

Lista de Exercícios 5 - Recursividade

Bacharelado em Ciência da Computação

Linguagens de Programação II

Professor: Dany Sanchez Dominguez

1. Escreva uma função que recebe como parâmetro um inteiro positivo N e retorna a soma de todos os números inteiros entre 0 e N

a) Implemente uma solução iterativa

$$S(N) = \sum_{i=1}^N i$$

b) Implemente uma solução recursiva

$$S(N) = \begin{cases} 1 & \text{para } N = 1 \\ N + S(N - 1) & \text{para } N > 1 \end{cases}$$

2. Dada a seguinte função iterativa

```
int Digitos(int N){
    int cont =1;

    while (abs(N )>9){
        N = N / 10;
        cont++;
    }

    return cont;
}
```

- a) Que a função faz?
 - b) Escreva uma definição recursiva para o mesmo processo.
 - c) Implemente uma versão recursiva da função.
3. Em sala de aulas foi apresentada uma função recursiva `potencia_rec()` que calculava a potência a^n para valores inteiros e positivos de n . Aprimore a função `potencia_rec()` para trabalhar com expoentes inteiros positivos e negativos.
 4. Utilizando as funções iterativas e recursivas discutidas em sala de aulas para computar o termo n da sequência de Fibonacci. Conte o número de adições necessárias para computar `fib(n)` para $0 \leq n \leq 10$ por meio dos métodos recursivo e iterativo. Existe algum tipo de padrão? Qual implementação é mais eficiente. Justifique.

5. Implemente um programa que leia dois números a e b , sendo ($a < b$), imprima todos os números da sequência contidos no intervalo $[a, b]$. Modifique seu programa para efetuar o mesmo procedimento utilizando uma função recursiva.

6. Dada a implementação abaixo

```
int F(int N){
    if (N < 4)
        return 3 * N;
    else
        return 2 * F(N - 4) + 5;
}
```

- a) Quais são os valores de $F(3)$, $F(7)$ e $F(12)$?
- b) Implemente uma versão iterativa da função anterior.
7. Implemente recursivamente uma função `Max_rec()` que retorne o maior valor armazenado em um vetor V , contendo n números inteiros.
8. Dada a função:

```
int F(int n){
    if ((n >= 0) && (n <= 2))
        return n;
    else
        return F(n-1)+F(n-2)+F(n-3);
}
```

- a) Qual é a resposta do algoritmo para a chamada $F(6)$?
- b) Quantas chamadas recursivas são feitas para $F(6)$?
9. Escreva uma função recursiva, `ImprimeSerie(int i, int j, int k)`, que imprime na tela a série de valores do intervalo $[i, j]$, com incremento k .
10. Escreva uma função recursiva que receba um inteiro positivo n e devolva a soma dos dígitos decimais de n . Por exemplo, ao receber 1729 sua função deve devolver 19.
11. Escreva uma função recursiva que determine o valor do n -ésimo elemento de uma sequência geométrica definida pelos termos: $a, ar, ar^2, ar^3, \dots, ar^{n-1}$. Os argumentos da função são o primeiro termo (a), a razão (r) e o inteiro n .
12. Faça uma função recursiva, em linguagem C, que calcule o valor da série S descrita a seguir para um valor $n > 0$ a ser fornecido como parâmetro para a mesma

$$S = 2 + \frac{5}{2} + \frac{10}{3} + \dots + \frac{1 + n^2}{n}$$

13. Em linguagem C a expressão $m\%n$, resulta o resto da divisão inteira entre m e n . Definimos o máximo divisor comum (MDC) de dois inteiros x e y na forma

$$mdc(x, y) = y \quad \text{se } (y \leq x \ \&\& \ x\%y = 0)$$

$$mdc(x, y) = mdc(y, x) \quad \text{se } (x < y)$$

$$mdc(x, y) = mdc(y, x\%y) \quad \text{caso contrário}$$

Escreva uma função recursiva e uma iterativa para calcular $mdc(x, y)$.

