Лабораторная работа №3.

Номер 1.

$(№ - 1)\%5 + 1$	Вариант
1	$y(x) = 1 + x/2$

main.cpp

```
#include <stdio.h>
using namespace std;

extern "C" int tass(int x);

int main()
{
        for (int i = 0; i < 28; i+=4)
        {
                printf("x = %d y(x) = %d ",i, tass(i));
                printf("\n");
        }

        return 0;
}
```

tass.s

```
.global tass
tass:
    mov 4(%esp), %eax  #Получаем переменную
    shr $1, %eax #Сдвиг вправо на 1 соответсвует делению на 2
    add $1, %eax #Прибавляем единицу
ret  #Возврат значения
```

Вывод:

```
aliya@aliya-MS-7B22:/media/3T/tst/lrr$ gcc -m32 main.cpp tass.s
aliya@aliya-MS-7B22:/media/3T/tst/lrr$ ./a.out
x = 0 y(x) = 1
x = 4 y(x) = 3
x = 8 y(x) = 5
x = 12 y(x) = 7
x = 16 y(x) = 9
x = 20 y(x) = 11
x = 24 y(x) = 13
```

Номер 2.

Задание 2. Разработайте программу, целиком написанную на ассемблере, вычисляющую значение $y(x)$ для $x = 13$ и выводящую полученное значение на стандартный вывод с использованием библиотеки libc (в частности, функции printf).

tass2.s

```
.data
printf_format:
    .string "%d\n"
.globl main
main:
    movl $13, %eax #eax=13
    shr $1, %eax  #Сдвиг вправо на единицу
    add $1, %eax #Добавление единицы
    pushl %eax  #eax в стек
    pushl $printf_format #printf_format в стек
    call printf #вызов функции
    addl $8, %esp  #очищаем стек
    movl $0, %eax #eax=0
    ret
```

Вывод:

```
aliya@aliya-MS-7B22:/media/3T/tst/lrr$ gcc -m32 task2.s
aliya@aliya-MS-7B22:/media/3T/tst/lrr$ ./a.out
7
```

Номер 3.

Задание 3. Опишите функцию на произвольном языке высокого уровня (включая C/C++) и вызовите её из ассемблерной функции.

(№ − 1)%3 +1	Вариант
1	Вывод двух параметров на экран с пояснениями

tass3_m.cpp

```
#include <stdio.h>

extern "C" void Watch(int first, int second)
{
    printf("First number  %d\n",first);
     printf("Second number  %d\n",second);
}
```

tass3_f.cpp

```
.data
FIRST: .int 33
SECOND: .int 22
.global main
main:
/*Добавляем в стек*/
pushl SECOND
pushl FIRST
call Watch  #Вызов функции
addl $8, %esp
xor %eax, %eax
ret
```

Вывод:

```
aliya@aliya-MS-7B22:/media/3T/lrr$ gcc -m32 tass3_m.cpp tass3_f.s
aliya@aliya-MS-7B22:/media/3T/lrr$ ./a.out
First number  33
Second number  22
```

Номер 4.

Задание 4. Бонус (+2 балла). Опишите на ассемблере одну подпрограмму с параметрами $a, b, \ldots$ и результатами x и y и вызовите её из другой ассемблерной программы.

$(№ - 1)\%2 + 1$	Вариант
1	$\begin{cases} x = a + c \cdot b \\ y = a - c \cdot b \end{cases}$

Примем нестандартное соглашение, отличающееся от cdecl только в том, что: возврат значения по возможности выполняется не через eax, а через eax и ecx-два указателя или два целых числа до 4х байт, при этом, если нужно возвратить одно целое число ecx зануляем, т.е. два нуля через нашу функцию передать нельзя, 0-специальное значение для ecx.

tass4.s

```
.data
a: .int 100
b: .int 21
c: .int 11
msg: .string "a = 100 b = 21 c=11 \n x= %d y= %d\n"
.global main
main:
pushl a
pushl b
pushl c
call operator
addl $3*4, %esp
push %ecx
push %eax
push $msg
call printf
addl $12, %esp
movl $0, %ecx
movl $0, %eax
ret
```

operator.s

```
.global operator
operator:
mov 4(%esp), %edx
mov 8(%esp), %ecx
mov 12(%esp), %eax
imul %edx,%ecx
mov %eax,%edx
add %ecx,%eax
sub %ecx,%edx
mov %edx,%ecx
xor %edx,%edx
ret
```

Вывод:

```
aliya@aliya-MS-7B22:/media/3T/tst/lrr$ gcc -m32 tass4.s operator.s
aliya@aliya-MS-7B22:/media/3T/tst/lrr$ ./a.out
a = 100 b = 21 c=11
 x= 331 y= -100
```