

PUC-Rio – Software Básico – INF1018
Prova 2 – Turma 3WA – 05/12/2017

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 Y {
float f1;
float f2;
char c1;
double d;
} y = {-0.3125, 0.0, -127, 15.750};

int main (void) {
    dump (&y, sizeof(struct Y));
    printf ("\n%ld\n", sizeof(y));
    printf ("\n%ld\n", sizeof(struct Y));
    return 0;
}
```

Supondo que y seja armazenado no endereço de memória 0x101110, 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. (2,0 ponto) Considere o seguinte código em C:

```
#include <stdio.h>

extern int a;
float b = 0.0;
int res(float u);

int f(int x, int y) {
    return (float) (x + y);
}

int main(void) {
    float t[3];
    int i;

    for (i = 0; i < 3; i++)
        t[i] = f(i, a);
        b += t[i];

    printf("%f\n", b);
    return res(b);
}
```

- (a) Ao compilarmos o código acima e examinarmos o módulo objeto gerado com o comando `nm`, liste quais símbolos aparecem como **T** (símbolo na área de código exportado) **D** (símbolo na área de dados exportado), **d** (símbolo na área de dados local), e **U** (símbolo indefinido).
- (b) Se alterarmos a declaração da variável `int i` por `static int i`, a saída do comando `nm` do item anterior é alterada? Justifique sua resposta.
3. Traduza as funções `fooba` e `boo` 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 g (double a, double s);

double fooba (double x, double p, int n) {
    int i;
    double res = 0.0;
    for (i=0; i<n; i++) {
        res += g (x, 1.0);
        x += p;
    }

    return res;
}
```

(b) (3,0 pontos)

```
#define TAM 20
struct x { char c; double d; };

double fecha (struct x *acc, int vol);

float boo (float com, float s) {
    int i;
    struct x acc[TAM];

    for (i=0; i<TAM; i++) {
        acc[i].c = i;
        acc[i].d = com;
        com += s;
    }
    return fecha(acc, TAM);
}
```

Boa Prova!