14. Escreva uma função recursiva que receba um número n e imprima os divisores do número.
15. Um número é perfeito se ele for igual a soma de seus divisores. Por exemplo, $6 = 1 + 2 + 3$, então 6 é perfeito. Escreva um programa que receba um número inteiro maior que zero e imprima se o número é perfeito. Utilize uma função recursiva para determinar se o número é perfeito.
16. Um algoritmo para calcular a raiz cúbica de um número x é definido pelos seguintes passos:
- (a) Comece considerando um limite inferior e outro superior para a raiz. Se o número for maior ou igual que 1, utilize 1 como limite inferior e x como limite superior. Caso contrário, use x como limite inferior e 1 como limite superior.
 - (b) Se a diferença entre os limites inferior e superior for menor que um certo valor de precisão (por exemplo 0.00001) o valor da raiz é a média aritmética desses limites e encerramos o cálculo.
 - (c) Se a diferença entre os limites é maior que a precisão, verifique se a média dos limites inferior e superior é maior ou menor que a raiz cúbica de x , para isso, eleve a média ao cubo e compare com x . Se a média for menor do que a raiz cúbica x substitua o limite inferior e retorne ao passo anterior (o limite superior permanece inalterado). Se a média for maior do que a raiz cúbica x substitua o limite superior e retorne ao passo anterior (o limite inferior permanece inalterado).

Usando o algoritmo descrito acima construa um programa que receba o valor de x e calcule a raiz cúbica. Utilize uma função para calcular a raiz que receba o valor de x e os limites inferior e superior e retorna a raiz. Para a precisão defina uma constante simbólica. Implemente uma solução recursiva e outra iterativa para o problema.

17. Escreva uma função recursiva que receba uma string e retorne a quantidade de caracteres maiúsculos na string.
18. Crie uma função recursiva em C que receba por parâmetro uma string e que deve devolver a mesma string invertida. Por exemplo, a string CARRO será transformada em ORRAC. Nota: NÃO é para criar uma nova string. É para alterar diretamente a string passada por parâmetro.
19. Escreva uma função recursiva que receba uma string e retorne verdadeiro se a string for um palíndromo ou falso caso contrário.

20. Considere a função **Comb**(**n**, **k**), que representa o número de grupos distintos com k pessoas que podem ser formados a partir de n pessoas. Por exemplo, $\text{Comb}(4, 3) = 4$, pois com 4 pessoas (A, B, C, D), é possível formar 4 diferentes grupos: ABC, ABD, ACD e BCD. Sabe-se que:

$$\text{Comb}(n, k) = \begin{cases} 1 & \text{se } k = 1 \\ 1 & \text{se } k = n \\ \text{Comb}(n-1, k-1) + \text{Comb}(n-1, k) & \text{se } 1 < k < n \end{cases}$$

- a) Implemente uma função recursiva para **Comb**(**n**, **k**)
 - b) Mostre a árvore de recursão para **Comb**(5, 3)
 - c) Sabendo-se ainda $\text{Comb}(n, k) = n!/(k!(n-k)!)$, implemente uma função não recursiva para **Comb**(**n**, **k**).
21. Dada a definição da função de Ackerman:

$$A(m, n) = \begin{cases} n + 1 & \text{se } m = 0 \\ A(n-1, 1) & \text{se } m > 0 \text{ e } n = 0 \\ A(m-1, A(m, n-1)) & \text{se } m > 0 \text{ e } n > 0 \end{cases}$$

válida para valores inteiros não negativos de m e n , implemente uma versão recursiva do algoritmo e faça o diagrama de execução de $A(1, 2)$.

22. **Combinações do Alfabeto** Crie um programa que receba um número inteiro positivo N e escreva na tela, todas as possibilidades de combinações das N primeiras letras do alfabeto. Por exemplo, se o número passado por parâmetro for 3, a saída deverá ser:

ABC
ACB
BAC
BCA
CAB
CBA

Seu programa deve utilizar uma função recursiva.