MAC0438 – Programação Concorrente
Instituto de Matemática e Estatística
Primeira Prova – 7 de Maio de 2012

Nome: _____

Assinatura: _____

Nº USP: _____

Questão 1

Considere o seguinte programa:

```
int x = 10; c = true;

co <await x == 0>; c = false;
// while (c) <x = x - 1>;
oc
```

---

a) O programa terminará se o escalonador tiver justiça fraca? Justifique.
b) O programa terminará se o escalonador tiver justiça forte? Justifique.
c) Adicione o seguinte trecho de código como mais um processo dentro do co:

```
while (c) {if (x < 0) <x = 10>;}
```

Responda de novo as questões das letras **a)** e **b)** para este novo programa com três processos.

Questão 2

Considere a seguinte solução para o problema da seção crítica proposta por Leslie Lamport em 1987:

```
int lock = 0;
process CS[i=1 to n] {
   while (true) {
      <await (lock == 0)>; lock = i; Delay;
      while (lock != i) {
         <await (lock == 0)>; lock = i; Delay;
      }
      Seção crítica;
      lock = 0;
      Seção não crítica;
   }
}
```

---

a) Considere que o comando `Delay` seja removido do algoritmo. Alguma das 4 propriedades necessárias para resolver o problema da seção crítica não será atendida? Justifique detalhadamente sua resposta para **todas as propriedades não atendidas**, apresentando uma sequência de execução que prove que a propriedade não é atendida.

b) Considere que o comando `Delay` seja mantido no algoritmo. Considere também que esse comando faça processos dormirem o tempo que for necessário para que cada um dos processos `i` que esteja esperando `lock` ser zero tenha tempo de executar o trecho que faz `lock` ser igual a `i`. Nesse cenário, alguma das 4 propriedades necessárias para resolver o problema da seção crítica não será atendida? Justifique detalhadamente sua resposta para **todas as propriedades não atendidas**, apresentando uma sequência de execução que prove que a propriedade não é atendida.

Questão 3

Suponha que um computador possui a instrução atômica *Compare-and-Swap* que executa o seguinte algoritmo:

```
CSW (a, b, c):
   <if (a == c)
      {c = b; return (0);}
    else
      {a = c; return (1);}
   >
```

`a`, `b` e `c` são variáveis inteiras. Usando `CSW`, proponha um algoritmo para resolver o problema da seção crítica para `n` processos. Não se preocupe com a propriedade de entrada garantida.

Questão 4

Um aluno de MAC0438 propôs o seguinte algoritmo para uma barreira reutilizável de n processos:

```
int count = 0; go = 0 # variáveis compartilhadas

código executado pelo Worker[1]:
   <await (count == n-1);>
   count = 0;
   go = 1;

código executado pelos Workers[2:n]:
   <count++;>
   <await (go == 1);>
```

---

a) Por que o algoritmo não funciona?

b) Corrija o algoritmo. Não crie novas variáveis compartilhadas. Usar novas variáveis locais é permitido.