

Міністерство освіти і науки, молоді та спорту України

Національний аерокосмічний університет
ім. М.Є. Жуковського

Кафедра 503

Лабораторна робота № 1

з дисципліни

«Архітектура комп'ютерів»

Тема: «Програмування лінійних програм у МП Intel x86»

Варіант № 37

Виконав: ст. гр. 525д

Петренко Р.Е.

Перевірив: ст. викладач

Дужий В.І.

Харків 2012

1. **Задание.** Дано выражение:

$$(X^2 - 9) / (X - 3) = X + 3$$

где исходное значение необходимо разделить на заданную константу, т.е. $X = X/37$.

Написать линейный алгоритм вычисления значения левой и правой частей арифметического выражения и реализовать его в виде программы на С и на ассемблере. В процессе вычислений программа должна контролировать возможное переполнение.

2. Исходные данные.

X – переменная, длинное целое знаковое число.

9, 3 – константы, длинные целые знаковые.

3. Требуемый результат.

Left, Right – переменные, длинные целые знаковые числа для фрагмента на С.

Left_a, Right_a – переменные, длинные целые знаковые числа для фрагмента на ассемблере.

4. Алгоритм решения задачи на псевдокоде.

```
Ввести исходные переменные;
Вычислить левую часть тождества на С;
Вычислить правую часть тождества на С;
// Сбросить признак ошибки err;
err = 0;
// Разделить исходную переменную X на константу 35;
X1 = x/35;
// Вычислить тождество на ассемблере;
// Вычислить левую часть тождества;
tmp = X1*X1; // Вычислить произведение
// Определить, имеется ли переполнение;
Если переполнение, err = 1;
denom = X1-3; // Вычислить знаменатель
// Определить, имеется ли переполнение;
Если переполнение, err = 1;
// Продолжить вычисление левой части тождества;
Left = (tmp-9)/denom;
// Вычислить правую часть тождества на ассемблере;
Right = X1+3;
// Вывести значение левой и правой частей тождества на С;
// Если нет переполнения, вывести значение левой и правой
// частей тождества на ассемблере;
Если (err=0) Вывести сообщение об ошибке;
иначе Вывести значение левой и правой частей тождества.
```

Детальный алгоритм решения задачи представлен ниже на рисунке 1.

