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 лабораторной работе №2

Тема работы: Работа с числами

Выполнил

студент: гр. 151003 Матошко И.В

Проверил: Фадеева Е.П.

Минск 2021

содержание

[1 Постановка задачи 3](#_Toc87539357)

[2 Текстовый алгоритм решения задачи 4](#_Toc87539358)

[3 Структура данных 6](#_Toc87539359)

[4 Схема алгоритма решения задачи по ГОСТ 19.701-90 7](#_Toc87539360)

[5 Результаты расчетов 11](#_Toc87539361)

[ПРИЛОЖЕНИЕ А 12](#_Toc87539362)

[ПРИЛОЖЕНИЕ Б 15](#_Toc87539363)

# Постановка задачи

Даны взаимно простые натуральные числа P и Q (P<Q). Разработать алгоритм и программу поиска периодической и непериодической части десятичной дроби, равной P/Q.

Вывести на печать: последовательность периодической и непериодической части дроби.

Пробелы !!!!!!!!!!

# Текстовый алгоритм решения задачи

Таблица 1 ­­– Алгоритм решения

|  |  |
| --- | --- |
| Номер  шага | Назначение шага |
|  |  |
|  | Ввод P1 |
|  | Ввод Q1 |
|  | Вывод (P1 div Q1) |
|  | P1 := P1 mod Q1 |
|  | Проверка выполнения условия (P1>0). Если условие истинно, идти к шагу 7, иначе – к шагу 49 |
|  | p := P1 |
|  | q := Q1 |
|  | Начало цикла А1. Проверка выполнения условия (p< >q). Если условие истинно, идти к шагу 10, иначе – к шагу 14 |
|  | Проверка выполнения условия (q>p). Если условие истинно, идти к шагу 11, иначе – к шагу 12 |
|  | q := q - p |
|  | p := p - q |
|  | Конец цикла А1. Идти к шагу 9 |
|  | P1 := P1 div p |
|  | Q1 := Q1 div p |
|  | p := 0 |
|  | q := 0 |
|  | z := Q1 |
|  | Начало цикла А2. Проверка выполнения условия (z mod 2 = 0). Если условие истинно, идти к шагу 20, иначе – к шагу23 |
|  | inc(p) |
|  | z := z div 2 |
|  | Конец цикла А2. Идти к шагу 19 |
|  | Начало цикла А3. Проверка выполнения условия (z mod 5 = 0). Если условие истинно, идти к шагу 24, иначе – к шагу 27 |
|  | inc(q) |
|  | z := z div 5 |
|  | Конец цикла А3. Идти к шагу 23 |
|  | Проверка выполнения условия (p<q). Если условие истинно, идти к шагу 28, иначе – к шагу 29 |
|  | p := q |
|  | k:=1 |
|  | Начало цикла А4. Проверка выполнения условия (k<=p). Если условие истинно, идти к шагу 31, иначе – к шагу 36 |
|  | P1 := P1 \* 10 |
|  | Вывод (P1 div Q1) |
|  | P1 := P1 mod Q1 |
|  | k:=k+1 |
|  | Конец цикла А4. Идти к шагу 30 |
|  | Проверка выполнения условия (z>1). Если условие истинно, идти к шагу 37, иначе – к шагу 49 |
|  | q := 9 |
|  | p := 1 |
|  | Начало цикла А5. Проверка выполнения условия (q mod z > 0). Если условие истинно, идти к шагу 40, иначе – к шагу 43 |
|  | inc(p) |
|  | q := q mod z \* 10 + 9 |
|  | Конец цикла А5. Идти к шагу 39 |
|  | k:=1 |
|  | Начало цикла А6. Проверка выполнения условия (k<=p). Если условие истинно, идти к шагу 45, иначе – к шагу 49 |
|  | P1 := P1 \* 10 |
|  | Вывод (P1 div Q1) |
|  | P1 := P1 mod Q1 |
|  | Конец цикла А6. Идти к шагу 44 |
|  | Останов. |

# Структура данных

Таблица 2 – Данные

|  |  |  |
| --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение |
| P1 | Integer | Числитель дроби |
| Q1 | Integer | Знаменатель дроби |
| p | Integer | НОД и длина предпериода |
| z | Integer | Остаток от деления на 2 и 5(доп. делитель) |
| k | Integer | Счетчик цикла |
| q | Integer | Поиск длины периода |

