

PUC-Rio – Software Básico – INF1018  
Prova 1 – Turma 3WC – 08/05/2025

Nome:  
Turma:

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 Bravo {
    int i;
    short s;
    short *p;
    char c;
};
int main(void) {
    struct Bravo bb;
    bb.i = -20;
    bb.s = 31;
    bb.p = &bb.s;
    bb.c = 'e';
    dump(&bb, sizeof(struct Bravo));
    return 0;
}
```

Descreva o que este programa irá imprimir quando executado, *justificando* os valores exibidos mostrando as contas e outras informações usadas para chegar ao resultado. Suponha que a máquina de execução seja *little-endian* com as convenções de alinhamento do Linux no IA-64 vistas em sala. Além disto considere que `s` seja alocado na posição de memória `0x7ffc0131d9a0` e que o valor do carácter 'a' na tabela ASCII seja 97 (decimal). Coloque **PP** nas posições correspondentes a *padding*.

2. (2,0 pontos) Implemente uma função em C, que dado um inteiro sem sinal retorne o número de bits iguais a 0. Use o seguinte protótipo:

```
int numBits0(unsigned int u);
```

3. (5,0 pontos) Traduza a função `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. **Comente seu código!**

```
int valida(int val, int lim);

int boo(int lim, int *v, int n) {
    int i, soma = 0;
    for(i=0; i<n; i++) {
        if (valida(v[i], lim))
            soma += v[i];
    }
    return soma;
}
```

Boa Prova!