

Лабораторная работа №6.
Условия.
(Колпаков, Краснов, Шокоров)

Задание 1. Вычислите для заданных целых x, y .

3	$f(x, y) = \begin{cases} y/x & y \% x = 0 \\ y \cdot x, & y \% x \neq 0 \end{cases}$
---	--

```
#include <iostream>

using std::cout;
using std::cin;
using std::endl;

int main(void) {
    int firstOperand, secondOperand, result;
    cout << "Enter 'y' and 'x': ";
    cin >> firstOperand >> secondOperand;

    asm __volatile__ (
        "mov %[Y], %%eax\n"           // eax = firstOperand.
        "mov $0, %%edx\n"             // зануляем регистр от мусора.
        "idiv %[X]\n"                 // firstOperand / secondOperand.
        "cmpl $0, %%edx\n"            // if (firstOperand % secondOperand == 0).
        "jne not_even\n"              // прыгаем, если не равно.
        "mov %%eax, %[RESULT]\n"      // result = firstOperand / secondOperand.
        "jmp end\n"                   // прыгаем на выход.
        "not_even:\n"
        "movl %[Y], %%eax\n"           // eax = firstOperand. (приобретаем)
        "imul %[X], %%eax\n"          // eax = firstOperand * secondOperand
        (докупаем) .
        "movl %%eax, %[RESULT]\n"      // result = firstOperand * secondOperand
        (фиксируем прибыль) .
        "end:\n"
        : [RESULT]="=rm"(result)
        : [Y]"rm"(firstOperand), [X]"rm"(secondOperand)
        : "cc", "eax"
    );

    cout << "Result: " << result << endl;
    return 0;
}
```

```
kdator@kdator:~/study$ ./main
Enter 'y' and 'x': 5 7
Result: 35
kdator@kdator:~/study$ ./main
Enter 'y' and 'x': 8 4
Result: 2
kdator@kdator:~/study$ ./main
Enter 'y' and 'x': 6 6
Result: 1
```

Задание 2. Вычислите для заданного вещественного x .

3	$y(x) = \begin{cases} 0, & x \leq 0.8 \\ x - 0.8, & x > 0.8 \end{cases}$
---	--

```
#include <iostream>

int main() {
    double x, y, cmp = 0.8;

    std::cout << "Enter 'x': ";
    std::cin >> x;

    asm (
        "fldl %[CMP]\n"
        "fldl %[X]\n"
        "fsubp \n\t"
        "ftst\n"
        "fnstsw\n"
        "sahf\n"
        "jnb greater\n"
        "fstpl %[Y]\n"
        "jmp end\n"
        "greater:\n"
        "fldz\n"
        "fstpl %[Y]\n\t"
        "end:\n"
        : [Y] "=rm" (y)
        : [X] "rm" (x), [CMP] "rm" (cmp)
    );

    std::cout << "Result: " << y << std::endl;

    return 0;
}
```

```

Enter 'x': 0.85
Result: 0.05
PS C:\Users\NikitaShokorov\Desktop\LabsAndOthers\Архвс\Лаба 6> ./main
Enter 'x': 0.9
Result: 0.1
PS C:\Users\NikitaShokorov\Desktop\LabsAndOthers\Архвс\Лаба 6> ./main
Enter 'x': 0.2
Result: 0

```

Задание 3. Задайте с клавиатуры N и напечатайте первые N членов последовательности (целой).

1	Числа Фибоначчи: $\varphi_0 = \varphi_1 = 1, \varphi_{i+1} = \varphi_i + \varphi_{i-1}$
---	---

```

.data
OUT:
    .asciz "%d\n"
H:
    .asciz "Enter a number: "
INPUT:
    .asciz "%d"

.text
.globl main
main:
    pushq %rbp
    sub $16, %rsp
    // ВЫЗОВ СООБЩЕНИЯ ГОСТЕВОГО ВВОДА.
    leaq H(%rip), %rdi
    movb $0, %al
    callq printf

    // ВЫЗОВ scanf.
    leaq (%rsp), %rsi
    leaq INPUT(%rip), %rdi
    movb $0, %al
    callq scanf
    movl (%rsp), %eax

    // инициализируем нужные регистры единицами.
    movq $1, %rbx
    movq $1, %r12
    movq $1, %r13
    movq $1, %r14

```

```
// начинаем итерацию цикла.
begin_iteration:

movl (%rsp), %eax
cmp %rax, %rbx
jg end

// вычисление числа.
movq %r14, %r13

movq %r13, %r15
addq %r12, %r15
movq %r15, %r14

movq %r13, %r12

// показываем текущее число.
movq %r12, %rsi
leaq OUT(%rip), %rdi
movb $0, %al
callq printf

// инкремент цикла.
addq $1, %rbx

jmp begin_iteration

end:

add $16, %rsp
popq %rbp
xor %eax, %eax
ret
```

```

kdator@kdator:~/study$ ./fib
Enter a number: 10
1
2
3
5
8
13
21
34
55
89

