Exercício de Funções e Ponteiro - Aula 01/11/14

Resolvam os exercícios de Ponteiros e Funções

1) Crie um programa para manipular vetores. O seu programa deve implementar uma função chamada **inverte\_vetor**, que recebe como parâmetro dois vetores **V1** e **V2**, ambos de tamanho **N**. A função deve copiar os elementos de **V1** para **V2** na ordem inversa. Ou seja, se a função receber **V1 = {1,2,3,4,5},** a função deve copiar os elementos para **V2** na seguinte ordem: **V2 = {5,4,3,2,1}.** Além disso, a função também deve retornar o maior valor encontrado em **V1**. A função deve obedecer ao seguinte protótipo:  
**int inverte\_vetor(int \*v1, int \*v2, int n);**

2) Em seguida, implemente no mesmo modulo outra função chamada   
**multiplica\_escalar**, que recebe como parâmetro dois vetores **V1** e **V2** (ambos de tamanho **N**), e um número inteiro **X**. A função deve multiplicar cada um dos elementos de **V1** por **X** e armazenar os resultados em **V2**.  
A função deve obedecer ao seguinte protótipo:  
**void multiplica\_escalar(int \*v1, int \*v2, int x, int n);**

3) Em seguida crie a função principal do programa utilizando as funções **inverte\_vetor**e**multiplica\_escalar** para inverter um **vetor de tamanho 10** fornecido pelo usuário e em seguida multiplicar esse vetor por um **número também fornecido pelo usuário**. Por ultimo, o programa deverá exibir o vetor resultante **V2**.

4) Crie um programa para calcular a área e o perímetro de um hexágono. O seu programa deve implementar uma função chamada **calcula\_hexagono** que calcule a área e o perímetro de um hexágono regular de lado **L**. A função deve obedecer o seguinte protótipo:

**void calcula\_hexagono(float l, floar \*area, float \*perimetro);**

Lembrando que a área e o perímetro de um hexágono regular são dados por:

Por extenso para facilitar o entendimento da fórmula:

A= (Três \* L ao quadrado \* Raiz quadrada de 3 sobre 2)

P=¨(6 vezes L)

Para os cálculos, obrigatoriamente você deve utilizar as funções **sqrt** e **pow** da biblioteca **math.h.**