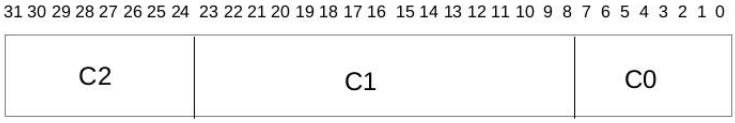
PUC-Rio – Software Básico – INF1018
Prova Final – 15/12/2016

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;
   double d;
   short s;
  } x = {'d', -4, 17.375, 1536};

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

Supondo que x seja armazenado no endereço de memória 0x905A6E0, 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*, que as convenções de alinhamento são as do Linux no IA-64 e que o valor do caracter 'a' na tabela ASCII é 97, em decimal. 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 seguinte formato para a composição de um inteiro sem sinal



onde `C0` corresponde aos bits de 0 a 7, `C1` aos bits de 8 a 23 e `C2` aos bits de 24 a 31.

Escreva uma função que receba dois inteiros sem sinal nesse formato (`x` e `y`) e um valor inteiro de 32 bits com sinal `val` e retorne um inteiro sem sinal, com o mesmo formato de `x` e `y`, que tenha o valor do campo `C0` igual ao valor correspondente de `x`, o valor do campo `C2` igual ao valor correspondente de `y` e o valor do campo `C1` igual ao valor recebido em `val`.

Caso o valor de `val` não possa ser representado nos 16 bits de `C1`, o valor deve ser truncado.

O protótipo da função é

```
unsigned int combina(unsigned int x, unsigned int y, int val);
```

3. Traduza as funções `foo` e `boo` abaixo para assembly IA-64 (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 cada função faz, apenas traduza-as literalmente.)

   (a) (2,5 pontos)

```
int foo(int *v, int n, int x) {
  int i;
  for (i = 0; i < n; i++) {
    if (v[i] == x)
      return i;
  }
  return -1;
}
```

   (b) (2,5 pontos)

```
struct X {
  double v1;
  int v2;
  struct X *prox;
};
int bar(int x);

double boo(struct X *px) {
  double acc1;
  int acc2;
  while (px) {
    acc1 += px->v1;
    acc2 += bar(px->v2);
    px = px->prox;
  }
  return acc1/(double)acc2;
}
```

4. (1,0 ponto) Utilizando a linguagem C, escreva uma função para testar se sua plataforma de execucão (processador) é uma arquitetura big-endian ou não. O protótipo dessa função deve ser

```
int isBigEndian(void);
```

   Esta função deve retornar verdadeiro (1) se a plataforma de execução for big-endian e falso (0) caso contrário.