

**PUC-Rio – Software Básico – INF1018**  
**Prova Final – 11/12/2012**

**Atenção:** Não esqueça de colocar o seu nome e número de matrícula na(s) folha(s) de respostas. E leia o enunciado com cuidado!

1. (2,5 pontos) Considere o trecho de código C a seguir:

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

struct X {
    double d;
    short s;
    int i;
    char c[2];
};

int main (void) {
    struct X a = {-35.75, 521, -1030, {-10 & 1, ((18 << 2) | 3)}};

    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*, com as regras de alinhamento Linux/IA-32, e que a variável *a* começa no endereço 0x2A000000. Escreva 'LL' nos bytes de *padding*.

2. Construa uma função C que converte um array de inteiros sem sinal para uma representação *big endian*. A função recebe um array de inteiros, o tamanho desse array e um buffer de tamanho suficiente para armazenamento do resultado. A função deve preencher o buffer com a mesma sequência de inteiros do array original, porém com cada inteiro armazenado em *big endian*. Suponha, se necessário, que sua função vai executar em uma máquina *little-endian*.

```
void toBig (unsigned int origem[], unsigned char *dest, int tam)
```

3. Traduza as funções abaixo para assembly IA-32 do gcc/Linux (visto em sala), utilizando as regras básicas de alinhamento, passagem de parâmetros, retorno de resultado e uso de registradores em C. **Comente o seu código!**

(Não se preocupe se você não entender o que a função faz, apenas traduza-a literalmente).

- (a) (2,5 pontos)

```
struct S {
    double val1;
    int a;
    float val2;
};

double sum (struct S lista[], int tam) {
    double soma = 0.0;
    int i;
    for (i=0;i<tam;i++)
        soma += lista[i].val1 + lista[i].val2;
    return soma;
}
```

(b) (2,5 pontos)

```
struct Info {
    ''
}
struct X {
    int chave;
    struct Info *info;
    struct X *prox;
}

struct Info *foo(struct X *x, int c){
    if (x==NULL)
        return NULL;
    if (x->chave == c)
        return x->info;
    else
        return foo(x->next);
}
```