Faculdade de Computação (FACOM) - UFMS
Algoritmos e Programação 2
Prof. Dr. Anderson Bessa da Costa

Lista 9 - Algoritmos Recursivos Parte 2

1. Considere o seguinte código que imprime os números de 1 até 10 na tela, em ordem decrescente:

```
1 void imprime_10_1() {
2   int i;
3
4   for(i = 10; i > 0; i++)
5     printf("%d ");
6 }
```

   Implemente uma versão recursiva para esse algoritmo.

2. Modele a recorrência (como vimos em sala) e implemente uma solução do seguinte problema: Imagine v como um vetor de inteiros. Calcule o produto dos elementos do vetor.

3. Reescreva a função abaixo tornando-a recursiva. Esta função conta o número de algarismos (dígitos) que um número inteiro n possui. Assuma $n > 0$.

```
1 int digitos (int n) {
2     int cont = 1;
3     while (n > 9) {
4         n = n / 10;
5         cont += 1;
6     }
7
8     return cont;
9 }
```

4. Escreva um método recursivo tal que para um inteiro positivo n imprima números ímpares:

   (a) entre 1 e n
   (b) entre n e 1

5. Escreva uma função recursiva para adicionar os primeiros n termos da série abaixo:

$$1 + \frac{1}{2} - \frac{1}{3} + \frac{1}{4} - \frac{1}{5}, ... \tag{1}$$

6. Palíndromo é uma palavra, frase ou número que permanece igual quando lida de trás para frente. Para este exercício, iremos considerar como entrada apenas *strings*. Assim, "arara", "121", "11", "ovo" e "renner" são exemplos de palíndromos.

   (a) Complete a definição recursiva abaixo, para determinar se um dado vetor v é palíndromo. A sua recorrência deve retornar 1 se o vetor for palíndromo e 0 caso contrário. Dica: Você pode usar quantos e quais parâmetros desejar, apenas seja razoável.

$$pali(v[..], \qquad\qquad) = \begin{cases} \\ \\ \end{cases}$$

(b) Com base na sua definição recursiva, implemente uma função recursiva em C/C++ que verifica se um vetor fornecido é palíndromo.

7. Escreva a definição recursiva do seguinte problema: computar qualquer número a potencia de qualquer número não negativo. Identifique o(s) caso(s) base(s) e a(s) chamada(s) recursivas(s).

8. Escreva recursivamente e não recursivamente funções para imprimir um inteiro não negativo em binário. As funções não usam operações de bit.

9. Utilizando o valor de e como 2.71828..., a base para a função do logaritmo natural, nós temos a seguinte somatória para estimar o valor de e elevado a qualquer valor real x,

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + ...$$

Implemente uma função recursiva chamada `e_exp` para dado um valor real x e um valor n, estime o valor de e^x calculando a somatória da série para os n primeiros termos. Você precisará implementar a função `fatorial(n)` que retorna o cálculo de $n!$. Para calcular a potência de um número $base^{expoente}$, faça uso da função `double pow(double base, double expoente)` da biblioteca `math.h`.