

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ "САНКТ-ПЕТЕРБУРГСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М. А.
БОНЧ-БРУЕВИЧА"

Факультет инфокоммуникационных сетей и систем
Кафедра программной инженерии и вычислительной техники

ЛАБОРАТОРНАЯ РАБОТА №2
«ОРГАНИЗАЦИЯ УСЛОВНЫХ ПЕРЕХОДОВ»
по дисциплине «Машинно-зависимые языки программирования»

Выполнил:
студент 2 курса
дневного отделения
группы ИКПИ-23
Даненко Д. А.

Санкт-Петербург 2023

А. Постановка задачи

Вычислить заданное условное целочисленное выражение для данных в форматах **INTEGER** (*signed int*), **WORD** (*unsigned int*), используя команды сравнения, условного и безусловного переходов.

$$\begin{array}{ll} 11 & , a = b \\ a * a / b & , \text{если } a < b \\ b * a / 11 & , \text{если } a > b \end{array}$$

Результат X тоже целочисленный и его диапазон (формат) зависит от специфики решаемого условного выражения. Исходные данные должны вводиться корректно. Результат также должен быть проверен на область допустимых значений. Обмен данными между Си и ASM — модулем должен осуществляться через глобальные переменные, определенные в модуле Си.

Б. Разработка алгоритма

Все два формата данных будут продемонстрированы в одном проекте.

Входные данные состоят из 4 чисел A , B (*signed int*, *unsigned int*).

Выходные данные состоят из числа X — результата вычисления на языке Си и ASM для каждого типа задания (*signed int*, *unsigned int*).

Промежуточные данные: переменная типа long под названием "t" для проверки на корректность ввода чисел пользователем.

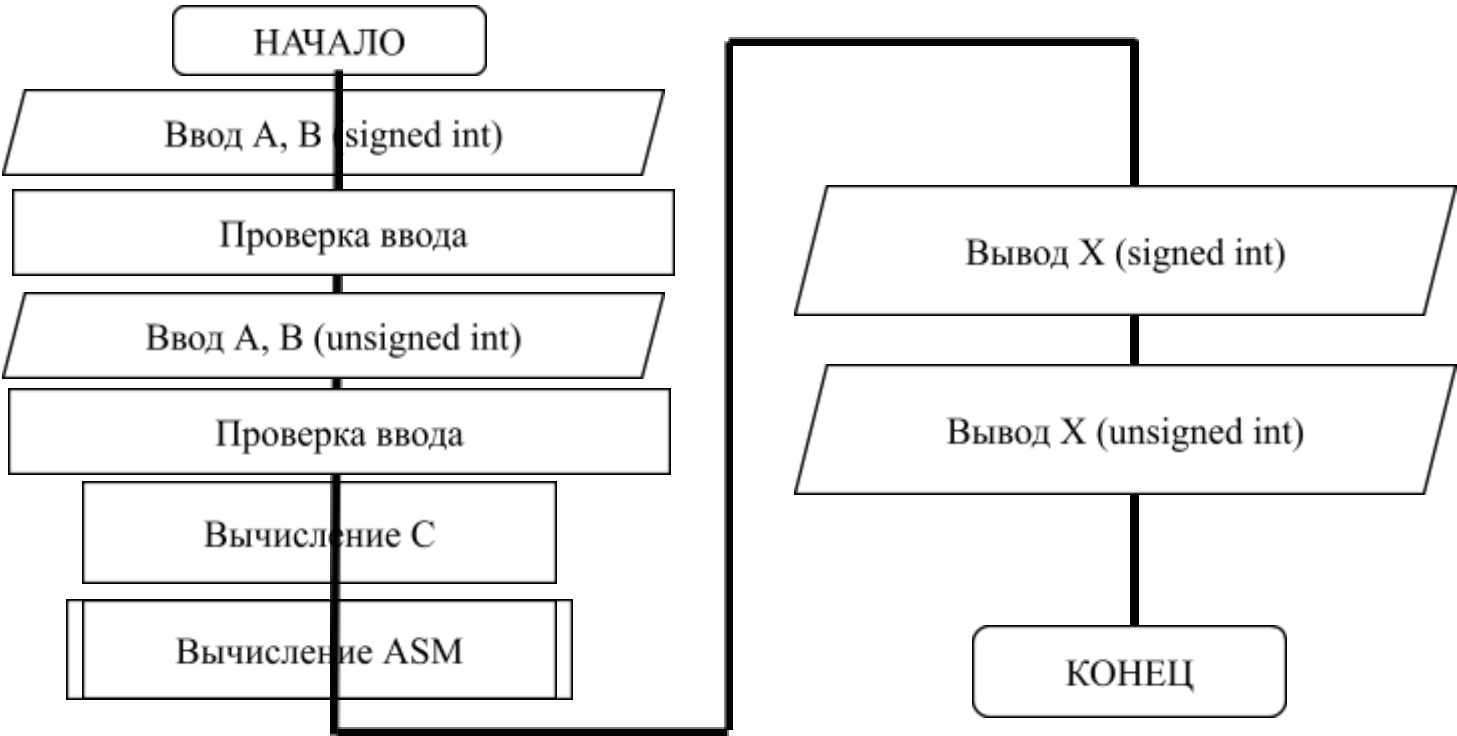
Размеры типов данных.

