

# Prova Final de Software Básico — 2007.1

Duração: 1 hora e 45 minutos

28 de junho de 2007

**Atenção:** Seja preciso e objetivo em suas respostas. Clareza conta ponto. Não esqueça de colocar o seu nome e número de matrícula na folha de respostas.

**Boa sorte!**

## Questão 1 (3 pontos)

Considere o programa C a seguir:

```
void dump (void *p, int n) {
    unsigned char *p1 = p;
    while (n--) {
        printf("%p - %02x\n", p1, *p1);
        p1++;
    }
}

struct X {
    char c;
    float f;
    short s;
};

int main (void) {
    struct X a[2] = {{'f', -0.825, -772}, {'3', 1024.0, 127}};
    dump (a, 2*sizeof(struct X));
    return 0;
}
```

Diga o que este programa irá imprimir quando executado, *justificando os valores exibidos e mostrando as suas contas*. Suponha que a máquina de execução é *little endian* e que o vetor *a* começa no endereço 0x2A000000. Escreva 'LL' nos bytes que contiverem lixo. Lembre-se que o código ASCII de 'a' é 97 e de '0' é 48.

## Questão 2 (2 pontos)

Escreva em C ou em assembly, e *sem utilizar instruções de ponto flutuante*, uma função que receba como parâmetro um número em ponto flutuante em precisão simples (`float`), e retorne o valor real do expoente utilizado na representação IEEE para números de ponto flutuante. A função deve ter o seguinte protótipo:

```
int expoente(float f);
```

### Questão 3 (5 pontos)

Traduza as funções a seguir para Assembly da família de processadores Intel Pentium (IA-32), utilizando as regras usuais do C/Linux para passagem de parâmetros, de valor de retorno, e de salvamento de registradores na pilha. **Atenção:** Comente o seu código.

1. (3 pontos)

```
struct arv {
    int val;
    struct arv * fe;
    struct arv * fd;
};

typedef int (*f_map) (int);

void transformaNos (struct arv *a, f_map f) {
    if (a == NULL)
        return 0;
    else {
        transformaNos(a->fe, f);
        a->val = f(a->val);
        transformaNos(a->fd, f);
    }
}
```

2. (2 pontos)

```
double calc (double* p1, double* p2, int a) {
    double total;
    total = (p1[0] * p2[0]) + (p1[1] * p2[1]);
    return total / a;
}
```