

INFORMAÇÕES SOBRE ESTA ATIVIDADE:

- Esta atividade valerá 2,0 pontos (1,0 ponto de trabalho mais 1,0 ponto de AAI).
- A atividade deve se iniciar no laboratório e ser entregue pelo SINEF **até dia 16/set/2012**.
- Deve ser entregue um arquivo compactado chamado **Ativ03.zip** com um arquivo chamado **grupo.txt** contendo o nome dos integrantes e os arquivos **c** cujos nomes devem seguir o seguinte padrão: **Ativ03Ex01.c, Ativ03Ex02.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 03 - Lista de Exercícios – Funções

Dica: É boa prática a utilização de protótipos (quase mandatório).

1. Crie um programa C ANSI que leia dois inteiros e chame as seguintes funções a serem criadas:
 - soma
 - subtração
 - multiplicação
 - divisão
 - resto
2. Crie um programa C ANSI que leia uma *string* e chame uma função que imprima cada uma das suas posições. Nessa função que imprime cada posição, deve ser chamada uma outra função que retorne 1 se for vogal e 0 se for consoante e a partir do retorno imprima se é a letra é vogal ou consoante (Utilize op. ternário). Por exemplo, lido "ANA"
A » vogal
N » consoante
A » vogal
3. Crie um programa C ANSI que leia um arranjo de cinco números inteiros e depois chame uma função que o imprima.
Dica: a função de impressão deverá ter um parâmetro com o tamanho do arranjo.
4. Crie um programa C ANSI que leia um arranjo de cinco números inteiros e depois chame uma função que calcule a média e depois chame uma outra função que calcule a soma.
Dica: a função de impressão deverá ter um parâmetro com o tamanho do arranjo.
5. Crie um programa C ANSI que leia uma string e depois chame uma função que retorne a primeira posição da string em que foi encontrada uma vogal. Para saber se é vogal, deve ser criada uma outra função chamada **isVogal** que retorne 0 se não for vogal ou diferente de 0 se for vogal.
6. Crie um programa C ANSI que possua uma função que calcule o fatorial de forma iterativa:
 $0! = 1$
 $1! = 1$
 $2! = 2 * 1! = 2 * 1 = 2$
 $3! = 3 * 2! = 3 * 2 * 1! = 3 * 2 * 1 = 6$
7. Crie um programa C ANSI que possua uma função que calcule o *fibonacci* de forma recursiva:
$$F(n) = \begin{cases} 0, & \text{se } n = 0; \\ 1, & \text{se } n = 1; \\ F(n-1) + F(n-2) & \text{outros casos.} \end{cases}$$
8. Crie um outro programa C ANSI que, como na última questão, faça o cálculo do *fibonacci*, porém de forma iterativa.
Dica: Observe o desempenho para valores grandes de n. Suponha que faça com n = 40.