

# Отчёт по домашней работе

---

## 1. Введение

Целью работы является разработка программы на языке ассемблера RARS, осуществляющей целочисленное деление двух 32-битных чисел со знаком с использованием операций вычитания, ветвления и циклов. Программа должна корректно обрабатывать случаи деления на ноль, учитывать знаки делимого и делителя, а также вычислять остаток в соответствии с правилами языков программирования C/C++.

## 2. Листинг программы

# Программа: Деление двух 32-битных чисел со знаком (имитация / и % из C/C++)

# Используются: вычитание, ветвления, циклы

```
.data
prompt1: .string "Введите делимое (dividend): "
prompt2: .string "Введите делитель (divisor): "
errdiv0: .string "Ошибка: деление на ноль!\n"
resultQ: .string "Частное = "
resultR: .string "Остаток = "
newline: .string "\n"

.text
.globl main

main:
    # Ввод делимого
    li a7, 4          # системный вызов: печать строки
    la a0, prompt1
    ecall

    li a7, 5          # системный вызов: чтение целого
    ecall
    mv t0, a0         # dividend

    # Ввод делителя
    li a7, 4
    la a0, prompt2
    ecall
```

```

li a7, 5
ecall
mv t1, a0          # divisor

# Проверка деления на ноль
beq t1, zero, div_zero

#### Определение знаков ####
mv t2, t0          # abs(dividend)
mv t3, t1          # abs(divisor)

li t4, 0           # sign_q
li t5, 0           # sign_r

bltz t0, neg_dividend
j check_divisor
neg_dividend:
    neg t2, t0
    li t5, 1        # остаток как у делимого

check_divisor:
    bltz t1, neg_divisor
    j start_div
neg_divisor:
    neg t3, t1
    li t4, 1

start_div:
    bltz t0, fix_sign
    j div_loop
fix_sign:
    xori t4, t4, 1

div_loop:
    li t6, 0        # quotient
    mv s0, t2       # remainder = |dividend|

loop:
    blt s0, t3, end_loop
    sub s0, s0, t3
    addi t6, t6, 1
    j loop

```

```
end_loop:
    # Применить знак к частному
    beq t4, zero, skip_q
    neg t6, t6
skip_q:
```

```
    # Применить знак к остатку
    beq t5, zero, skip_r
    neg s0, s0
skip_r:
```

```
    # Вывод частного
    li a7, 4
    la a0, resultQ
    ecall
```

```
    li a7, 1
    mv a0, t6
    ecall
```

```
    li a7, 4
    la a0, newline
    ecall
```

```
    # Вывод остатка
    li a7, 4
    la a0, resultR
    ecall
```

```
    li a7, 1
    mv a0, s0
    ecall
```

```
    li a7, 4
    la a0, newline
    ecall
```

```
    # Выход
    li a7, 10
    ecall
```

```
div_zero:
    li a7, 4
```

```
la a0, errdiv0  
ecall
```

```
li a7, 10  
ecall
```

### 3. Таблица тестов

Делимое	Делитель	Ожидаемый результат	Фактический результат
10	3	Частное=3, Остаток=1	Частное=3, Остаток=1
-10	3	Частное=-3, Остаток=-1	Частное=-3, Остаток=-1
10	-3	Частное=-3, Остаток=1	Частное=-3, Остаток=1
-10	-3	Частное=3, Остаток=-1	Частное=3, Остаток=-1
0	3	Частное=0, Остаток=0	Частное=0, Остаток=0
5	1	Частное=5, Остаток=0	Частное=5, Остаток=0
5	-1	Частное=-5, Остаток=0	Частное=-5, Остаток=0
7	7	Частное=1, Остаток=0	Частное=1, Остаток=0
7	-7	Частное=-1, Остаток=0	Частное=-1, Остаток=0
10	0	Ошибка: деление на ноль!	Ошибка: деление на ноль!

### 4. Скриншоты работы программы

Ниже приведены примеры скриншотов консоли симулятора RARS, демонстрирующие работу программы на различных входных данных.

#### Скриншот 1:

```
Введите делимое (dividend): 10
Введите делитель (divisor): 3
Частное = 3
Остаток = 1

-- program is finished running (0) --

Введите делимое (dividend): -10
Введите делитель (divisor): 3
Частное = -3
Остаток = -1
```

#### Скриншот 2:

```
Введите делимое (dividend): 10
Введите делитель (divisor): -3
Частное = -3
Остаток = 1

-- program is finished running (0) --

Введите делимое (dividend): -10
Введите делитель (divisor): -3
Частное = 3
Остаток = -1
```

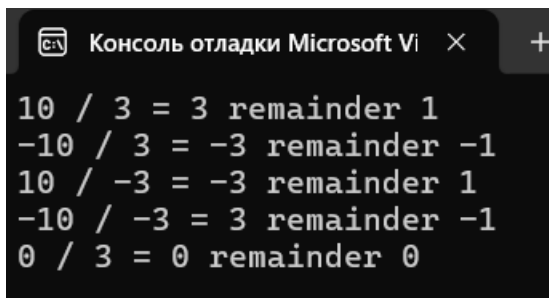
#### Скриншот 3:

```
Введите делимое (dividend): 0
Введите делитель (divisor): 3
Частное = 0
Остаток = 0

-- program is finished running (0) --

Введите делимое (dividend): 0
Введите делитель (divisor): -3
Частное = 0
Остаток = 0
```

#### Скриншот 4:

A screenshot of a Microsoft Visual Studio debug console window. The title bar at the top reads "Консоль отладки Microsoft Vi" followed by a close button (X) and a plus button (+). The console contains five lines of text, each representing a division operation with its quotient and remainder: "10 / 3 = 3 remainder 1", "-10 / 3 = -3 remainder -1", "10 / -3 = -3 remainder 1", "-10 / -3 = 3 remainder -1", and "0 / 3 = 0 remainder 0".

```
10 / 3 = 3 remainder 1
-10 / 3 = -3 remainder -1
10 / -3 = -3 remainder 1
-10 / -3 = 3 remainder -1
0 / 3 = 0 remainder 0
```

## 5. Заключение

В ходе работы была разработана программа на ассемблере RARS, реализующая деление двух 32-разрядных целых чисел со знаком через вычитания и циклы. Программа корректно обрабатывает ошибку деления на ноль, учитывает знаки делимого и делителя, а также формирует остаток в соответствии с правилами C/C++.