

1. Elabore um programa ilustrando o uso da estrita alternância.
 - a. Explique o código fonte utilizado

```
#Estrita Alternância
from threading import Thread
import time

global turn

def regiaoCritica():
    time.sleep(1)

def processamentoA(times, delay):
    global turn
    for x in range(times):
        print ("Secao de Entrada A - ",x+1)
        while (turn != 0):
            continue
        print ("Regiao Critica A")
        regiaoCritica()
        print ("Secao de Saida A")
        turn = 1
        print ("Regiao nao critica A\n")
        time.sleep(delay)

def processamentoB(times, delay):
    global turn
    for x in range(times):
        print ("Secao de Entrada B - ",x+1)
        while (turn != 1):
            continue
        print ("Regiao Critica B")
        regiaoCritica()
        print ("Secao de Saida B")
        turn = 0
        print ("Regiao nao critica B\n")
        time.sleep(delay)
```

```

print ("Exemplo de Estrita Aternancia")
execTimes = 3
turn = 0

#no processamento você pode passar quantas vezes que a exec e
#qual o tempo de delay para simular o efeito comboi
tA = Thread(target=processamentoA, args=(execTimes,1,))
tA.start()
tB = Thread(target=processamentoB, args=(execTimes,10,))
tB.start()

```

b. Descreva os resultados obtidos

Efeito comboio, o outro processo fica aguardando a finalização da seção não crítica desnecessariamente, e a variável “turn” deveria ser considerada como uma região crítica, então parece ter um erro conceitual neste código.

2. Elabore um programa ilustrando o uso da Solução de Peterson.

a. Explique o código fonte utilizado

```

#Solucao Peterson
from threading import Thread
import time

global turn, i, j, flag

def regiaoCritica():
    time.sleep(1)

def processamentoA(times, delay):
    global turn, i, j, flag
    for x in range(times):
        print ("Secao de Entrada A - ",x+1)

```

```

        flag[i] = True
        turn = j
        while (flag[j] and turn == j):
            continue
        print ("Regiao Critica A")
        regiaoCritica()
        print ("Secao de Saida A")
        flag[i] = False
        print ("Regiao nao critica A\n")
        time.sleep(delay)

def processamentoB(times, delay):
    global turn, i, j, flag
    for x in range(times):
        print ("Secao de Entrada B - ",x+1)
        flag[j] = True
        turn = i
        while (flag[i] and turn == i):
            continue
        print ("Regiao Critica B")
        regiaoCritica()
        print ("Secao de Saida B")
        flag[j] = False
        print ("Regiao nao critica B\n")
        time.sleep(delay)

print ("Exemplo de Solucao de Peterson")
execTimes = 5
turn = 0
i = 0
j = 1
flag = []
flag.append(False)
flag.append(False)

```

```
#no processamento você pode passar quantas vezes que a exec e
#qual o tempo de delay para simular o efeito comboi
tA = Thread(target=processamentoA, args=(execTimes,1,))
tA.start()
tB = Thread(target=processamentoB, args=(execTimes,5,))
tB.start()
```

b. Descreva os resultados obtidos

Problema do efeito comboio resolvido, mas o código ainda possui o erro conceitual.

3. Elabore um programa ilustrando a utilização de semáforo.

a. Explique o código fonte utilizado

```
#Semáforo
from threading import Thread, Semaphore
import time

s = Semaphore()

def regiaoCritica():
    time.sleep(1)

def processamentoA(times, delay):
    for x in range(times):
        print ("Secao de Entrada A - ",x+1)
        s.acquire()
        print ("Regiao Critica A")
        regiaoCritica()
        print ("Secao de Saida A")
        s.release()
        print ("Regiao nao critica A\n")
        time.sleep(delay)

def processamentoB(times, delay):
    for x in range(times):
        print ("Secao de Entrada B - ",x+1)
        s.acquire()
```

```
    print ("Regiao Critica B")
    regiaoCritica()
    print ("Secao de Saida B")
    s.release()
    print ("Regiao nao critica B\n")
    time.sleep(delay)

print ("Exemplo de Semafaro")
execTimes = 5

#no processamento você pode passar quantas vezes que a exec e
#qual o tempo de delay para simular o efeito comboi
tA = Thread(target=processamentoA, args=(execTimes,1,))
tA.start()
tB = Thread(target=processamentoB, args=(execTimes,5,))
tB.start()
```

b. Descreva os resultados obtidos