

PUC-Rio – Software Básico – INF1018
Prova 2 – Turma 3WA – 22/06/2017

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 X {
    float f1;
    double d1;
    char c;
} x = {-260.25, 1024.5, -7};

int main (void) {
    dump (&x, sizeof(struct X));
    printf ("\n%d\n", sizeof(x));
    printf ("\n%d\n", 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. (1,0 ponto) Considere o seguinte código em C:

```
#include <stdio.h>

extern double r;
int c = 0;
double sen(double v);

double baba(double *a, int n) {
    int i;
    for (i=0; i<n; i++) {
        r += sen(*a);
        a++;
    }
    return r;
}

int main(void) {
    double vec[] = {1.1, 2.2, 3.3};
    r = 0.0;
    printf("%f\n", baba(vec,3));
    c = getchar();
    return c;
}
```

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

3. (1,5 pontos) Escreva o código assembly equivalente ao código de máquina que a função compila do trabalho que você entregou geraria para o trecho de código *Simples* a seguir:

```
p1 = p1 + v1
v1 = v2 * $7
```

4. Traduza as funções `foo` 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. (Não se preocupe em entender o que as funções fazem, apenas traduza-as literalmente.)

- (a) (2,5 pontos)

```
float foo (double *v, double s, int n) {
    float f = 1.0;
    while (n--) {
        f += *v + s * n;
        v++;
    }
    return f;
}
```

- (b) (2,5 pontos)

```
#define MAX 5
double acerta (double* score, int tam);

double fooba (float *u, float *v, int tam) {
    double prod[MAX];
    int i = 0;
    for (i=0; i < tam; i++) {
        prod[i] = (double)*u * (double)*v;
        u++;
        v++;
    }
    return acerta(prod, tam);
}
```

Boa Prova!