

INFORMAÇÕES SOBRE ESTA ATIVIDADE:

- Esta atividade valerá 2,0 pontos (*1,0 ponto de trabalho mais 1,0 ponto de AAI*).
- A atividade é complementar e ser entregue pelo SINEF **até dia 24/set/2012**.
- Deve ser entregue um arquivo compactado chamado **Ativ03b.zip** com um arquivo chamado **grupo.txt** contendo o nome dos integrantes e os arquivos **c** cujos nomes devem seguir o seguinte padrão: **Ativ03bEx01.c, Ativ03bEx02.c, ... Isto será avaliado!**
- Apenas **um** do grupo deverá entregar. **Isto também será avaliado!**
- Atividade não serão recebidas/pontuadas **após o prazo de entrega**. Portanto, fique atento ao prazo.
- **Cópias** (total ou parcial) serão penalizadas com **nota zero**.
- A atividade é em grupo de, no máximo, **dois** alunos.

Atividade 03b - Lista de Exercícios Complementar – Funções Recursivas

Dica: É boa prática a utilização de protótipos.

1. Crie um programa C ANSI que receba um valor **n** (int) informado pelo usuário e retorne o fatorial (*long int*) utilizando uma função recursiva (fatorialRecursivo) e uma função iterativa (fatorialIterativo) exibindo o tempo despendido em cada uma das funções. Quando começa a discrepância de desempenho? Coloque essa resposta como comentário na código fonte do programa.

Abaixo segue um programa simples que calcula o tempo que o programa demorou para executar:

```
#include<stdio.h>
#include<time.h>

int main(int argc, char *argv[]){
    long int inicio, fim;
    long int tempo;
    inicio= clock();

    //Seu código

    fim = clock();
    tempo = fim-inicio;
    printf("%lf", ((double)tempo/CLOCKS_PER_SEC));
    return 0;
}
```

2. Crie um programa C ANSI que contenha uma função **recursiva** que dado um inteiro **x** e um inteiro **n**, calcule **xⁿ**, isto é, uma função recursiva para a função **pow** da biblioteca **math.h**. Calcule o tempo dispendido para altos valores de **n** comparando com o tempo utilizando a função **pow**. Quando começa a discrepância de desempenho? Coloque essa resposta como comentário na código fonte do programa.
3. Crie um programa C ANSI que contenha uma função **recursiva** que passado um arranjo de *double* e o tamanho deste arranjo, retorne o maior valor desse arranjo.
4. Dado a seguinte série:

f(0) = 1	f(6) = 8
f(1) = 1	f(7) = 8
f(2) = 2	f(8) = 16
f(3) = 2	...
f(4) = 4	
f(5) = 4	

Sabendo que f(0), f(1) e f(2) são valores fixos, deduza a fórmula da função f(n) para todo n>2 e crie uma função **recursiva** que a solucione.