



Nome: \_\_\_\_\_ Número: \_\_\_\_\_ Turma: LI31N |A

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

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

```

.intel_syntax noprefix
.text
.globl f
f:
  mov    eax, [esp+8]
  sar    eax, 8
  mov    cl, [esp+6]
  movzx  ecx, cl
  mov    edx, eax
  sub    edx, ecx
  jl     L1
  inc    ecx
L1: xor   eax, ecx
  ret

```

eax	ecx

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

```
f(0x33445537, 0xFF7755AA);
```

2. [7] Considere a função `xpto()` implementada em C:

```

int xpto(short * s, int sz, void (*pf)(int*, int)) {
    int acc = 0;
    while (sz) {
        pf(&acc, *s);
        s++; sz--;
    }
    return acc;
}

```

Escreva uma função equivalente a `xpto()` em *assembly* IA32.

```

.intel_syntax noprefix
.text
.globl xpto
xpto:

```

3. [6] Implemente a função `int remove_char(char *s, char c)` que remove da sequência de caracteres apontada por `s` todas as ocorrências do carácter `c`. A função deverá actualizar o terminador de sequência de caracteres e retornar o número de caracteres removidos. Não poderá utilizar as funções da biblioteca standard do C.

```

int remove_char(char *s, char c) {

```