PUC-Rio – Software Básico – INF1018
Prova 1 – Turma 3wb – 13/10/2016

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;
  short s;
  char *p;
};
char strings[2][5] = {"abcd","ABCD"};
int main (void) {
  struct X x = {'B' | 0x55, -1025, 37, &strings[1][0] };
  dump (&x, sizeof(struct X));
  return 0;
}
```

Sabendo das informações abaixo e supondo que a máquina de execução é *little-endian* com as convenções de alinhamento do Linux no IA-64 (vistas em sala), mostre o que esse programa irá imprimir quando executado. Coloque **PP** nas posições correspondentes a *padding*.

endereço de `strings` na memória	0x601040
endereço de `x` na memória	0x7fff33632020
valor do caracter 'A' na tabela ASCII	65 (decimal)

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

2. (2,0 pontos) No nosso primeiro trabalho, implementamos algumas operações básicas sobre inteiros *signed* de 128 bits, representando-os com a estrutura a seguir:

```
typedef struct int128 {
  long high; /* parte alta do valor:  bits 127 a 64 */
  long low ; /* parte baixa do valor: bits 63 a 0   */
} Int128;
```

Escreva agora uma função que verifique se um valor inteiro *signed* de 128 bits é positivo, isto é, se ele é **maior que zero**. A função deverá retornar zero para falso e um valor diferente de zero para verdadeiro. Seu protótipo é

```
int int128_pos (Int128 *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 as funções fazem, apenas traduza-as literalmente.)

   Comente seu código!

   (a) (2,5 pontos)

```
int foo (int a[], int b[], int n, int m) {
  if (n <= m)
    return a[n];
  else
    return b[m];
}
```

   (b) (3,0 pontos)

```
#define NULL 0
int f (char *s1, char *s2);

struct Y {
  char *st;
  struct Y *next;
};

int boo (struct Y *p1, struct Y *p2) {
  while ((p1 != NULL) && (p2 != NULL)) {
    if (f(p1->st, p2->st) > 1)
      return 1;
    p1 = p1->next;
    p2 = p2->next;
  }
  return 0;
}
```

Boa Prova!