PUC-Rio – Software Básico – INF1018
Prova Final – 04/07/2013

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 {
  int i;
  short s;
  float f;
} x = {2055, -4, -5.75};

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

Supondo que x seja armazenado no endereço de memória 0x804a018, mostre o que o programa irá imprimir quando executado. Considere que a máquina de execução é *little-endian*, e que as convenções de alinhamento são as do Linux no IA-32. Se houver posições de *padding*, indique seu conteúdo com **pp**. (ATENÇÃO: valores sem contas e explicações **NÃO** valem ponto!)

2. (1,5 pontos) Considere o arquivo C a seguir:

```
#include <math.h>

extern double min;
double max = INFINITY;
int tam = 32;
static double qa[32];

void quad(double *da, int n) {
  int i;
  for (i = 0; i < n; i++, da++) {
    if (*da > min && *da < max)
      qa[i] = pow(*da, 2.0);
    else
      qa[i] = 0;
  }
}
```

Supondo que este arquivo C seja compilado, gerando um arquivo objeto, liste quais símbolos apareceriam como **D**, **T** e **U** na saída do programa `nm` para esse arquivo objeto, considerando que, nesta saída, "D", "T" e "U" tem o significado a seguir:

- D - símbolo na área de dados, exportado
- T - símbolo na área de código, exportado
- U - símbolo indefinido

3. Traduza as funções `foo` e `boo` a seguir para assembly IA-32 (o assembly visto em sala), utilizando as regras usuais de alinhamento, **uso de registradores**, passagem de parâmetros e retorno de resultados em C. **Comente o seu código!**
(Não se preocupe se você não entender o que a função faz, apenas traduza-a literalmente).

   (a) (2,5 pontos)

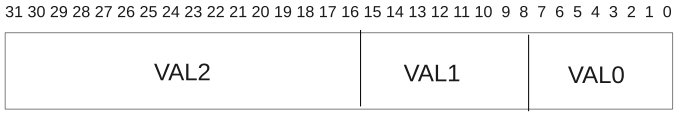
```
int foo(float *pf, int n) {
  double m = 1.0;
  int i;
  for (i = 0; i < n; i++) {
    m *= pf[i];
  }
  return m;
}
```

   (b) (2,5 pontos)

```
struct no {
  int val;
  struct no *fe;
  struct no *fd;
};

struct no *foo(struct no *a, int v) {
  if (a == NULL) return NULL;
  if (a->val == v) return a;
  if (a->val < v)
    return foo(a->fd, v);
  else
    return foo(a->fe, v);
}
```

4. (1,0 pontos) Considere um inteiro contendo os campos indicados abaixo:



   Escreva uma função C que receba um inteiro nesse formato e retorne a soma dos valores dos campos `VAL0`, `VAL1` e `VAL2`. A função deve ter o protótipo:

```
unsigned int somacampos(unsigned int u);
```