```

Задание 4. Найдите с заданной точностью ϵ сумму ряда (если это возможно). Если ряд расходится, выведите сообщение об этом.

$$S = \sum_{i=1}^{\infty} (-1)^i \frac{1}{i}$$

main.S

```

.data
    mes:
        .string "sum = %.20g"
    e:
        .double 0.000001
.global _main
_main:
    fldl e
    pushl $-1
    fildl (%esp)
    fldl
    fldl
    fldl
    fdivr %st,%st(1)
    fmul %st(3),%st(0)
    fldl
    fldl
    fchs
    fldl

    #sub $8, %esp
    #fstpl (%esp)
    #pushl $mes
    #call _printf

begin_iteration:

```

```

fcomi %st(7),%st
jbe end_loop
fxch %st(6)
fchs
fxch %st(6)
fxch %st(5)
fadd %st(4),%st(0)
fst %st(5)
fdivr %st(6),%st
#fmul %st(6),%st(0)
fadd %st,%st(1)
fxch %st(2)
fsub %st(2),%st
fabs
jmp begin_iteration
end_loop:
sub $8, %esp
fstp %st(0)
fstpl (%esp)
pushl $mes
call _printf
addl $12, %esp
fstp %st(0)
fstp %st(0)
fstp %st(0)
fstp %st(0)
fstp %st(0)
ret

```

e = 0.000001

```

PS C:\Users\NikitaShokorov\Desktop\LabsAndOthers\Архвс\Лаба 6> g++ -m32 -o main main.S
PS C:\Users\NikitaShokorov\Desktop\LabsAndOthers\Архвс\Лаба 6> ./main
sum = -0.69314743055975769

```

Сумма ряда в онлайн калькуляторе

Численный ответ:
[src]

-0.693147180559945309417232121458

Задание 5. Бонус (+2 балла). Реализуйте умножение двух целых чисел без знака ($x \cdot y$) «в столбик» (то есть не как сумму $x + \dots + x$), не используя mul/imul и команды сопроцессора.

```

#include <iostream>

using std::cout;
using std::cin;
using std::endl;

int main(void) {
    int x, y, result;
    cout << "Enter 'x' and 'y': ";
    cin >> x >> y;

    if (cin.fail()) {
        cin.clear();
        cin.ignore();
        cout << "Error: wrong parameters" << endl;
        return 0;
    }

    if (y < 0) {
        cout << "Error: parameter 'y' must be positive" << endl;
        return 0;
    }

    asm __volatile__ (
        "movl $0, %%eax\n"           // сюда будем записывать результат
        "movl %[Y], %%ebx\n"         // кладём y.
        "movl %[X], %%edx\n"         // кладём x.
        "jmp begin_loop\n"           // начинаем проверку цикла.
        "iteration:\n"               // переход на новую итерацию.
        "test $0x1, %%ebx\n"         // побитовое И без изменения ebx.
        "je miss_sum\n"             // смотрим флаги на основе предыдущего
        "addl %%edx, %%eax\n"         // иначе добавляем к eax множитель x.
        "miss_sum:\n"               // переход к смещению чисел.
        "sal $1, %%edx\n"            // двигаем влево множитель x.
        "sar $1, %%ebx\n"            // двигаем вправо сомножитель y.
        "begin_loop:\n"             // переход к проверке условия цикла.
        "cmpl $0, %%ebx\n"           // if (y != 0).
        "jne iteration\n"           // если не равен, возвращаемся на новую
        "movl %%eax, %[RESULT]\n"    // записываем результат на выход.
        : [RESULT]="=rm"(result)
        : [Y]"rm"(y), [X]"rm"(x)

```

```
        : "cc", "eax"  
    );  
  
    cout << "Result: " << result << endl;  
    return 0;  
}
```

```
kdator@kdator:~/study$ ./main  
Enter 'x' and 'y': 7 8  
Result: 56  
kdator@kdator:~/study$ ./main  
Enter 'x' and 'y': 4 10  
Result: 40  
kdator@kdator:~/study$ ./main  
Enter 'x' and 'y': 10 6  
Result: 60  
kdator@kdator:~/study$ ./main  
Enter 'x' and 'y': -1 6  
Result: -6  
kdator@kdator:~/study$ ./main  
Enter 'x' and 'y': -65 2  
Result: -130  
kdator@kdator:~/study$ ./main  
Enter 'x' and 'y': 0 0  
Result: 0  
kdator@kdator:~/study$ ./main  
Enter 'x' and 'y': -1 0  
Result: 0  
kdator@kdator:~/study$ ./main  
Enter 'x' and 'y': 0 -1  
Error: parameter 'y' must be positive  
kdator@kdator:~/study$
```