Programação em Sistemas Computacionais
Primeiro Mini-teste na turma: **LI31D**, 2 de Maio de 2011

Nome: _____ **Número:** _____ **Turma:** LI31D |A

Considere, em todas as questões, a utilização de uma arquitectura *little endian* a 32 bits.

1. [7] Considere a função `ex1()`, em *assembly* IA32

```
    .intel_syntax noprefix
    .text
    .globl ex1
ex1:
    mov         eax, [esp+4]
    sub         eax, [esp+8]
    jb          ex1_L1
    inc         eax
ex1_a_L1:
    mov         cl, [esp+12]
    movsx       ecx, cl
    cmp         ecx, 0
    jl          ex1_L2
    inc         eax
ex1_a_L2:
    ret
```

eax	ecx

Preencha o quadro com as representações hexadecimais presentes nos registos EAX e ECX ao longo da execução da função ex1, para a seguinte invocação a partir de código em C:

```
ex1(0x80000000,1, (unsigned char)0xFE);
```

2. [7] Considere a função `f()` implementada em *assembly* IA32

```
f:
    push    ebp
    mov     ebp, esp
    push    esi
    xor     esi, esi
    push    edi
    mov     edi, [ebp+16]
    push    ebx
    mov     ebx, [ebp+8]
    mov     al, [ebx]
    test    al, al
    jz      f_end
f_L2:
    mov     [esp], edi
    mov     [esp+4], eax
    call    [ebp+12]
    cmp     eax, 0
    jl      f_L1
    add     esi, eax
f_L1:
    inc     ebx
    mov     al, [ebx]
    test    al, al
    jnz     f_L2
f_end:
    mov     eax, esi
    mov     ebx, [ebp-12]
    mov     edi, [ebp-8]
    mov     esi, [ebp-4]
    leave
    ret
```

Escreva uma função equivalente a `f()` em C.

```
int f(const char * s, int (*action)(int, char), int v) {




}
```

3. **[6]** Implemente a função `int separate_signal(int[] *a, int n)` que separa os inteiros negativos dos inteiros positivos presentes em `a` colocando os negativos nos índices menores e os positivos nos índices maiores. O parâmetro `n` indica quantos inteiros estão presentes em `a`. Não poderá utilizar as funções da biblioteca standard do C.

```
int separate_signal(int[] *a, int n)  {









}
```