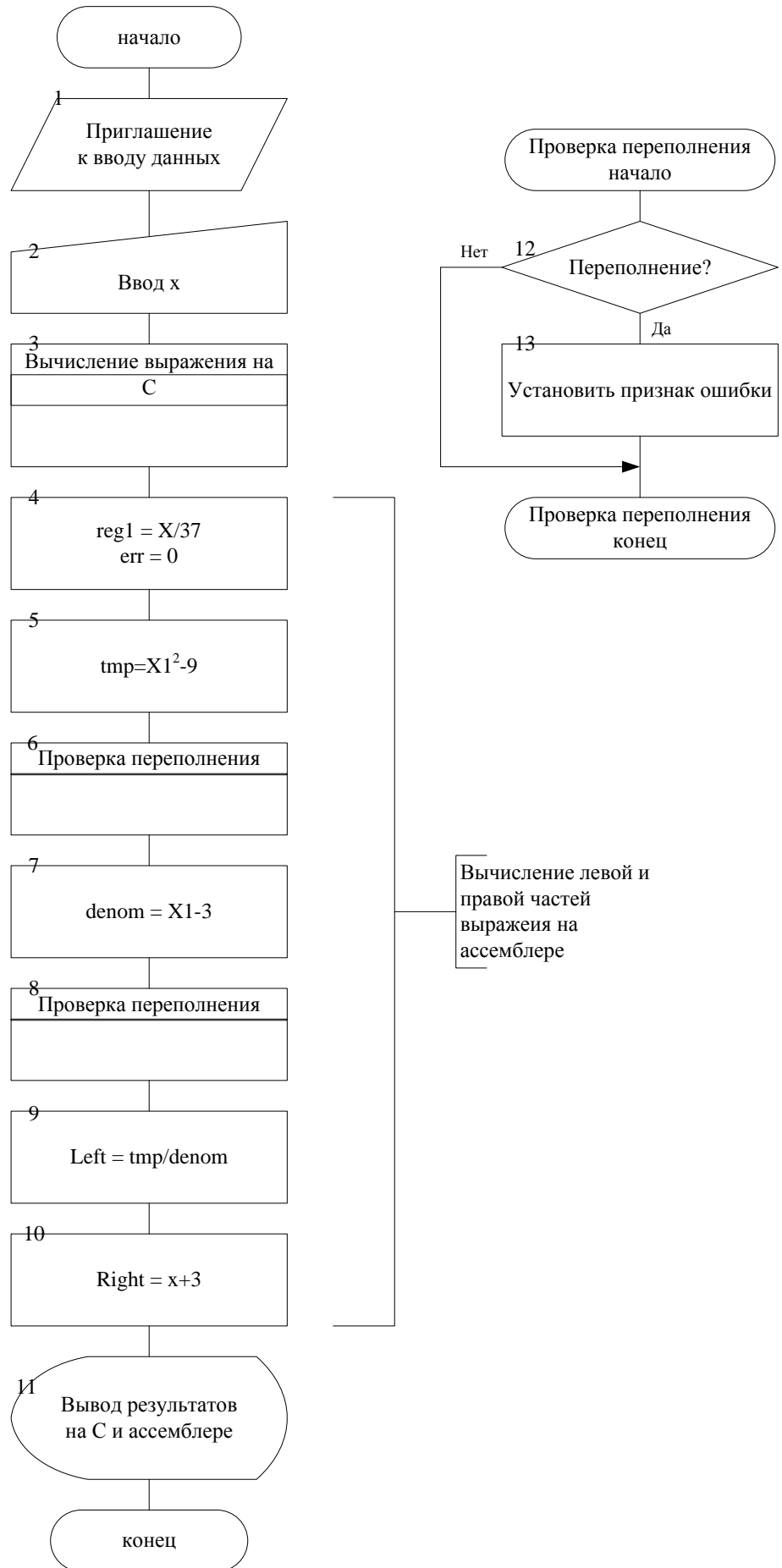


Рисунок 1 - Схема алгоритма вычисления тождества
 $(X^2-9)/(X-3)=X+3$

6. Текст программы с комментариями.

```
//+=====
// File lab_1.c
// Линейная арифметическая программа
// Вариант 37
// Эта программа проверяет арифметическое тождество
//  $(x^2-9)/(x-3) = x+3$ ,
// где  $x=x/37$ 
//
// Для этого вычисляют значение левой и правой частей этого тождества
// на С и на ассемблере
//
// (С) Дужий В.И., 2012
//-----
#include <stdio.h>

#define    DENOM    37
long int   x, x1, left, right, left_a, right_a;
int err;    // Ошибка в левой части выражении на С
int err_la; // Ошибка в левой части выражения в ассемблере

int main()
{
    printf("\n\t\t(С) Дужий В.И., 2012");
    for (;;)
    {
        printf("\n\tПроверить тождество  $(x^2-9)/(x-3) = x+3$ ,");
        printf("\n\tгде  $x=x/37$ ");
        printf("\nПожалуйста, введите целое число X : ");
        scanf("%li",&x);
//===== С =====
// Разделить исходные переменные
        x1 = x/DENOM;
// Сбросить признак ошибки на С
        err = 0;
// Вычислить выражение
        if ((x1-3)==0)
            err = 1;
        else
            left = (x1*x1-9)/(x1-3);
            right= x1+3;
//===== Assembler =====
// Объявить константу для ассемблера
__asm
{
// err_la=err_ra=0; Нет ошибок
        mov     err_la,0
// Разделить исходные переменные на знаменатель DENOM
// x1=x/DENOM
        mov     ebx,DENOM
        mov     eax,x
        cdq
        idiv    ebx
        mov     esi,eax
asm(
// Вычислить левую часть тождества
//  $(x1*x1-9)/(x1-3) \rightarrow left$ 
// Вычислить числитель
```

```

        mov     eax,esi
        imul    eax,eax
        jno     L1
        mov     err_la,1
        jmp     End
L1:      sub     eax,9
// Вычислить знаменатель
        mov     ebx,esi
        sub     ebx,3
        jnz     L2
        mov     err_la,$1
        jmp     End
L2:      cdq
        idiv    ebx
        mov     left_a,eax
// Вычислить правую часть тождества
// x1-3 --> right_a
End:     mov     right_a,esi
        add     right_a,3
    }
// Вывод результатов
    if (err)
    {
        printf("\n*** ( C ) *** Ошибка в левой части выражения");
    }
    else
    {
        printf("Левая часть тождества ( C ): %li",left);
        printf("\nПравая часть тождества ( C ): %li",right);
    }
    if (err_la)
    {
        printf("\n*** (Asm) *** Ошибка в левой части выражения");
    }
    else
    {
        printf("\nЛевая часть тождества (Asm): %li",left_a);
        printf("\nПравая часть тождества (Asm): %li\n",right_a);
    }
    return 0;
}

```

7. Тестовые примеры.

Номер	Исходные данные, X	Ожидаемый результат	Полученный результат	Цель теста
1	3700			Положительное число
2	-3700			Отрицательное число
3	100000			Положительное число
4	-100000			Отрицательное число
5	1850000	???		Переполнение (>+2147000000)
6	111	???		Деление на 0
7	145	???		Деление на 0