PUC-Rio – Software Básico – INF1018
Prova Final – 10/12/2015

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

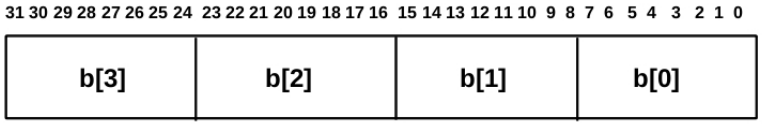
```
#include <stdio.h>
void dump (void *p, int n) {
  unsigned char *p1 = (unsigned char *) p;
  while (n--) {
    printf("%p - %02x\n", p1, *p1);
    p1++;
  }
}

struct X {
  char c;
  int i;
  float f;
  short s;
};

int main (void) {
  struct X x1 = {-3, 1031, -7.875,-36};
  dump (&x1, sizeof(struct X));
  return 0;
}
```

Supondo que `x1` seja alocado na posição de memória 0xbf84b290, diga o que esse programa irá imprimir quando executado. Coloque **PP** nas posições correspondentes a *padding*.

Suponha que a máquina de execução é *little-endian* e que as convenções de alinhamento são as do Linux no IA-32 (vistas em sala).

(ATENÇÃO: mostre como você chegou aos valores exibidos. Valores sem contas **NÃO** valem ponto!).

2. (2,5 pontos) Escreva uma função C que receba um array **b** de 4 bytes e retorne um inteiro de 32 bits que armazene esses quatro bytes da forma indicada a seguir:



O protótipo da função é

```
unsigned int junta(unsigned char b[]);
```

3. Traduza as funções `foo` e `bar` abaixo para assembly IA-32 (o assembly visto em sala), utilizando as regras usuais de alinhamento, passagem de parâmetros, salvamento de registradores e resultados em C/linux. Traduza o mais diretamente possível o código de C para assembly.

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

   Comente seu código!

   (a) (2,5 pontos)

```
int f (int i, int n);

void foo (int *a, int n) {
  int i = 0;
  for (i=0; i<n; i++) {
    a[f(i,n)] = i;
  }
}
```

   (b) (2,5 pontos)

```
struct X {
  double d;
  float f;
  struct X *next;
};

double g (double a, double b);

double bar (struct X *p) {
  double m = 1.0;
  while (p!=NULL) {
    m *= g(p->d, p->f);
    p = p->next;
  }
  return m;
}
```

Boa Prova!