$\text{sizeof}(\text{signed int}) = \text{sizeof}(\text{unsigned int}) = \text{WORD}$ (16 бит)

В. Таблица идентификаторов

N	Обозначение в задаче	Идентификатор	Назначение
1	A (<i>signed int</i>)	ia	Входные данные
2	B (<i>signed int</i>)	ib	
4	A (<i>unsigned int</i>)	wa	
5	B (<i>unsigned int</i>)	wb	
14	X (<i>signed int</i>)	ires	Выходные данные
15	X (<i>unsigned int</i>)	wres	

Г. Схема алгоритма



Д. Таблица результатов

Результаты вычислений приведены ниже в таблице вычислений.

Тип	A	B	X
Signed Int	-32768	-32767	C: -32769 ASM: -32769
	32767	-32768	C: -97609914 ASM: -97609914
	-123456	-123456	C: 11 ASM: 11
Unsigned Word	65535	65534	C: 390433699 ASM: 390433699
	65534	65535	C: 65532 ASM: 65532
	0	0	C: 11 ASM: 11

```
> $ ./c
a = -32768
b = -32767
-32769
-32769
```

```
> $ ./c
a = 32767
b = -32768
-97609914
-97609914
```

```
> $ ./c
a = -123456
b = -123456
11
11
```

> \$./c	> \$./c	> \$./c
a = 65535	a = 65534	a = 0
b = 65534	b = 65535	b = 0
390433699	65533	11
390433699	65532	11

Е. Программа

```

/*                                Файл C                                */

#include <stdint.h>
#include <stdio.h>
#include <stdlib.h>
extern void uint_asm_func(void);
extern void int_asm_func(void);

int32_t ia, ib;
int32_t ires;

uint16_t wa, wb;
uint32_t wres;

void C_func() {
    if (ia == ib) {
        printf("11\n");
    } else if (ia < ib) {
        printf("%d\n", (int32_t)ia * (int32_t)ia / (int32_t)ib);
    } else {
        printf("%d\n", (int32_t)ib * (int32_t)ia / 11);
    }
}

/*
    mov cx, ax
    mov ax, dx
    shl ax, 16
    add ax, cx
*/

int main(int argc, char *argv[]) {
    // 11 , a = b
    // a*a/b , если a < b
    // b*a/11 , если a > b

    // printf("a = ");
    // scanf("%d", &wa);
    // printf("b = ");
    // scanf("%d", &wb);
    wa = 65535;
    wb = 65534;
    // C_func(wa, wb);

```

```

// uint_asm_func();
// printf("%u\n", wres);

printf("a = ");
scanf("%d", &ia);
printf("b = ");
scanf("%d", &ib);
C_func(ia, ib);
int_asm_func();
printf("%u\n", ires);

return 0;
}
;                                     файл ASM
section .data
    extern ia,    wa
    extern ib,    wb
    extern ires, wres
    extern var
section .text
    global uint_asm_func
    global int_asm_func

uint_asm_func:
    xor    eax, eax
    xor    ebx, ebx
    xor    ecx, ecx
    xor    edx, edx

    mov    ax,    [wa]    ; ax = wa
    mov    bx,    [wb]    ; bx = wb
    cmp    ax,    bx    ; ax==bx?
    je     @a_is_b
    jb     @a_lower_than_b
    ja     @a_higher_than_b

@a_is_b:
    mov    eax, 11    ; eax = 11
    mov    [wres], eax    ; wres = 11
    ret

@a_lower_than_b:
    mul    ax                ; ax:dx = ax * ax
    div    bx                ;
    mov    [wres], ax        ;
    ret

@a_higher_than_b:
    xor    eax,    eax
    xor    ebx,    ebx

```

```

mov    ax,    [wa]    ; eax = a
mov    bx,    [wb]    ; ebx = b
mul    ebx                ; eax:edx = a*b

mov    ecx, 11    ; ecx = 11
div    ecx                ; eax:edx / eax

mov    [wres], eax    ; wres = eax
ret

```

int_asm_func:

```

xor    eax, eax
xor    ebx, ebx
xor    ecx, ecx
xor    edx, edx

```

```

mov    ax,    [ia]    ; ax = wa
mov    bx,    [ib]    ; bx = wb
cmp    ax,    bx    ; ax==bx?
je     @a_is_b_int
jb     @a_lower_than_b_int
ja     @a_higher_than_b_int

```

@a_is_b_int:

```

mov    eax, 11    ; eax = 11
mov    [ires], eax    ; wres = 11
ret

```

@a_lower_than_b_int:

```

imul    ax                ; ax:dx = ax * ax
idiv    bx                ;
mov    [ires], ax        ;
ret

```

@a_higher_than_b_int:

```

xor    eax,    eax
xor    ebx,    ebx
mov    ax,    [ia]    ; eax = a
mov    bx,    [ib]    ; ebx = b
imul    ebx                ; eax:edx = a*b

mov    ecx, 11    ; ecx = 11
idiv    ecx                ; eax:edx / eax

mov    [ires], eax    ; wres = eax
ret

```

Ж. Выводы

Тот факт, что результаты, выполненные на ASM, достаточно совпадают с результатами, выполненными на C, свидетельствует о том, что программа составлена правильно.