Faculdade de Computação - UFPA
Disciplina: Programação 1

Tópico: recursão

1) Faça uma **função recursiva** que receba dois números inteiros positivos K e N e calcule K^N. Utilize apenas multiplicações.

2) Faça uma **função recursiva** que receba um número inteiro positivo N e calcule o somatório dos números de 1 a N.

3) Faça uma **função recursiva** que permita somar os elementos de um vetor de inteiros.

4) Mostre a pilha de chamadas recursivas para o exercício 3.

5) O máximo divisor comum dos inteiros x e y é o maior inteiro que é divisível por x e y. Escreva uma **função recursiva** que retorna o máximo divisor comum de x e y. O mdc de x e y é definido como segue: se y é igual a 0, então mdc (x, y) é x; caso contrário, mdc (x, y) é mdc (y, x%y), onde % é o operador de resto.

6) Mostre a pilha de chamadas recursivas para o exercício 5.

7) Multiplicação de números naturais. Outro exemplo de uma definição recursiva é a definição da multiplicação de números naturais. O produto a * b, em que a e b são inteiros positivos, pode ser definido como a somado a si mesmo b vezes. Essa é uma definição iterativa. Uma definição recursiva equivalente é:
a * b = a se b == 1
a * b = a * (b - 1) + a se b > 1
Implemente uma **função recursiva** para multiplicar dois números naturais sem utilizar o operador de multiplicação (*).

8) A sequência de Ricci é uma sequência bastante semelhante à de Fibonacci, diferindo desta apenas pelo fato que os dois primeiros termos da sequência (F(0) e F(1)) devem ser definidos pelo usuário.

Sabendo-se que a sequência de Fibonacci é definida por:

- F(0) = 0
- F(1) = 1
- F(n) = F(n - 1) + F(n - 2) , $n \geq 2$

Crie um programa que imprima os n primeiros termos da sequência de Ricci, utilizando um **subprograma** recursivo que retorna o n-ésimo termo da referida série.

Entrada:
1. Os valores iniciais da série de Ricci (F(0) e F(1));
2. Os número de termos dessa sequência a serem impressos.

Saída:
1. Os n termos dessa sequência.

Exemplo de entrada:
5 8
6

Exemplo de saída:
5 8 13 21 34 55

9) A função de Ackermann-Péter é uma função importante em Teoria da Computação, que é definida assim:

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

Onde *m* e *n* são números inteiros não negativos. Cuidado, esta função requer quantidades enormes de processamento conforme seus parâmetros aumentam. Evite usar parâmetros cuja soma é maior que 7.

Faça um programa que tem uma função para calcular esta função. As operações de entrada e saída devem ser feitas fora da função recursiva.

Entradas:

- Dois valores inteiros não negativos que determinam respectivamente *m* e *n* na definição acima, numa mesma linha.

Saídas:

- O valor da função para os parâmetros lidos.

Exemplo de Entrada:

2 4

Exemplo de Saída:

11

Aviso: *Não utilize valores grandes para testar a aplicação desenvolvida, o programa pode demorar muito para efetuar os cálculos!*.