# Схема алгоритма решения задачи по ГОСТ 19.701-90



Рисунок 1 – Схема алгоритма решения задачи по ГОСТ 19.701-90



Рисунок 2 – Схема алгоритма решения задачи по ГОСТ 19.701-90



Рисунок 3 – Схема алгоритма решения задачи по ГОСТ 19.701-90



Рисунок 4 – Схема алгоритма решения задачи по ГОСТ 19.701-90

# Результаты расчетов

Вследствие выполнения программы на экран выводятся следующие результаты:

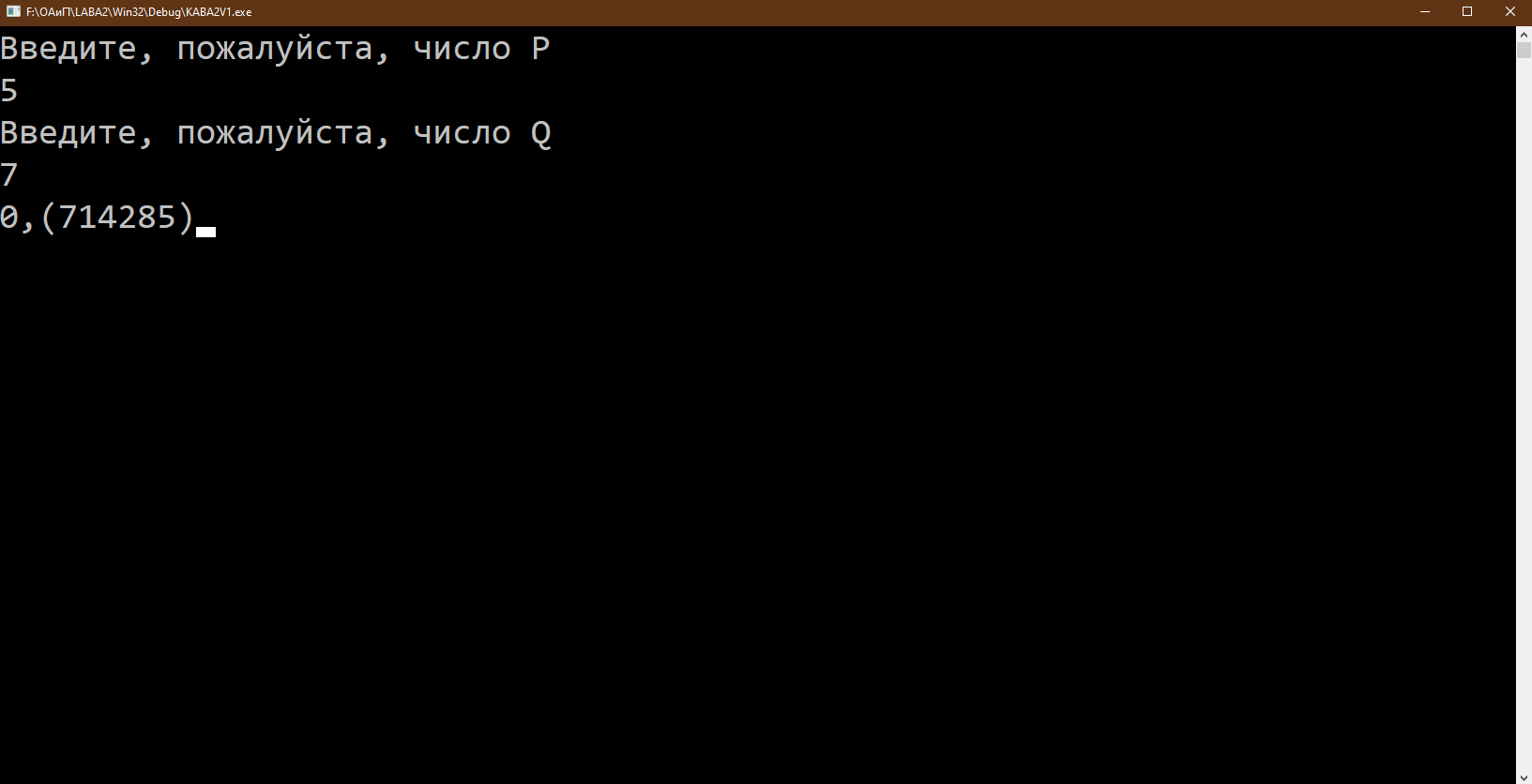


Рисунок 6 – Результаты расчётов

ПРИЛОЖЕНИЕ А

(обязательное)

Исходный код программы

Program KABA2V1;

{ Программа выводит периодическую и непериодическую часть дроби P/Q}

{$APPTYPE CONSOLE} // Консольное приложение

uses

System.SysUtils;

var

P1, Q1, p, q, k, z, error1, error2: integer;

proverka1:string;

