

Lista de Exercícios

Recursividade

Lembre-se => Sempre que for programar funções recursivas, tenha em mente duas coisas:

1. Caso base (ou condição de parada)
2. Passos recursivos (chamadas da função)

1. Implemente uma função recursiva que, dados dois números inteiros x e n , calcula o valor de x^n .

- Caso base: $X^0 = 1$
- Passo da recursão: $x^n = x * x^{n-1}$

2. Usando recursividade, calcule a soma de todos os valores de um vetor de reais.

- Caso base? Tamanho do array = 0. Soma é 0.
- $v[n-1]$ + soma do restante do array.

3. Dado um array de inteiros e o seu número de elementos, inverta a posição dos seus elementos.

- Caso base? Tamanho do array menor ou igual a 1.
- Troca 1o. e último elementos e inverte resto do array.

4. Escreva uma função recursiva que determine quantas vezes um dígito K ocorre em um número natural N . Por exemplo, o dígito 2 ocorre 3 vezes em 762021192.

- Caso base? Quando todos os dígitos já foram examinados, ou seja, $N = 0$
- Passo da recursão: $n_4n_3n_2n_1n_0$
 - (0 ou 1) + número de ocorrências em $N / 10$ ($n_4n_3n_2n_1$)

5) O máximo divisor comum (MDC) de dois números inteiros x e y pode ser calculado usando-se uma definição recursiva:

$MDC(x, y) = MDC(x - y, y)$, se $x > y$

$MDC(x, y) = MDC(y, x)$

$MDC(x, x) = x$

- Caso base? $x == y$
- Passo da recursão? Os dois outros casos acima. O problema é definido recursivamente.

Soluções

1.

```
int pot(int b, int p){  
    if ( p == 0 ) return 1;  
  
    return ( b * pot(b, p -1));  
}
```

2. int soma_a(int v[], int n){

```
    if ( n == 0 ) return 0;
```

```
    return v[n -1] + soma(v, n - 1) ;
```

3. void inverte(int v[], int esq, int dir){

```
int t;
```

```
    if (esq >= dir)        return;
```

```
    t = v[esq];
```

```
    v[esq] = v[dir];
```

```
    v[dir] = t;
```

```
    inverte(v, esq+1, dir - 1);
```

```
}
```

4. int conta_dig(int N, int K){

```
    if ( N == 0 ) return 0;
```

```
    return conta_dig( N / 10 , K) + ( N % 10 == K );
```

```
}
```

5. int mdc(int p, int q){

```
    if ( p == q ) return p;
```

```
    if ( p < q ) return mdc(q, p);
```

```
    return mdc(p - q, q);
```

```
}
```