Exercícios de Revisão – Concorrência

1) Seja a seguinte função para criar *threads*:

cria_thread(<nome>, <função>, <par1, par2, p3,..., par n>)

A função cria uma *thread* cujo nome é <nome> (uma variável declarada como do tipo p_thread) que executa uma função chamada <função> cujos argumentos são <par1, par2, par 3,..., par n>..

Por exemplo,

```
p_thread t0;
float x, A, B, C;
...
float raiz(float a, float b, float c, char sinal){
...
}

cria_thread(t0,raiz, A, 3, C, '+');
```

cria uma *thread* identificada por t0 que executa a função raiz passando como parâmetros A, 3, C e '+'. Utilizando como suporte à concorrência apenas o tipo p_thread (e a função para criação de thread) e também o tipo semaphore com as operações wait(semaphore) e signal(semaphore), escreva um programa em uma linguagem de programação real que, através de **várias *threads***, calcule o valor de Pi utilizando a seguinte série:

$$\frac{\pi}{4} = \frac{1}{1} - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11} + \ldots$$

O número de *threads* concorrentes para calcular Pi é *n* e *n* deve ser lido. Cada *thread* calcula uma parte da série e armazena o resultado em uma *mesma* variável. O programa deverá ser escrito em alguma linguagem de programação real.

2) Seja o seguinte trecho de programa:

```
A = 3;
B = A;
C = A + 1;
A = A + 2;
D = B + C;
C = B + D;
```

- **a)** Mostre um grafo de precedência correspondente ao trecho acima que represente a concorrência máxima possível de ser obtida sem que os valores finais das variáveis difiram daqueles obtidos através da execução seqüencial;
- **b)** Utilizando como suporte a biblioteca descrita no Exercício 1 faça um programa que represente o grafo do ítem a).

3) Faça uma um programa que leia uma *string* de tamanho *N* e retorne 1 se o *string* representa uma palavra palíndroma; 0 caso contrário. O programa deverá *N*/2 *threads* que trabalham de forma cooperativa para determinar se o *string* é o não uma palavra palíndroma. Para desenvolvimento utilize a biblioteca descrita no Exercício 1.