{ P1 - числитель дроби;

Q1 - знаменатель дроби;

p - НОД и длина предпериода;

z - остаток от деления на 2 и 5(доп. делитель);

k - счетчик для цикла for;

q - поиск длины периода;

error1,error2 - число ошибок в проверке

вводимых данных;

proverka1 - переменная для проверки вводимых данных. }

Begin

// Вводим значение числителя дроби и проверяем его.

repeat

Writeln('Введите, пожалуйста, число P');

readln(proverka1);

val(proverka1,P1,error1);

if (error1>0) or(P1<=0) then

writeln('Ошибка. Введены некорректные данные');

until (error1=0) and (P1>0);

// Вводим значение знаменателя дроби и проверяем его.

repeat

Writeln('Введите, пожалуйста, число Q');

readln(proverka1);

val(proverka1,Q1,error2);

if (error2>0) or(Q1<=0) then

writeln('Ошибка. Введены некорректные данные');

until (error2=0) and (Q1>0);

// Выводим на экран целую часть дроби.

write(P1 div Q1);

// Ищем остаток и избавляемся от целой части.

P1 := P1 mod Q1;

// Если числитель > 0, то есть и дробная часть.

if P1 > 0 then

begin

// Выводим запятую в записи дробного числа.

write(',');

// Находим НОД ( "медленный" алгоритм Евклида) для

// сокращения дроби, если оно возможно.

p := P1;

q := Q1;

while p <> q do

begin

if q > p then

q := q - p

else

p := p - q;

end;

// Сокращаем дробь P1/Q1

P1 := P1 div p;

Q1 := Q1 div p;

// Находим длину предпериода дроби. Используем для этого

// максимальное количество делителей 2 и 5.

p := 0; // Длина будущего предпериода.

q := 0;

z := Q1; // Будущий остаток знаменателя

// 2 и на 5.

while z mod 2 = 0 do

begin

inc(p);

z := z div 2;

end;

while z mod 5 = 0 do

begin

inc(q);

z := z div 5;

end;

if p < q then

p := q;

// Выводим на экран предпериод дроби

for k := 1 to p do

begin

P1 := P1 \* 10;

write(P1 div Q1);

P1 := P1 mod Q1;

end;

// Если есть период, находим его длину и выводим на экран.

if z > 1 // Если делители знаменателя не только 2 и 5, то есть период.

then

begin

write('(');

// Поиск длины периода с поощью девяток.

q := 9;

// p - количество разрядов в периоде.

p := 1;

while q mod z > 0 do

begin

inc(p);

// Делим 999.. по модулю z, во избежание

//переполнения.

q := q mod z \* 10 + 9;

end;

// Выводим на экран цифры периода.

for k := 1 to p do

begin

P1 := P1 \* 10;

write(P1 div Q1);

P1 := P1 mod Q1;

end;

write(')');

end;

end;

readln;

readln;

End.

ПРИЛОЖЕНИЕ Б

(обязательное)

Тестовые наборы

Тестовый набор 1 «Проверка ввода»

Тест 1

Тестовая ситуация: проверка ввода букв в поле для чисел

Исходные данные: P1 = «Hello World»

Ожидаемый результат:

Ошибка. Введены некорректные данные

Полученный результат:



Тест 2

Тестовая ситуация: проверка ввода нецелочисленных значений

Исходные данные: P1 = «123,4»

Ожидаемый результат:

Ошибка. Введены некорректные данные

Полученный результат:



Тест 3

Тестовая ситуация: проверка ввода отрицательных значений

Исходные данные: P1 = «-123»

Ожидаемый результат:

Ошибка. Введены некорректные данные

Полученный результат:



Тестовый набор 2

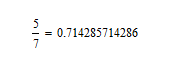
«Проверка решений»

Тест 1

Тестовая ситуация: проверка решений

Исходные данные: P1 = «5», Q1 = «7»

Ожидаемый результат (расчёт Mathcad 14):



Полученный результат:

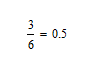


Тест 2

Тестовая ситуация: проверка решений, если дробь сокращается

Исходные данные: P1 = «3», Q1 = «6»

Ожидаемый результат (расчёт Mathcad):



Полученный результат:

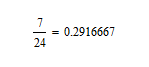


Тест 3

Тестовая ситуация: проверка решений, если есть непериодическая часть

Исходные данные: P1 = «7», Q1 = «24»

Ожидаемый результат (расчёт Mathcad):



Полученный результат:

