**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 1

**«Язык программирования ЛИСП»**

Выполнил: студент гр. ИТ-31

Болтуть М.С.

Принял: преподаватель

Левцова Т.С.

Гомель 2018

**Цель работы:** Знакомство с языком Лисп. Изучение представления данных в языке Лисп в виде атомов, списков, консов и символьных выражений.

**Ход работы:**

1. **Пример простейшей программы на языке ЛИСП**

Разработать программу вычисления факториала на процедурном (Pascal, C, можно C#), объектно-ориентированном (C++ или Java) и функциональном языке программирования (Lisp). Найти максимальное значение аргумента, от которого можно вычислить факториал без потери точности (в целых числах).

**Листинг кода на языке С:**

#include <stdio.h>

#include <iostream>

int main()

{

setlocale(LC\_ALL, "Russian");

long num = 19;

int fact = 1;

for (int i = 0; i < num; i++)

{

fact \*= i + 1;

}

printf("Факториал %d: %ld", num, fact);

getchar();

return 0;

}

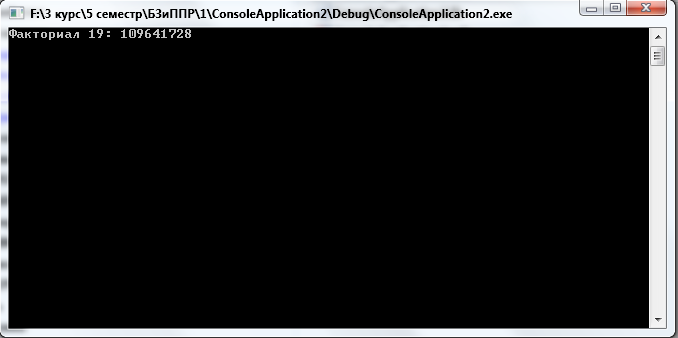


Рисунок 1 – Результат вычисления факториала на языке С

Предел вычисления – 19!

**Листинг кода на языке С#:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication1

{

class Program

{

static void Main(string[] args)

{

int num = 20;

long fact = 1;

for (int i = 0; i < num; i++)

{

fact \*= i + 1;

}

Console.WriteLine("Факториал {0}: {1}", num, fact);

Console.ReadKey();

}

}

}



Рисунок 2 – Результат вычисления факториала на языке С#

Предел вычисления – 20!

**Листинг кода на языке Лисп:**

(defun fact1 (n)

(cond ((zerop n) 1)

(t (\* n (fact1 (- n 1))))))

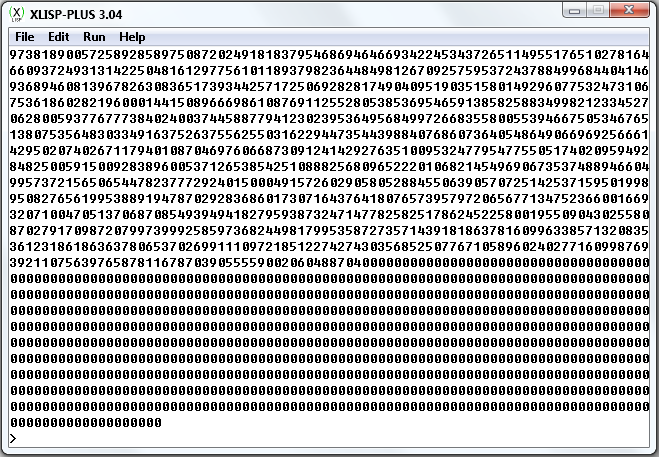


Рисунок 3 – Результат вычисления факториала на языке Лисп

Предел вычисления – 3117!

1. **Базовые функции языка ЛИСП**

Удалить первый и последний элемент списка.

**Листинг кода на языке ЛИСП:**

(defun del\_first\_last (list);удаление первого и последнего элемента из списка

(reverse (cdr (reverse (cdr list))))

)

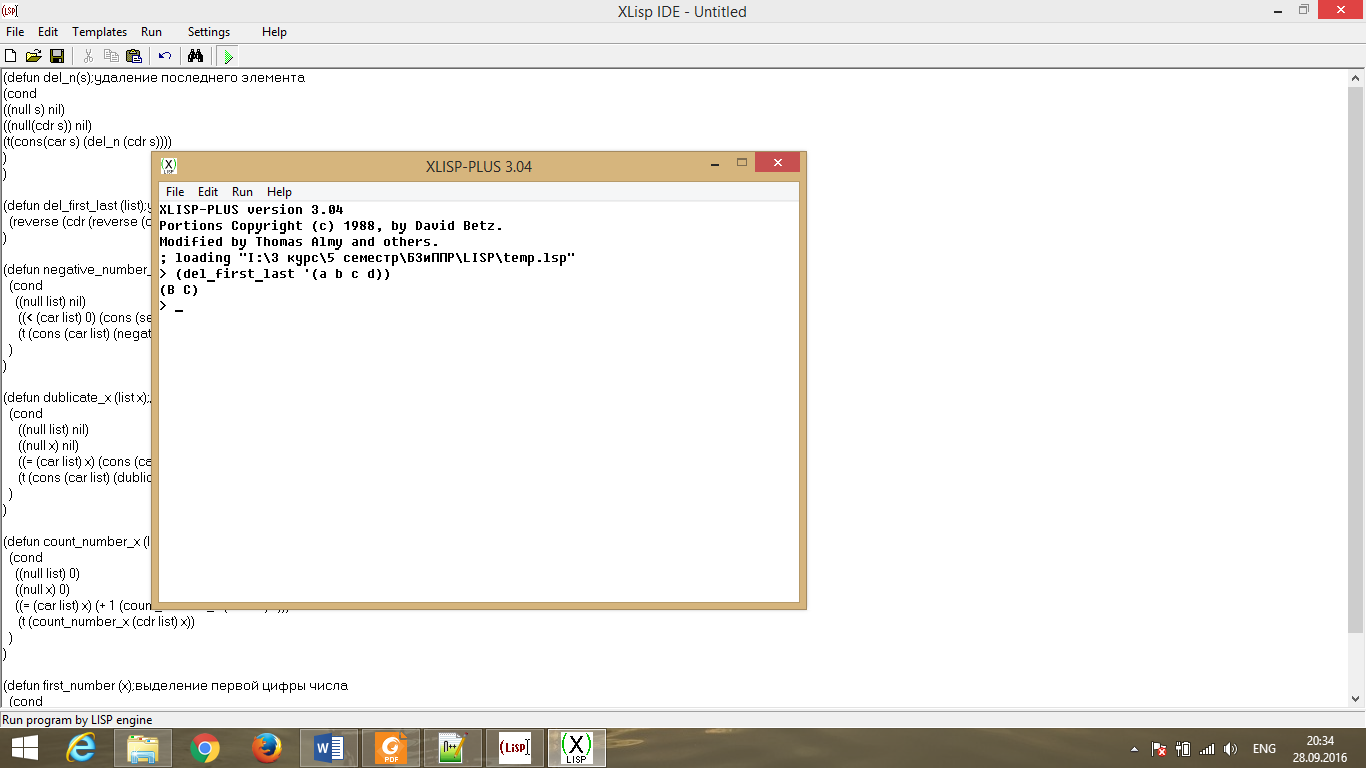


Рисунок 4 – Результат выполнения кода

В простом списке чисел заменить все отрицательные числа нулями.

**Листинг кода на языке ЛИСП:**

(defun negative\_number\_replace(list);замена отрицательных чисел нулями

(cond

((null list) nil)

((< (car list) 0) (cons (setf (car list) 0) (negative\_number\_replace(cdr list))))

(t (cons (car list) (negative\_number\_replace (cdr list))))

)

)

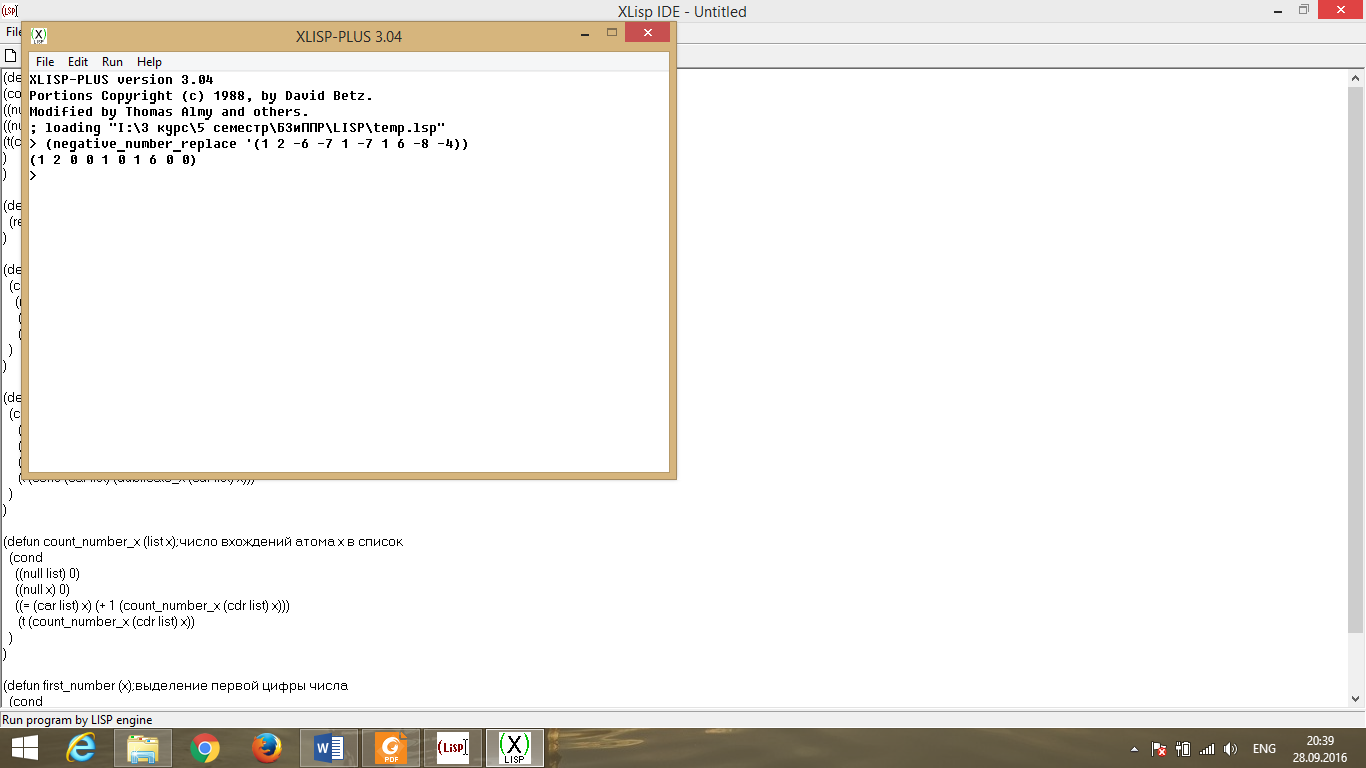


Рисунок 4 – Результат выполнения кода

Из простого списка чисел удалить все нулевые элементы.

**Листинг кода на языке ЛИСП:**

(defun del\_zero\_numbers (s);удаление нулевых элементов

(cond

((null s) nil)

((= (car s) 0) (del\_zero\_numbers (cdr s)))

(t(cons (car s) (del\_zero\_numbers (cdr s))))

)

)

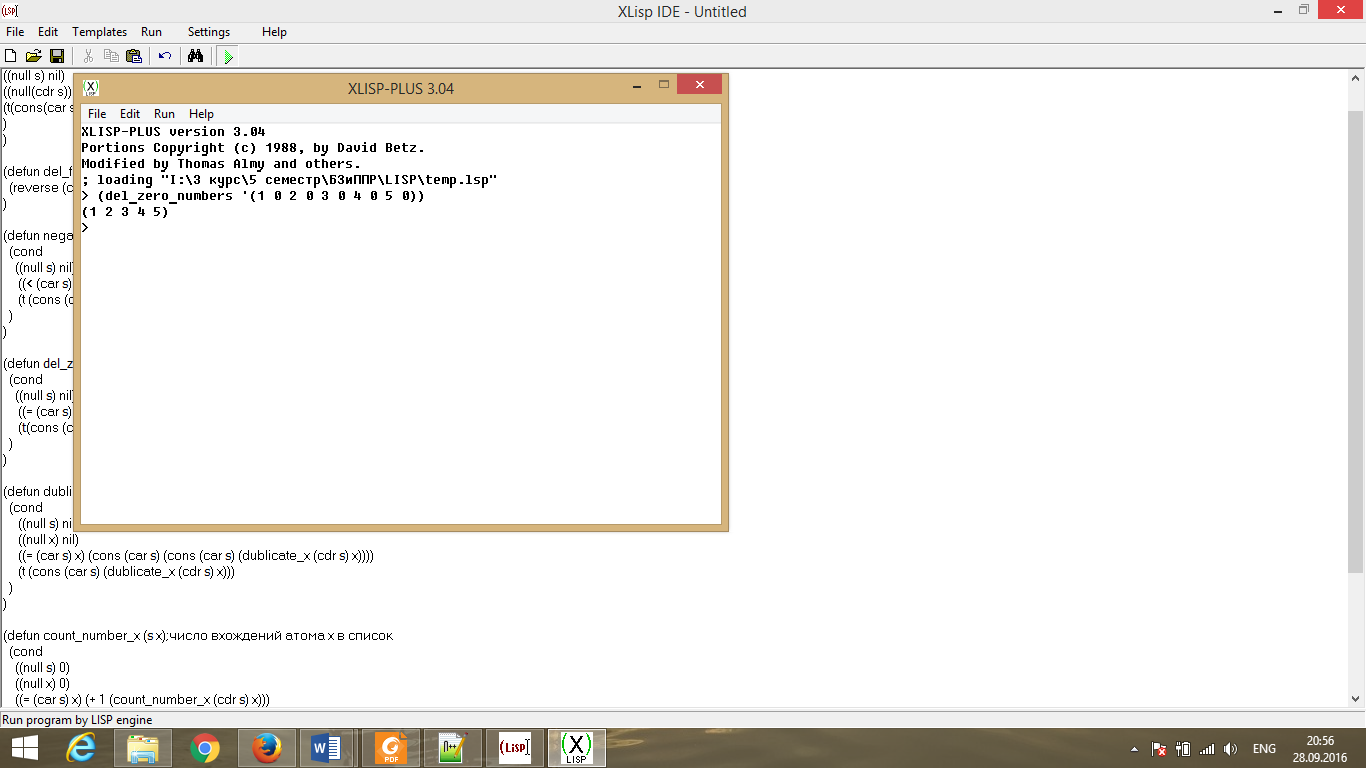


Рисунок 5 – Результат выполнения кода

Продублировать все вхождения атома X в данный список

**Листинг кода на языке ЛИСП:**

(defun dublicate\_x (s x);дублирование хождений атома х в список

(cond

((null s) nil)

((null x) nil)

((= (car s) x) (cons (car s) (cons (car s) (dublicate\_x (cdr s) x))))

(t (cons (car s) (dublicate\_x (cdr s) x)))

)

)

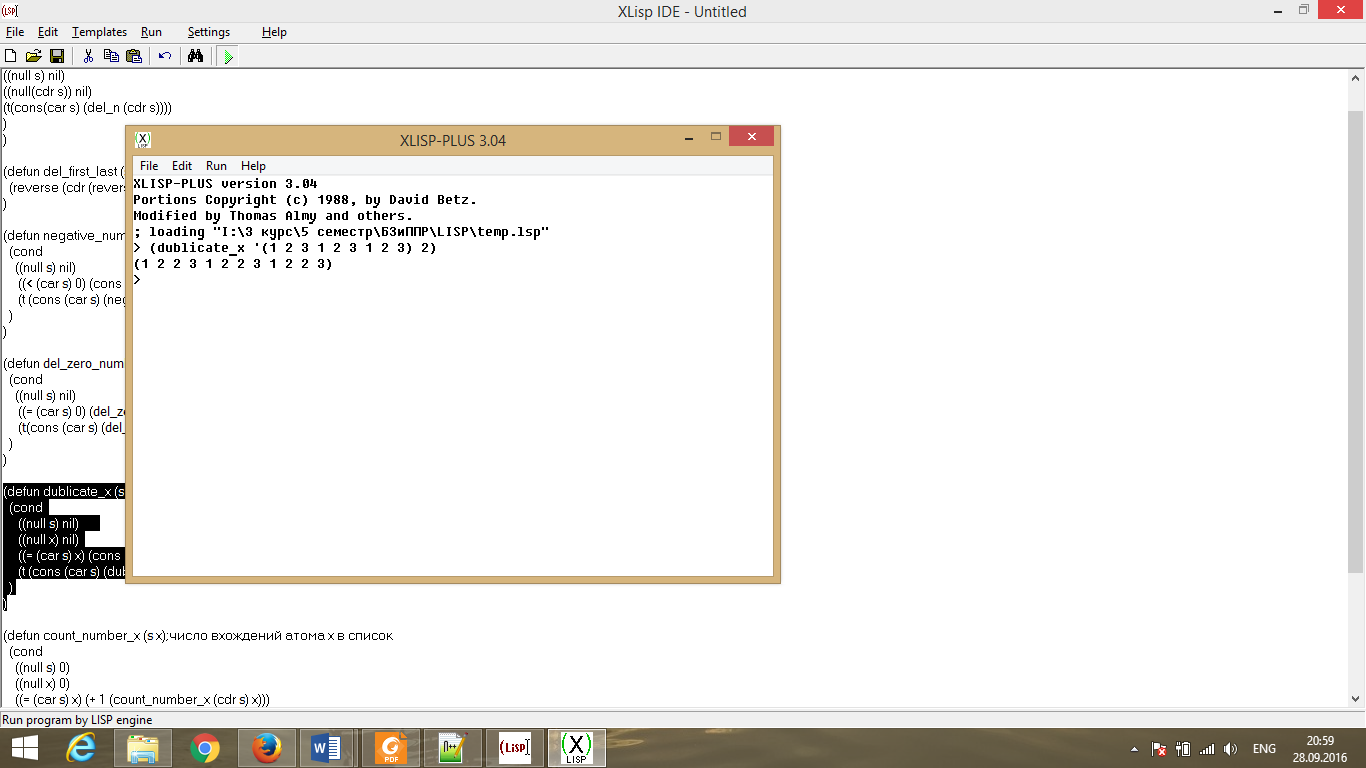


Рисунок 6 – Результат выполнения кода

Подсчитать число вхождений атома X в список

**Листинг кода на языке ЛИСП:**

(defun count\_number\_x (s x);число вхождений атома х в список

(cond

((null s) 0)

((null x) 0)

((= (car s) x) (+ 1 (count\_number\_x (cdr s) x)))

(t(count\_number\_x (cdr s) x))

)

)

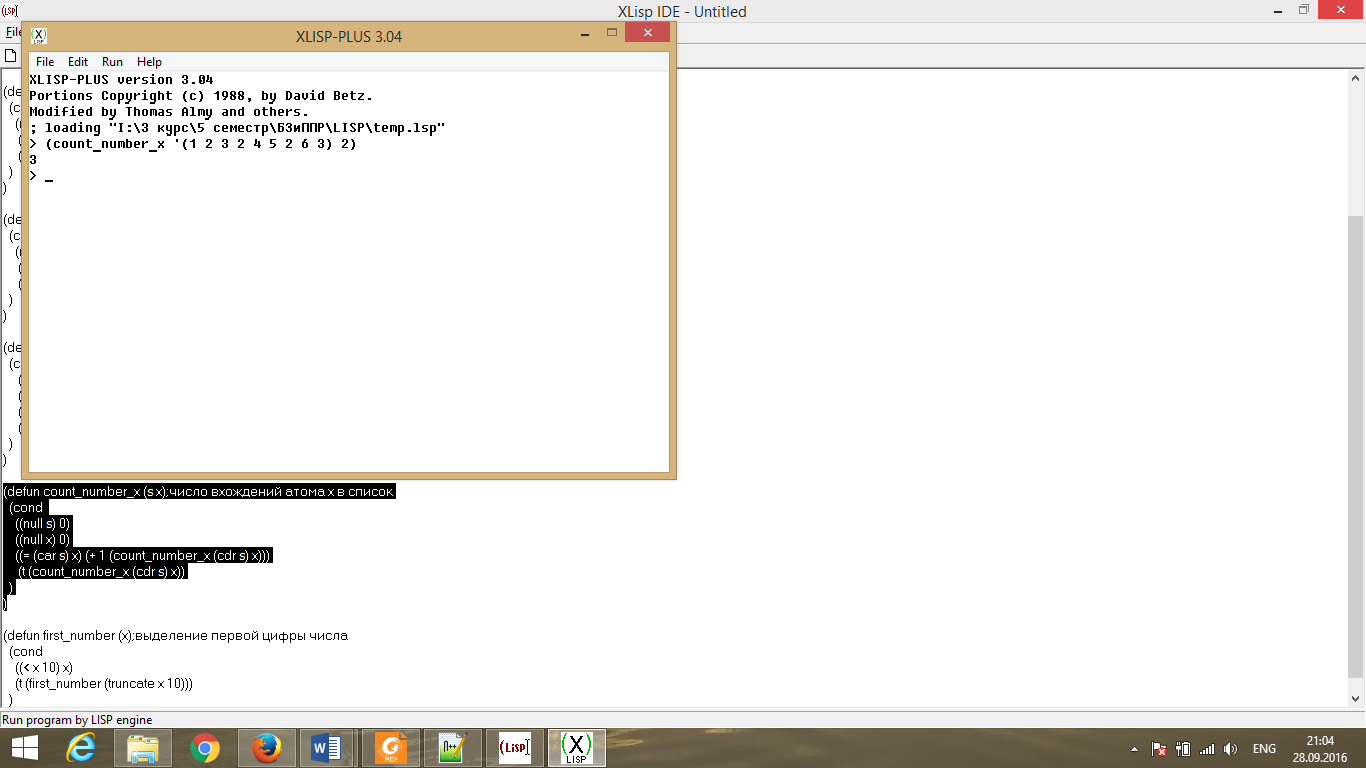


Рисунок 7 – Результат выполнения кода

1. **Числовые функции языка ЛИСП.**

Выделить первую цифру натурального N.

**Листинг кода на языке ЛИСП:**

(defun first\_number (x);выделение первой цифры числа

(cond

((< x 10) x)

(t (first\_number (truncate x 10)))

)

)

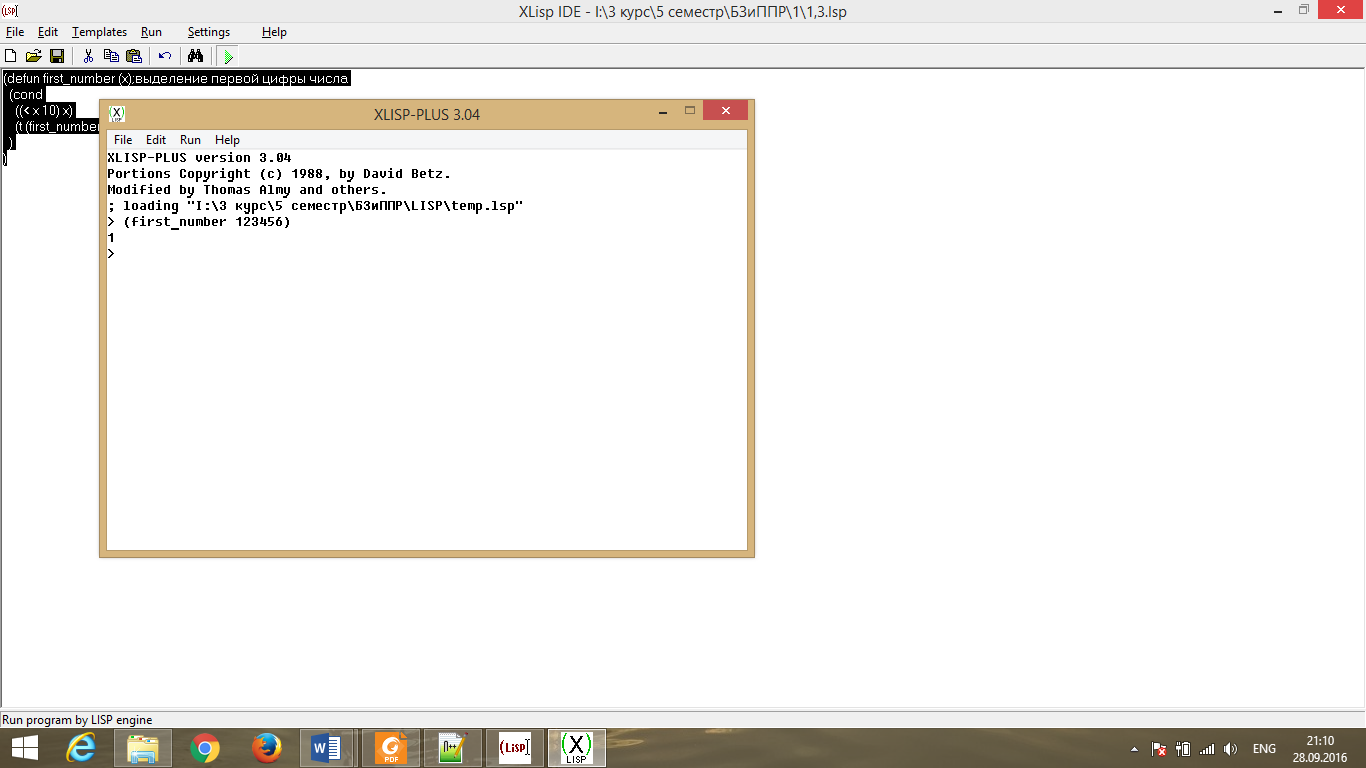


Рисунок 8 – Результат выполнения кода

Подсчитать число и сумму цифр целого N.

**Листинг кода на языке ЛИСП:**

(defun number\_and\_summ(n)

(if (zerop n) '(0 1)

(list (+ (rem n 10) (car (number\_and\_summ (floor n 10))))

(1+ (cadr (number\_and\_summ (floor n 10)))))))

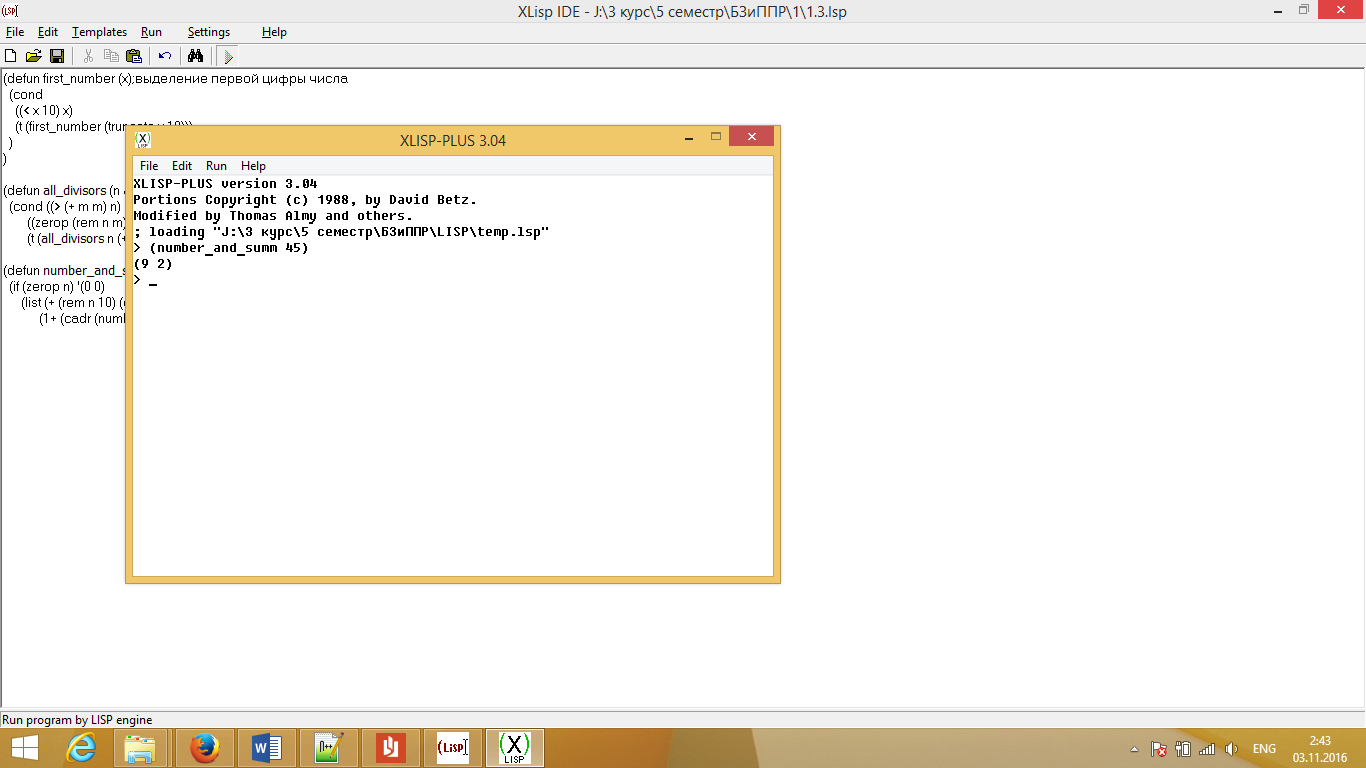


Рисунок 10 – Результат выполнения программы

Найти все общие делители натуральных M и N.

**Листинг кода на языке ЛИСП:**

(defun common\_divisors (m n)

(cond

((= n 0) nil)

((= (rem m n) 1) (cons n (common\_divisors m (1- n))))

(t (common\_divisors m (1- n)))

)

)

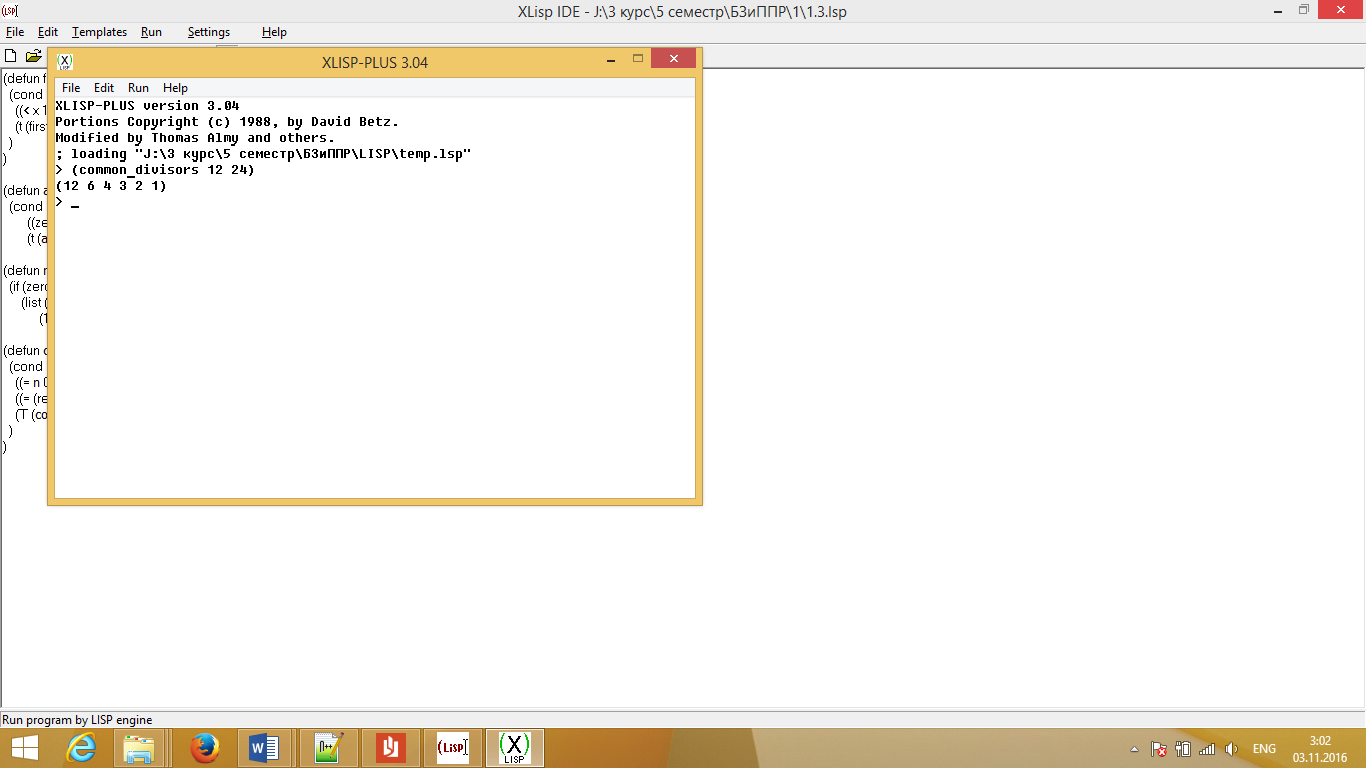


Рисунок 11 – Результат выполнения программы

Найти наибольший общий делитель чисел из заданного списка.

**Листинг кода на языке ЛИСП:**

(defun nod (list)

(cond

((= (length list) 1) (car list))

(T (gcd (car list) (nod (cdr list))))

)

)

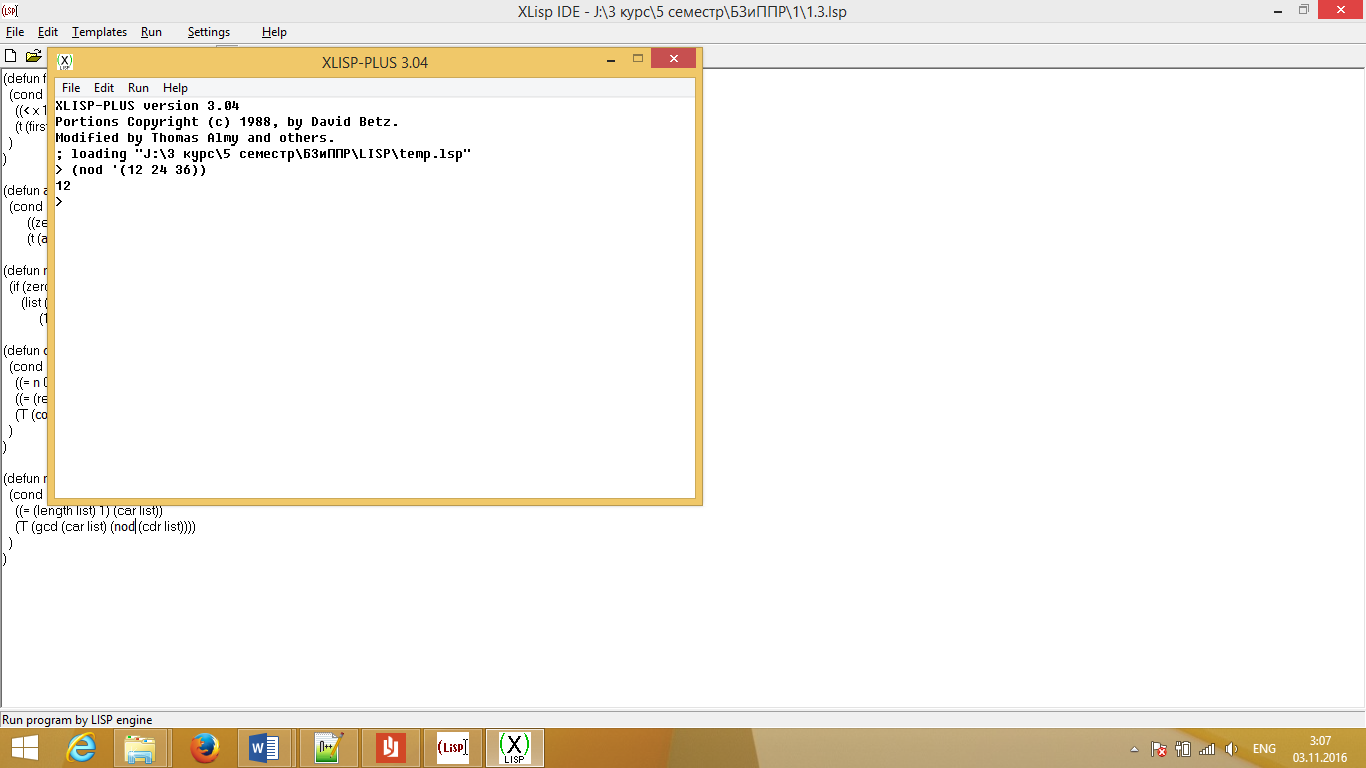


Рисунок 12 – Результат выполнения программы

**Вывод:** В ходе работы ознакомился с функциональным языком программирования ЛИСП. Изучил представление данных в языке ЛИСП в виде атомов, консов, списков. Произвел операции над данными.