

**PUC-Rio – Software Básico – INF1018**  
**Prova Final – 30/6/2016**

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 {
    char c;
    double d;
    int i;
    short s;
};
struct X x1 = {5<<3, -2.375, 4100, -19};

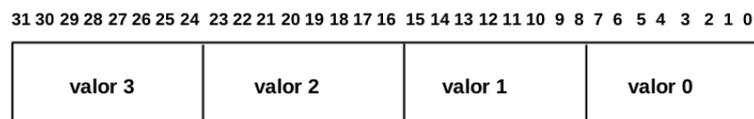
int main (void) {
    dump (&x1, sizeof(struct X));
    return 0;
}
```

Supondo que `x1` seja alocado na posição de memória `0x103d29018`, diga o que esse programa irá imprimir quando executado. Coloque **PP** nas posições correspondentes a *padding*.

Suponha que a máquina de execução é *little-endian* e que as convenções de alinhamento são as do Linux no IA-64 (vistas em sala).

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

2. (2,5 pontos) Suponha que determinada biblioteca usa uma representação compactada que armazena 4 valores inteiros de 8 bits, com sinal, em um `unsigned int`, conforme mostrado na figura a seguir.



Escreva em C uma função `incrementa` que receba um `unsigned int vals`, com a representação compactada descrita, e um inteiro `n` entre 0 e 3, que especifica um dos valores armazenados em `vals`. A função deve somar 1 ao valor especificado, sem alterar os demais valores, e retornar o `unsigned int` com a representação compactada resultante. No caso do resultado da soma não ser representável em 8 bits, o campo deverá ser zerado.

O protótipo da função é

```
unsigned int incrementa(unsigned int vals, int n);
```

3. Traduza as funções `foo` e `bar` 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) (2,5 pontos)

```
int f (int n);

void foo (int *a, int *b, int n) {
    int i;
    for (i = 0; i < n; i++) {
        b[*a] = f(i);
        a++;
    }
}
```

- (b) (2,5 pontos)

```
struct X {
    double d;
    float f;
};

double g (double a, double b);

double bar (struct X *p, int n, double val) {
    while (n--) {
        val += g(p->f, p->d);
        p++;
    }
    return val;
}
```

Boa Prova!