

PUC-Rio – Software Básico – INF1018
Prova 2 – Turma 3WA – 26/06/2018

1. (2,5 pontos)

```
#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;
    float f;
} x = {-33, 2043.25, -256.625};

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

Supondo que x seja armazenado no endereço de memória 0x601040, diga o que o programa irá imprimir quando executado, deixando claro como você chegou a esses valores. Considere que a máquina de execução é *little-endian*, e que as convenções de alinhamento são as do Linux no IA-64. Se houver posições de *padding*, indique seu conteúdo com **PP**. (ATENÇÃO: valores sem contas e explicações **NÃO** valem ponto!)

2. (1,5 pontos) Escreva o código assembly equivalente ao código de máquina que a função geracod do segundo trabalho de software básico geraria para o trecho de código *SB* a seguir:

```
v1 := $1
v2 := p1
v2 += v1
ret v2
```

3. Traduza as funções *prode* e *fooba* a seguir para assembly IA-64, utilizando as regras usuais de alinhamento, passagem de parâmetros, salvamento de registradores e resultados em C/Linux.

(a) (2,5 pontos)

```
double prode (float *u, float *v, int n) {
    float f = 0.0;
    while (n--) {
        f += *u * *v;
        u++;
        v++;
    }
    return f;
}
```

(b) (2,5 pontos)

```
#define TAM 3
struct X {
    double peso;
    double ab;
};

double norma (struct X *score, int p);

double fooba (double *u, int *v) {
    struct X cont[TAM];
    int i;
    int accp=0;

    for (i=0; i < TAM; i++) {
        cont[i].ab = *u;
        cont[i].peso = 1.0;
        accp += *v;
        u++; v++;
    }
    return norma(cont, accp);
}
```

4. (1,0 ponto) Considere o seguinte código em C:

```
#include<stdio.h>
int p = 0;
extern int bias;

int soma(int a, int b)
{
    static int acc = 0;
    return acc + a + b + bias;
}

static int dif(int a, int b)
{
    return a - b + bias;
}

int main(void) {
    bias = 1;

    printf("%d\n", soma(p,3));
    printf("%d\n", dif(p,3));

    return 0;
}
```

Liste quais símbolos do módulo objeto do código acima seriam importados e exportados na saída do programa `nm`, ou seja apareceriam como **D** (símbolo na área de dados, exportado), **T** (símbolo na área de código, exportado) e **U** (símbolo indefinido).

Boa Prova!