полученный результат:



Тест 2

Тестовая ситуация:

Num1 := 54321993452292802603982469527853553749992572983731

Num2 := 27934143096447509587759065208465254053862867883358

Исходные данные:

Num1 := 54321993452292802603982469527853553749992572983731

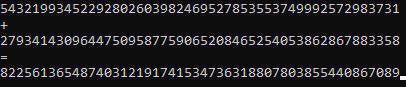
Num2 := 27934143096447509587759065208465254053862867883358

Ожидаемый результат:

SumNum :=

82256136548740312191741534736318807803855440867089

Полученный результат:



Тест 3

Тестовая ситуация:

Num1 := 5104559485376765765491012190923766237373315392827

Num2 := 19234529493696598545943266736217414553661748885261

Исходные данные:

Num1 := 5104559485376765765491012190923766237373315392827

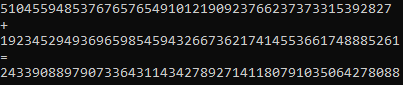
Num2 := 19234529493696598545943266736217414553661748885261

Ожидаемый результат:

SumNum :=

24339088979073364311434278927141180791035064278088

Полученный результат:



Тестовые наборы(Проверка ввода)

Тест 4

Тестовая ситуация: Проверка ввода букв в поле для чисел

Исходные данные:

Num1Str = ‘я люблю мороженое.’

Ожидаемый результат:

Ошибка ввода

Полученный результат:

