

PUC-Rio – Software Básico – INF1612
Prova Final – 11/12/03

1. (3,0 pontos) Considere o programa C a seguir:

```
#include <stdio.h>

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

struct um {
    char c;
    short s;
    float f;
};
struct outro {
    struct um *c1;
    struct um c2;
};
int main (void) {
    struct um a = {'b', -32, 1025.25};
    struct outro b = {&a, a};

    dump (&b, sizeof(b));
    return 0;
}
```

Considerando que **a** seja alocado na posição de memória **0xbffffcd0** e **b** seja alocado na memória logo a seguir (em endereços mais altos), diga o que esse programa irá imprimir quando executado, explicando como você chegou aos valores exibidos (mostre suas contas!!). O valor ascii de 'a' é 97 (0x61). Suponha que a máquina de execução é *little-endian*.

2. (4,0 pontos) Dado um número representando um ponto flutuante em precisão simples (float), garantidamente positivo e com expoente positivo, escreva em C ou assembly (nesse caso, seguindo as convenções de C de passagem de parâmetros e retorno de resultados), *sem utilizar instruções de ponto flutuante*, funções para:

- (a) retornar o *valor real* do expoente, nos bits menos significativos de um inteiro sem sinal:

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

- (b) retornar o campo mantissa nos bits menos significativos de um inteiro sem sinal:

```
unsigned int mantissa (float f);
```

- (c) retornar um inteiro correspondente ao valor do float truncado (jogando fora a parte fracionária do valor), ignorando a possibilidade de *overflow*:

```
unsigned int truncado (float f);
```

(as funções desenvolvidas nos itens anteriores podem ser usadas aqui)

dica: Se fizer em C, use o código abaixo para ter acesso à representação binária do número float:

```
unsigned int rep = *(unsigned int *)&f;
```

3. (3,0 pontos) Traduza a função `ff` abaixo para Assembly da família de processadores Intel Pentium, utilizando as regras usuais de alinhamento, passagem de parâmetros e resultados em C. Comente seu código.

```
typedef int (*func) (int);
void ff (int n, int *p, func f) {
    int l[n];
    int i;
    for (i=0;i<n;i++)
        l[i] = f(p[i]);
}
```