PUC-Rio – Software Básico – INF1018
Prova 1 – 26/04/2012 – Turma 3WB

Atenção: Não esqueça de colocar o seu nome e número de matrícula na(s) folha(s) de respostas. Seja claro e objetivo. **Boa sorte!**

1. (2,5 pontos) No formato UTF-8, caracteres UNICODE tem representação de tamanho variável, podendo ocupar de 1 a 4 bytes. A tabela abaixo apresenta o número de bytes e a codificação utilizada para representar cada faixa de valores de códigos UNICODE:

Código UNICODE	Representação em UTF-8
U+0000 - U+007F	0xxxxxxx
U+0080 - U+07FF	110xxxxx 10xxxxxx
U+0800 - U+FFFF	1110xxxx 10xxxxxx 10xxxxxx
U+10000 - U+10FFFF	11110xxx 10xxxxxx 10xxxxxx 10xxxxxx

   Escreva, em C, uma função *numbytes34* com o protótipo:

```
int numbytes34(char *utf8);
```

   Essa função recebe uma string com uma sequência de caracteres UTF-8 e retorna o número total de **bytes** desta sequência que correspondem a símbolos das terceira e quarta faixas de valores (ou seja, simbolos na faixa U+800 a U+10FFFF).

   O final da string dada como entrada para a função é indicado por um byte nulo (0x00).

2. (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++;
  }
}

char cseq[] = "abc";

struct X {
  int x;
  char c[2];
  char *p;
  short s;
} x = {-1030, {'d', -5 & 0x77}, cseq, 2050 | 0xa5a5};

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

   Considerando que x seja armazenado no endereço de memória 0x80494e4, e *cseq* no endereço 0x0x80494e0, diga o que o programa irá imprimir quando executado, explicando como você chegou a esses valores (valores sem contas e explicações NÃO SERÃO CONSIDERADOS!). Indique com **pp** os bytes de *padding*.

   Considere que o valor do caracter 'a' na tabela ASCII é 97 (decimal), e suponha que a máquina de execução é *little-endian*, e que as convenções de alinhamento são as do Linux no IA-32.

3. Traduza as funções `f` e `g` abaixo para assembly IA-32 do gcc/Linux (visto em sala), utilizando as regras básicas de passagem de parâmetros, retorno de resultado e uso de registradores em C. Comente o seu código!

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

   (a) (2,5 pontos)

```
void f(int a[], int b[], int n) {
 for (i = 0; i < n; i++) {
   if (a[i] < 10)
     b[i] = a[i] + 10;
   else
     b[i] = a[i];
 }
}
```

   (b) (2,5 pontos)

```
#define NULL 0

struct N {
  int v1;
  int v2;
  char c;
  struct N *next;
};

struct N *find(struct N *n, int val) {
  if (n == NULL)
    return NULL;
  else {
    if ((n->v1 == val) || (n->v2 == val * 2)) {
      n->c = 1;
      return n;
    }
    else
      return find(n->next, val);
  }
}
```