

PUC-Rio – Software Básico – INF1018
Prova 1 – Turma 3wa – 26/04/2016

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 X {
    int num1;
    struct X *next;
    char c1;
    short a;
};
int main (void) {
    struct X x1 = {34, NULL, 'a', -2050};
    struct X x2 = {-1036, &x1, 11<<3, 542};
    dump (&x2, sizeof(struct X));
    return 0;
}
```

Sabendo das informações abaixo (que podem ou não ser necessárias), e supondo que a máquina de execução é *little-endian* com as convenções de alinhamento do Linux no IA-64 (vistas em sala), diga o que esse programa irá imprimir quando executado. Coloque **PP** nas posições correspondentes a *padding*.

endereço de x1 na memória	0x7fff5f80bac0
endereço de x2 na memória	0x7fff5f80bae0
valor do caracter 'a' na tabela ASCII	97

ATENÇÃO: mostre como você chegou aos valores exibidos. Valores sem contas **NÃO** valem ponto!

2. Traduza as funções foo e boo abaixo para assembly IA-64 (o assembly visto em sala), utilizando as regras usuais de alinhamento, passagem de parâmetros, salvamento de registradores e resultados em C/linux. Traduza o mais diretamente possível o código de C para assembly.

Não se preocupe em entender o que as funções fazem, apenas traduza-as literalmente.

Comente seu código!

- (a) (2,2 pontos)

```
void foo (int *nums, int *indices, int qtos) {
    int i; int *pindices = indices;
    for (i=0; i<qtos; i++) {
        nums[*pindices] = 0;
        pindices++;
    }
}
```

(b) (2,3 pontos)

```
struct x {
    int v1;
    struct x *next;
};

int boo (struct x *px, int val) {
    if (px == NULL) return val;
    else {
        if (px->v1 > val) val = px->v1;
        return boo(px->next, val);
    }
}
```

3. (2,0 pontos) Escreva uma função `mostrastuct` que receba um ponteiro para um struct contendo campos de tipo `int` e `long` e mostra na tela os campos dessa struct, usando a base 10. A função deve ter o protótipo a seguir:

```
int mostrastuct (char* desc, void* v);
```

onde `v` é um ponteiro para a struct a ser exibida e `desc` contém uma descrição dos campos dessa struct. A string `desc` representa, na ordem, o tipo de cada campo da struct. O caracter 'i' indica um `int` (inteiro com sinal, 32 bits) e o caracter 'l' indica um `long` (inteiro com sinal, 64 bits). A função deve mostrar o valor de cada campo em uma linha separada.

Boa Prova!