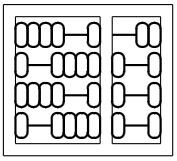
MC 102 - Algoritmos e programação de computadores
Instituto de Computação — UNICAMP

Sétima lista de exercícios



1. Escreva uma função recursiva que calcule a diferença entre o valor de um elemento máximo e o valor de um elemento mínimo de uma vetor $v[0 \ldots n-1]$.

2. Escreva uma função recursiva que calcule a soma dos dígitos decimais de um inteiro positivo. Por exemplo, a soma dos dígitos de 132 é 6.

3. Escreva uma função recursiva que calcule $\lfloor \lg n \rfloor$, ou seja, o *piso* do logaritmo de n na base 2.

4. Execute a funçào ff abaixo com os argumentos 7 e 0.

```
int ff(int n, int ind) {
    int i;
    for (i = 0; i < ind; i++)
        printf(" ");
    printf ("ff (%d, %d)\n", n, ind);
    if (n == 1)
      return 1;
    if (n % 2==0)
      return ff(n/2, ind + 1);
    return ff((n-1)/2, ind +1) + ff((n+1)/2, ind +1);
}
```

5. Escreva uma função recursiva que implemente o algoritmo de Euclides.

6. Escreva uma função recursiva que determine se os elementos de um vetor formam um palíndromo. Por exemplo, se $v = [1, 20, 30, 20, 1]$ v é um palíndromo; se $v = [20, 30, 31, 20]$ não é um palíndromo.

7. Escreva uma função que implemente a busca binária recursivamente.

8. Codifique o algoritmo *mergesort*.

9. Codifique o algoritmo *quicksort*.

10. Dados dois inteiros positivos m e n, escreva duas funções recursivas, $quociente(m, n)$ e $resto(m, n)$, que retornem, respectivamente, o quociente e o resto de m por n. Suas funções não podem usar multiplicação ou divisão.

11. Escreva uma função que receba como entrada um número n e imprima todas as possíveis combinações de elementos pertencentes ao conjunto $\{1, \ldots, n\}$. Isto é, se $n = 3$ a saída deveria ser:

$$-1 - 12 - 123 - 13 - 2 - 23 - 3.$$

    Note que a primeira combinação é vazia.

12. Escreva uma função que receba como entrada dois inteiros positivos m e n tais que $m \leq n$ e imprima todas as combinações do conjunto $\{1, \ldots, n\}$ que possuam tamanho n. Por exemplo: se $m = 3$ e $n = 5$ o resultado deveria ser

$$123 - 124 - 125 - 134 - 135 - 145 - 234 - 235 - 245 - 345.$$

13. Escreva uma função recursiva que converte um inteiro na base decimal para a base binária.

14. Considere o seguinte problema, que é uma variante do problema original das torres de hanoi: existem $2n$ discos de tamanhos $\{1, 2, \ldots, 2n-1, 2n\}$; há três torres A, B, C; na torre A estão os discos de tamanhos ímpares em ordem crescente; na torre C estão os discos de tamanhos pares em ordem crescente; o seu objetivo é mover os discos da torre A para a torre C e os discos da torre C para a torre A, respeitando as mesmas regras do problema original das torres de hanoi.

15. Resolva o problema das torres de hanoi considerando que não é permitido que um disco seja movido diretamente da torre A para a torre C.

16. Execute a função a seguir sem usar um computador, considerando $n = 50$.

```
int f(int n) {
    if (n > 100) return n-10;
    else return f(f(n+11));
}
```

17. Determine o que as funções a seguir fazem.

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

```
int f(int a, int b) {
    int m;
    if (a != b) {
        m = (a+b) /2 ;
        f(a, m);
        printf("%d\n", m);
        f(b, m)
    }
}
```