

<p>INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO CEARÁ DEPARTAMENTO DE TELEMÁTICA AVALIAÇÃO DE SOFTWARE DE TEMPO-REAL / PROF. PAULO RÉGIS NOME:</p>
--

01. Elabore um escopo de código em linguagem C que apresente um bloco de recuperação para trás com um módulo primário e três módulos alternativos. Cada módulo deve retornar o valor 6, e deve ter escopo diferente.
02. No escopo do monitor abaixo, como deve ser inicializada a variável FREE e preencha as atribuições da variável busy.

```
INTERFACE MODULE resource_control;
```

```
    DEFINE allocate, deallocate;
```

```
    VAR busy : BOOLEAN;  
        Free : SIGNAL;
```

```
    PROCEDURE allocate;  
    BEGIN  
        IF busy THEN WAIT(free) END;  
        busy := XXX  
    END;
```

```
    PROCEDURE deallocate;  
    BEGIN  
        busy := XXX  
        SEND(free)  
    END;
```

```
    BEGIN (*inicialização do módulo*)  
        busy := XXX  
    END
```

03. Elabore um escopo de código em C que implemente a exclusão mútua sobre 3 recursos idênticos de 3 processos concorrentes. Utilizar semáforo de quantidade para implementar a exclusão mútua. Cada processo só pode acessar um recurso por vez. Cada processo deve acessar os 3 recursos. Não pode haver deadlock. As variáveis do semáforo de quantidade devem ser inicializadas corretamente.
04. O escopo de código a seguir apresenta uma espera seletiva realizada por um processo qualquer. Implemente, em linguagem C, a função *alt_wait()* e a função *receive_msg_sinc_timeout(&y,tempo,x)*. A função *receive_msg_sinc_timeout()* é uma função de recepção de mensagem sincronizada, armazenada e temporizada.

```
CHAN CAN[3];  
int x, y;  
x = alt_wait(3, CAN[0], CAN[1], CAN[2]);  
receive_msg_sinc_timeout(&y, tempo, x);
```