

PUC-Rio – Software Básico – INF1612
Prova Final – 28/6/05

1. (2,5 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 {
    double f;
    char c;
    int i;
    short s;
};
struct um a = {72.75, 9, -2150, -18};
int main (void) {
    dump (&a, sizeof(a));
    return 0;
}
```

Considerando que `a` seja alocado na posição de memória `0xbffffcd0`, diga o que esse programa irá imprimir quando executado, explicando como você chegou aos valores exibidos (*mostre suas contas!!*). Suponha que a máquina de execução é Pentium-Linux, ou seja, a representação de dados é a vista no curso.

2. Considere a representação IEEE de ponto flutuante vista no curso.
- (a) (1,5 pontos) Escreva em C, usando instruções de manipulação de bits, uma função que, dado um número representando um ponto flutuante em precisão simples, retorne, em um inteiro (em complemento a 2), o *valor real* do expoente:
- ```
int expoente (float f);
```
- (b) (1,0 ponto) Usando a função anterior, e sem utilizar comparações de ponto flutuante, escreva uma função C que, dado um número representando um ponto flutuante em precisão simples, retorne 1 se o módulo do número for menor que 1, e 0 caso contrário:
- ```
int abs_menor_que_um (float f);
```
3. Traduza as funções `bar` e `zee` abaixo para assembly IA-32 (o assembly visto em sala), utilizando as regras usuais de alinhamento, passagem de parâmetros e resultados em C. Comente seu código.

(a) (2,5 pontos)

```
int foo (int x);
int bar (int n) {
    int v;
    if (n<=1)
        return 0;
    else {
        v = foo(n);
        return 1 + bar(v);
    }
}
```

(b) (2,5 pontos)

```
struct X {
    int f;
    double val;
};
double zee (struct X *p, int n) {
    int i; double s = 0.0;
    for (i=0; i<n; i++)
        if (p[i].f)
            s = s + p[i].val;
    return s;
}
```