

Algoritmos e Estruturas de Dados I (DCC/003)

Aula Prática 10 (recursividade)

Instruções:

- Os exercícios deverão ser feitos em aula de laboratório durante o tempo da aula; quem não conseguir terminar terá ainda um prazo extra para entrega.
- professor irá esclarecer dúvidas em aula;
- Crie uma pasta com seu nome em um diretório temporário e vá gravando seus programas implementados.

Exercício 1 (Salve o código como **ap10-ex1.c**): Faça um programa **recursivo** que gere os valores de uma progressão geométrica. O usuário deverá entrar com os valores do termo inicial, a razão da PG, e finalmente o número de termos a serem gerados. Neste caso será um procedimento (void) que vai imprimindo cada termo e continuando a recursão.

Exercício 2 (Salve o código como **ap10-ex2.c**): Faça um programa **recursivo** que calcule a série de Ackerman, que é definida da seguinte forma:

$$c_{n+1} = \begin{cases} \frac{c_n}{2}, & \text{se } c_n \text{ é par} \\ 3c_n + 1, & \text{caso contrário} \end{cases}$$

O seu programa deve receber o valor (um valor positivo maior do que zero) do primeiro termo, e gerar termos até que seja encontrado o valor 1.

Exercício 3 (Salve o código como **ap10-ex3.c**): Um problema típico em ciência da computação consiste em converter um número da sua forma decimal para a forma binária. Por exemplo, o número 12 tem a sua representação binária igual a 1100. A forma mais simples de fazer isso é dividir o número sucessivamente por 2, onde o resto da i-ésima divisão vai ser o dígito i do número binário (da direita para a esquerda). Por exemplo: $12 / 2 = 6$, resto 0 (1º dígito da direita para esquerda), $6 / 2 = 3$, resto 0 (2º dígito da direita para esquerda), $3 / 2 = 1$ resto 1 (3º dígito da direita para esquerda), $1 / 2 = 0$ resto 1 (4º dígito da direita para esquerda). Resultado: $12 = 1100$.

Escreva uma **procedimento recursivo** *void calculaBinario (int x)* que, dado um número decimal x, imprima a sua representação binária corretamente.