**Задание 1.** Разработайте ассемблерную функцию, вычисляющую целое

выражение от целого аргумента (в соответствии с вариантом), а также

головную программу на языке C++, использующую разработанную функцию.

Код

#include <iostream>

extern "C" int calc(int x);

using namespace std;

\_\_asm\_\_

(

"\_calc:\n"

" movl 4(%esp), %eax\n"

" imul $-4,%eax\n"

" add $4,%eax\n"

" ret\n"

);

int main(void)

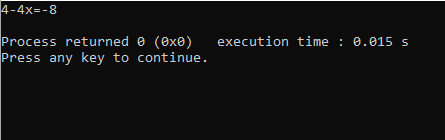
{

int x = 3;

int y = calc(x);

cout<<"4-4x="<<y<<endl;

}



**Задание 2.** Разработайте программу, целиком написанную на ассемблере,

вычисляющую значение для и выводящую полученное

значение на стандартный вывод с использованием библиотеки stdlib (в частности,

функции printf).

.data

printf\_format:

.string "%d\n"

.globl \_main

\_main:

movl $13, %eax

imul $-4,%eax

add $4,%eax

pushl %eax

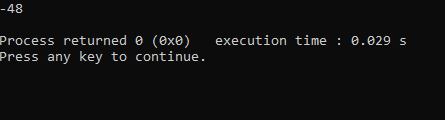
pushl $printf\_format

call \_printf

addl $8, %esp

movl $0, %eax

ret



**Задание 3.** Опишите функцию на произвольном языке высокого уровня

(включая C/C++) и вызовите её из ассемблерной функции.

1.Вывод двух параметров на экран с пояснением

#include <iostream>

#include <locale.h>

using namespace std;

extern "C" void fun1();

extern "C" void fun2();

void fun2()

{

int x,y;

cin>>x>> y;

cout <<"Введены числа: " << x <<"; "<< y;

}

int main ()

{

setlocale(0,"Russian");

fun1();

return 0;

}

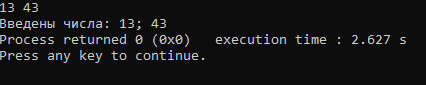
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

.global \_fun1

\_fun1:

call \_fun2

ret



**Задание 4.** Бонус (+2 балла). Опишите на ассемблере одну подпрограмму

с параметрами и результатами и и вызовите её из другой

ассемблерной программы.

.data

printf\_format:

.string "%d\n"

a:

.int 8

b:

.int 7

x:

.int 0

y:

.int 0

\_calc:

movl a, %eax

imul %eax, %eax

movl %eax, x

mov b, %eax

imul %eax, %eax

sub x, %eax

imul $-1, %eax

movl %eax, x

movl a, %eax

imul $2, %eax

movl %eax, y

movl b, %eax

imul y, %eax

movl %eax, y

ret

.globl \_main

\_main:

call \_calc

movl x, %eax

pushl %eax

pushl $printf\_format

call \_printf

addl $8, %esp

movl $0, %eax

movl y, %eax

pushl %eax

pushl $printf\_format

call \_printf

addl $8, %esp

movl $0, %eax

ret

