PUC-Rio – Software Básico – INF1018
Prova Final 09/12/2014

1. (2,5 pontos) Considere o programa C a seguir (as reticências na inicialização de **x** são propositais):

```
#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;
  int i;
  short s;
  float f;
};
int main(void) {
  struct X x = {...,...,...,...};
  dump(x, 20);
  return 0;
}
```

Imagine que, ao ser executado em uma máquina de arquitetura IA-32 (*little-endian*), obedecendo as convenções de alinhamento do Linux, esse programa imprimiu a saída a seguir:

```
0xffd1e340 - 23
0xffd1e341 - f2
0xffd1e342 - 78
0xffd1e343 - f7
0xffd1e344 - 03
0xffd1e345 - 10
0xffd1e346 - 00
0xffd1e347 - 00
0xffd1e348 - 01
0xffd1e349 - fe
0xffd1e34a - 04
0xffd1e34b - 08
0xffd1e34c - 00
0xffd1e34d - 00
0xffd1e34e - c0
0xffd1e34f - bf
0xffd1e350 - 70
0xffd1e351 - 84
0xffd1e352 - 04
0xffd1e353 - 08
```

Analisando a saída do programa e a declaração de `x` substitua as "reticências" na inicialização de `x` com os valores dos campos respectivos. Expresse esses valores em **notação decimal**. **Explique** como você chegou a esses valores (mostre suas contas e a posição dos valores na saída do dump). Valores sem explicação não serão considerados!

ATENÇÃO: O programa pede que o dump imprima 20 bytes. Esse número é suficiente para mostrar todo o conteúdo da estrutura, mas **podem ser exibidos bytes a mais!**

2. Traduza as funções `boo` e `foo` abaixo para assembly IA-32 do gcc/Linux (visto em sala), utilizando as regras usuais de alinhamento, passagem de parâmetros e retorno de resultado em C. Comente o seu código!

   (Não se preocupe se você não entender o que as funções fazem, apenas traduza-as literalmente).

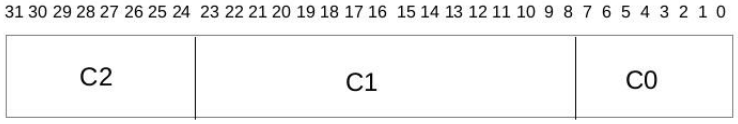
   (a) (3,0 pontos)

```
int bar(int i);

int boo(int *a, int n) {
  int i, x, y;
  x = 0;
  for (i = 0; i < n; i++) {
    y = bar(a[i]);
    if (y > 0)
      x += y;
  }
  return x;
}
```

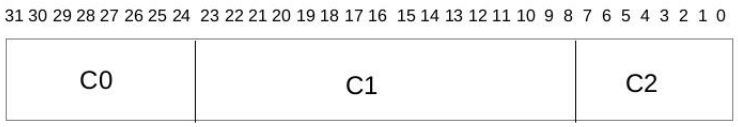
   (b) (3,0 pontos)

```
double sqrt(double x);

struct pt {
  float x;
  float y;
  struct pt *next;
};

double foo(struct pt *p) {
  double sum = 0.0;
  while (p != NULL) {
    sum += sqrt((p->x * p->x) + (p->y * p->y));
    p = p->next;
  }
  return sum;
}
```

3. (1,5 pontos) Considere um inteiro sem sinal contendo os campos indicados abaixo:



   Escreva uma função C que receba um inteiro sem sinal nesse formato e retorne um outro inteiro sem sinal, onde os campos `C0` e `C2` tenham suas posições trocadas em relação ao inteiro original. O inteiro a ser retornado deverá ter então o formato a seguir:



   O protótipo da função é

```
unsigned int invcampos(unsigned int u);
```