**Лабораторная работа 6.**

Задание 1.

#include <stdio.h>

extern "C" int func(int x, int y);

int main() {

    int x, y;

    scanf("%d %d", &x, &y);

    printf("func(%d, %d) = %d\n", x, y, func(x, y));

}

.text

.globl func

.type func, @function

func:

movl %esi, %eax

cltd

idivl %edi

testl %edx, %edx

je .if

movl %esi, %eax

imull %edi, %eax

ret

.if:

movl %esi, %eax

cltd

idivl %edi

ret

2 4

func(2, 4) = 2

4 2

func(4, 2) = 8

Задание 2.

#include <stdio.h>

#include <math.h>

extern "C" double funcc(double x);

int main() {

    double x;

    scanf("%le", &x);

    printf("funcc(%le) = %le\n", x, funcc(x));

}

.text

.globl funcc

.type funcc, @function

funcc:

movsd .consts(%rip), %xmm1

ucomisd %xmm0, %xmm1

jnb .if

subsd %xmm1, %xmm0

ret

.if:

pxor %xmm0, %xmm0

ret

.consts:

.long 2576980378

.long 1072273817

0.7

funcc(7.000000e-01) = 0.000000e+00

0.9

funcc(9.000000e-01) = 1.000000e-01

1

funcc(1.000000e+00) = 2.000000e-01

Задание 3.

#include <stdio.h>

#include <math.h>

extern "C" int fib(int N, int\* result);

int main() {

    int x;

    scanf("%d", &x);

    int\* fibs = (int\*) malloc((x + 1) \* sizeof(int));

    fib(x, fibs);

    for (int i = 0; i < (x + 1); ++i) {

        printf("fib(%d) = %d\n", i, fibs[i]);

    }

    free(fibs);

}

.text

.globl fib

.type fib, @function

fib:

movl $1, (%rsi)

movl $1, 4(%rsi)

leal 1(%rdi), %eax

cmpl $2, %eax

jle end

movq %rsi, %rax

leal -2(%rdi), %edx

leaq 4(%rsi,%rdx,4), %rcx

for:

movl (%rax), %edx

addl 4(%rax), %edx

movl %edx, 8(%rax)

addq $4, %rax

cmpq %rcx, %rax

jne for

end:

rep ret

6

fib(0) = 1

fib(1) = 1

fib(2) = 2

fib(3) = 3

fib(4) = 5

fib(5) = 8

fib(6) = 13

Задание 4.

Аналитически доказуемо, что гармонический ряд расходится, следовательно найти сумму невозможно.

#include <stdio.h>

#include <math.h>

extern "C" bool ser(double eps, double\* result);

int main() {

    double eps;

    scanf("%le", &eps);

    double res = 0;

    if (!ser(eps, &res))

        printf("the series does not converge\n");

    return 0;

}

.text

.globl ser

.type ser, @function

ser:

movl $0, %eax

ret

0.001

the series does not converge