

# Guia 2 - Sistemas de Numeração e Resentações de Dados

Luca Ribeiro Schettino Regne

<sup>1</sup>Instituto de Ciências Exatas e Informática - ICEI - (PUC-MG)

## 1. Exercício

### 1.1. Fazer as conversões de decimal para binário:

- a)  $27_{10} = 0000011011_2$
- b)  $51_{10} = 0000110011_2$
- c)  $713_{10} = 1011001001_2$
- d)  $312_{10} = 0100111000_2$
- e)  $360_{10} = 0101101000_2$

### 1.2. Escrever uma função dec2bin(x)

Esta função recebe um número inteiro decimal e devolve o binário correspondente. Faça um programa main que passa para a função os números decimais dos itens de "a" até "e" e que imprima os binários correspondentes na tela. Mostre o código e os resultados exibidos na tela, no relatório.

```
1 void dec2bin(int n) {  
2     for(int i = 31; i >= 0; i--) {  
3         if((n & (1 << i)) != 0) {  
4             printf("1");  
5         } else {  
6             printf("0");  
7         }  
8     }  
9     printf("\n");  
10 }
```

### 1.3. Fazer as conversões de binário para decimal:

- a)  $10101_2 = 21_{10}$
- b)  $11010_2 = 26_{10}$
- c)  $101001_2 = 41_{10}$
- d)  $111001_2 = 57_{10}$
- e)  $100011_2 = 35_{10}$

### 1.4. Escrever uma função bin2dec(x)

Esta função recebe um número binário e devolve o decimal correspondente. Faça um programa main que passa para a função os números binários dos itens de "a" até "e" e que imprima os binários correspondentes na tela. Mostre o código e os resultados exibidos na tela, no relatório.

```

1 void bin2dec(int n){
2     int num = n;
3     int dec = 0;
4
5     int base = 1;
6
7     int temp = num;
8
9     while (temp) {
10        int last_digit = temp % 10;
11        temp = temp / 10;
12        dec += last_digit * base;
13        base = base * 2;
14    }
15    printf("%d\n", dec);
16 }

```

### 1.5. Fazer as conversões de decimal para hexadecimal:

- a)  $73_{10} = 49_{16}$
- b)  $47_{10} = 2F_{16}$
- c)  $61_{10} = 3D_{16}$
- d)  $157_{10} = 9D_{16}$
- e)  $171_{10} = AB_{16}$

### 1.6. Escrever uma função dec2hex(x)

Esta função recebe um número inteiro decimal e devolve o hexadecimal correspondente. Faça um programa main que passa para a função os números decimais dos itens de "a" até "e" e que imprima os hexadecimais correspondentes na tela. Mostre o código e os resultados exibidos na tela, no relatório.

```

1 int dec2hex(long dec){
2     long quotient, tmp;
3     int i, j = 0;
4
5     char hex[100];
6
7     quotient = dec;
8
9     while (quotient != 0){
10        tmp = quotient % 16;
11        if (tmp < 10)
12            hex[j++] = 48 + tmp;
13        else
14            hex[j++] = 55 + tmp;
15        quotient = quotient / 16;
16    }
17
18    printf("%s\n", hex);
19 }

```

### 1.7. Fazer as conversões de hexadecimal para decimal:

- a)  $73_{16} = 115_{10}$
- b)  $ABC_{16} = 2748_{10}$
- c)  $100_{16} = 256_{10}$
- d)  $9A8_{16} = 2472_{10}$
- e)  $1000_{16} = 4096_{10}$

### 1.8. Escrever uma função hex2dec(x).

Esta função recebe um número hexadecimal e devolve o decimal correspondente. Faça um programa main que passa para a função os números hexadecimais dos itens de "a" até "e" e que imprima os decimais correspondentes na tela. Mostre o código e os resultados exibidos na tela, no relatório.

```
1 void hex2dec(char *hex) {
2     long long decimal = 0, base = 1;
3     int value, length = strlen(hex);
4
5     for(int i = length--; i >= 0; i--)
6     {
7         if(hex[i] >= '0' && hex[i] <= '9')
8         {
9             decimal += (hex[i] - 48) * base;
10            base *= 16;
11        }
12        else if(hex[i] >= 'A' && hex[i] <= 'F')
13        {
14            decimal += (hex[i] - 55) * base;
15            base *= 16;
16        }
17        else if(hex[i] >= 'a' && hex[i] <= 'f')
18        {
19            decimal += (hex[i] - 87) * base;
20            base *= 16;
21        }
22    }
23    printf("%lld\n", decimal);
24 }
```

### 1.9. Converter entre símbolos e códigos de representação alfanumérico (ASCII).

**Sugestão: veja a codificação da tabela ASCII acima.**

- a) "PUC – Minas" =  $5055432d4d696e6173_{16_{ASCII}}$
- b) "2021 – 1" =  $323032312d31_{16_{ASCII}}$
- c) "Brasil" =  $42726173696c_{16_{ASCII}}$
- d)  $124101122104105_{16} = A!_{ASCII}$
- e)  $622E682E2D6D67_{16} = b.h. - mg_{ASCII}$

### 1.10. Escrever a função ASCII2hex(x).

Esta função recebe um texto ASCII e devolve os caracteres hexadecimais correspondentes. Escrever a função hex2ASCII(xx). Esta função recebem caracteres hexadecimais e de-

volve o texto ASCII correspondentes. Faça um programa main que teste as funções acima. Mostre o código e os resultados exibidos na tela, no relatório

```
1 void ASCII2hex(char* ascii){
2     char *hex = malloc(((sizeof ascii) * 2) + 1);
3     int l = strlen(ascii);
4
5     for(i = 0; i < len; i++){
6         sprintf(hex+i*2, "%02X", ascii[i]);
7     }
8     printf("%s\n", hex);
9 }
```