

Estrutura de Dados 1

Exercícios

Prof. Paulo Henrique Ribeiro Gabriel

Sugestão: Resolva os exercícios a seguir *sem* utilizar o computador. Faça um esquema (pseudocódigo) da solução, escreva um código-fonte e, finalmente, implemente-o em C. As soluções são apresentadas no final desse documento; evite consultá-las antes de concluir os exercícios.

1. Implemente a multiplicação (recursiva) de inteiros utilizando somas sucessivas.

Dica: lembre-se de que a multiplicação de a por b é equivalente a b somas de a . Por exemplo, se $a = 4$ e $b = 3$, temos:

$$4 \times 3 = 4 + (4 \times 2) = 4 + (4 + (4 \times 1)) = 4 + 4 + 4 = 12$$

2. Seguindo a ideia descrita anteriormente, apresente um esquema recursivo para dividir dois número inteiros utilizando contagens sucessivas.

Dica: Sabemos que dividir a por b é o mesmo que descobrir quantas vezes b cabe em a . Por exemplo, para dividir 18 por 4 podemos contar o número de vezes que 4 cabe dentro de 18.

3. Escreva uma função recursiva para somar os elementos de um *array*.
4. Escreva uma função recursiva para encontrar o maior elemento de um *array* de inteiros.

Dica: O maior elemento de um *array* é o maior valor entre o último e o restante. Quando houver apenas um elemento, este é o maior.

Soluções

Exercício 1:

```
int multiplica(int a, int b)
{
    // 1 eh o elemento neutro da multiplicacao
    if (b == 1) return b;
    else return a + multiplica(a, b-1);
}
```

Exercício 2:

```
int divide(int a, int b)
{
    if (b == 1) return a;
    else if (a < b) return 0;
    else return 1 + divide(a-b, b);
}
```

Exercício 3:

```
int soma_elementos(int vet[], int n)
{
    if (n == 0) return vet[n];
    else return (vet[n]+ soma_elementos(vet, n-1));
}
```

Exercício 4:

```
int max_array(int vet[], int n)
{
    int x;
    if (n == 1) return vet[0];
    else {
        x = max_array(vet, n-1); // maximo de v[0..n-2]
        if (x > v[n-1]) return x;
        else return v[n-1];
    }
}
```