

PUC-Rio – Software Básico – INF1018
Prova Final – 6/7/10

1. (2,5 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[2];
    double d;
    short s;
    float f;
};

int main (void) {
    struct X a = {-5 & 133, (1 << 4) | 3, 70.75, -2060, 0.125};

    dump (&a, 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 a variável *a* começa no endereço *0x2F000B08*. Escreva 'LL' nos bytes que contiverem lixo. Lembre-se que o código ASCII de 'a' é 97.

2. Traduza as funções *foo* e *bar* abaixo para assembly IA-32 (o assembly visto em sala), utilizando as regras usuais de alinhamento, passagem de parâmetros e resultados em C/Linux. Comente seu código.

- (a) (3 pontos)

```
struct X {
    int a;
    struct X *next;
};

int foo (struct X *x, int val) {
    int acc;
    while (x != NULL) {
        if ((x->a)==val)
            x->a = 0;
        acc += x->a;
        x = x->next;
    }
    return acc;
}
```

(b) (3 pontos)

```
float comp (float *f, int n);
double gee (double d);

float bar (double b[], int n) {
    int i;
    float a[20];
    for (i=0;i<n;i++)
        a[i] = gee(b[i]);
    return comp(a, n);
}
```

3. (1,5 pontos) Escreva uma função em C que receba um número inteiro (int) e o imprima em binário, sempre com 32 bits. (Use os operadores bit-a-bit de C!)