

## Laboratório 04

### Revisão

1. Faça as seguintes conversões:

- (a)  $765_{10}$  para binário  
(b)  $1001111100110_2$  para hexadecimal  
(c)  $3462_{10}$  para hexadecimal  
(d)  $A73EF9B3_{16}$  para binário

2. Escreva uma função que retorne 1 se a memória tem organização little-endian e 0 caso contrário. A função deve ter a seguinte assinatura:

```
int is_little();
```

Escreva também uma função `main()` para testar sua função `is_little()`. Compile e execute o seu programa, e verifique se ele informa, corretamente, que a memória é *little endian*.

Dica: Você pode “quebrar” o sistema de tipos para inspecionar isoladamente os bytes de um valor inteiro de qualquer tamanho.

3. Escreva a função inversa da última função `num2string()`. Ela deverá converter um valor inteiro para uma string contendo um numeral que represente esse valor na base indicada.

A função `num2string()` deve receber um ponteiro para um buffer onde escreverá o resultado, o valor a ser convertido e a base para a conversão:

```
void num2string(char *s, int num, int base);
```

Para simplificar, assuma que o buffer tem tamanho suficiente para guardar todo o resultado.

4. Escreva a função *toggle()* que alterna um bit informado. A função tem a seguinte assinatura:

```
void toggle(int *value, int bit);
```

O parâmetro “bit” pode ser um valor entre 0 e 31, e indica qual bit de “value” deve ser alternado, ou seja, se o bit indicado for '0', ele deve ser mudado para '1', se for '1' deve ser mudado para '0'.

Os demais bits de “value” devem permanecer inalterados.

Os bits de “value” são numerados da seguinte forma:

31	30	29	28	27	26	25	24		23	22	21	20	19	18	17	16		15	14	13	12	11	10	9	8		7	6	5	4	3	2	1	0
?	?	?	?	?	?	?	?		?	?	?	?	?	?	?	?		?	?	?	?	?	?	?	?		?	?	?	?	?	?	?	?