

Лабораторная работа №6.

Программирование ветвящихся и циклических алгоритмов

Цель работы: ознакомиться с командами условного и безусловного перехода; научиться использовать их для реализации ветвящихся и циклических алгоритмов.

Л6.1. Задание на лабораторную работу

Задание 1. Вычислите для заданных целых x, y

$(N^{\circ} - 1) \% 3 + 1$	Вариант
1	$f(x) = \begin{cases} x/2, & x \% 2 = 0 \\ 3x + 1, & x \% 2 \neq 0 \end{cases}$

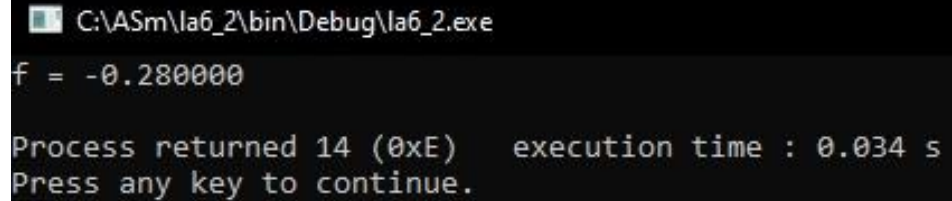
```
.data
printf_format:
.string "f = %d\n"
x:
.int 1
y:
.int 2
.globl main
main:
movl x, %eax
add y,%eax
add $-4,%eax
cmpl $0, %eax
jg s2
movl x, %eax
jmp end
s2:
movl y, %eax
add $2,%eax
end:
pushl %eax
pushl $printf_format
call printf
addl $8, %esp
movl $0, %eax
ret
```

```
C:\ASm\la6_1\bin\Debug\la6_1.exe
f = 1
Process returned 0 (0x0)   execution time : 0.104 s
Press any key to continue.
```

Задание 2. Вычислите для заданного вещественного x

$(N-1)\%11+1$	Вариант
8	$y(x) = \begin{cases} 2x \cdot (x-1) & x-1 < 5 \\ -2x, & x-1 \geq 5 \end{cases}$

```
.data
printf_format:
.string "f = %f\n"
x:
.double 1
y:
.double 0
a:
.double -1.28
.globl main
main:
fldl x //Загружаем x
ftst //Сравниваем x с нулем
fstsw %ax //Сохраняем текущее значение регистра SR в приемник
sahf //Загружаем флаги для сравнения
jbe s2
fldl a
fadd
jmp end
s2:
fldz
end:
fstpl (y) st(0) в y
pushl (y+4)
pushl (y)
push $printf_format
call printf
add $12, %esp
ret
```



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

2	Арифметическая прогрессия $a_0 = 0, a_{i+1} = a_i + x$, x ввести с клавиатуры
---	--

```
.data
input: .string "%d"
output: .string "a[%d]=%d\n"
inputN: .string "Enter N: "
inputX: .string "Enter X: "
x:
```

```

        .int 0
n:
        .int 0
i:
        .int -1
a:
        .int 0
.globl _main
_main:
    pushl %ebp
    movl %esp, %ebp
    subl $40, %esp
    movl $inputX, (%esp)
    call _printf
    leal 36(%esp), %eax
    movl %eax, 4(%esp)
    movl $input, (%esp)
    call _scanf
    movl -4(%ebp), %eax
    movl %eax, x
    movl %ebp, %esp
    popl %ebp
    pushl %ebp
    movl %esp, %ebp
    subl $40, %esp
    movl $inputN, (%esp)
    call _printf
    leal 36(%esp), %eax
    movl %eax, 4(%esp)
    movl $input, (%esp)
    call _scanf
    movl -4(%ebp), %eax
    movl %eax, n
    movl %ebp, %esp
    popl %ebp
_begin_iteration:
    cmpl $0, n
    je _end_loop
    addl $-1, n
    addl $1, i
_end_iteration:
    push a
    push a
    push i
    push $output
    call _printf
    addl $16, %esp
    movl x, %eax
    addl %eax, a
    jmp _begin_iteration
_end_loop:
    ret

```

```

C:\ASm\la6_3\bin\Debug\la6_3.exe
Enter X: 4
Enter N: 10
a[0]=0
a[1]=4
a[2]=8
a[3]=12
a[4]=16
a[5]=20
a[6]=24
a[7]=28
a[8]=32
a[9]=36

Process returned 4 (0x4)   execution time : 7.715 s
Press any key to continue.

```

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

4	$S = \sum_{i=0}^{\infty} \frac{1}{2^i}$
---	---

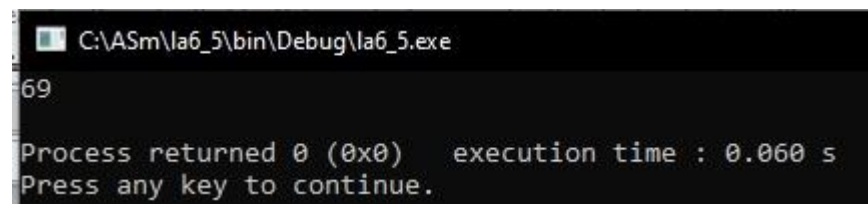
```
#include <iostream>
using namespace std;
int main()
{
    int i=4;
    float sum;
    float x = 2, y;
    asm(
        "fldl \n"
        "fldl \n"
        "begin: \n"
        "sub $1,%[i] \n"
        "jz end\n"
        "fld %[X]\n"
        "fdivrp %%st(1) \n"
        "fadd %%st(0),%%st(1)\n"
        "jmp begin\n"
        "end: \n"
        "fstp %[sum]\n"
        "fstp %[sum]\n"
        "ffree %%st(0)\n"
        "ffree %%st(0)\n"
        :[sum]="m"(y),[i]+"r"(i)
        :[X]"m"(x)
        : "cc"
    );
    cout << y;
    return 0; }
```

```
C:\ASm\la6_4\bin\Debug\la6_4.exe
1.875
Process returned 0 (0x0)   execution time : 0.049 s
Press any key to continue.
```

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

```
#include <iostream>
using namespace std;
int main()
{
    int x = 3, y = 23;
    int R;
    asm (
        "xor %%eax,%%eax\n"
```

```
"mov $0x10,%%ecx \n" //число разрядов
"a0: \n"
"cmp $0,%[X] \n" //X сдвигается до тех пор, пока в нем
"jz a3 \n" //есть единицы
"shr $1,%[X] \n" //сдвигаем X вправо и проверяем
"jnc a2 \n" //есть ли в очередном разряде единица
"add %[Y],%%eax \n" //если есть -- складываем
"a2: \n"
"add %[Y],%[Y] \n" //увеличиваем Y в два раза
"loop a0 \n" //пока не обработаем все 16 разряд
"a3:\n"
"movl %%eax, %[R]\n"
: [R]="m" (R),[X]+"r" (x),[Y]+"r"(y)
:
: "cc", "%ecx", "%eax"
);
cout << R <<endl;
return 0;
}
```



```
C:\ASm\la6_5\bin\Debug\la6_5.exe
69
Process returned 0 (0x0) execution time : 0.060 s
Press any key to continue.
```