

Nome: \_\_\_\_\_ Data: \_\_/\_\_/\_\_\_\_

**Exercício 1)** Implemente um método recursivo para determinar se uma sequência *S* de *n* objetos inteiros contém um dado inteiro *k*. O método não pode conter laços.

**Exercício 2)** Considere a implementação de listas simplesmente encadeadas utilizando apontadores e com célula cabeça-de-lista *head* do tipo *Nodo*. Considere que um dos campos do tipo *Nodo* é uma chave *ch* do tipo inteiro (*int*). Escreva a seguinte função:

*boolean EstaNaLista(int ch, Nodo head);*

Esta função retorna *TRUE* se *ch* estiver na lista e retorna *FALSE* se *ch* não estiver na lista. Considere que não há ocorrências de chaves repetidas na lista.

**Exercício 3)** No Brasil existe o CPF (Cadastro de Pessoas Físicas) que serve para identificar cada indivíduo no país. O número do CPF é composto de 11 dígitos, sendo os dois últimos os dígitos de verificação. Escreva uma Rotina de validação de CPF que recebe como entrada um número, do tipo *LONG*, e retorna um valor *BOOL* informando o resultado da validação. A rotiva deve verificar 2 itens:

a) Validar se o número contém apenas 11 dígitos. Qualquer ocorrência contrária, devolver *false*;

b) Validar o número pelo dígito verificador.

Obs: Descrição de como funciona a rotina de validação do CPF.

A fórmula para verificar a validade do número do CPF é simples e é explicada abaixo:

Vamos tomar como exemplo o número 123.456.789-09

• 1º Dígito Verificador

Primeiro calculamos a soma da multiplicação dos 9 primeiros dígitos por 10, 9, 8, ... , 3, 2, respectivamente. Ou seja

$$\text{Soma} = (1 \cdot 10) + (2 \cdot 9) + \dots + (8 \cdot 3) + (9 \cdot 2)$$

Em seguida, dividimos e multiplicamos por 11. (Nota: Ao multiplicarmos utilizamos o valor inteiro da divisão).

$$\text{Valor} = (\text{Soma} / 11) \cdot 11$$

Por fim, subtraímos Valor de Soma.

$$\text{Resultado} = \text{Soma} - \text{Valor}$$

Note que acabamos de realizar o módulo de Soma e 11. As duas operações anteriores podem ser substituídas por Resultado = Soma módulo 11.

Agora analisamos Resultado:

Se Resultado for igual à 1 ou à 0, então o 1º dígito verificador é 0;

Caso contrário, o 1º dígito verificador é o resultado da subtração de Resultado de 11.

• 2º Dígito Verificador

Primeiro calculamos a soma da multiplicação dos 9 primeiros dígitos por 11, 10, 9, ... , 4, 3, respectivamente e em seguida somamos com (Dígito1\*2), sendo que Dígito1 é o valor encontrado para o 1º dígito verificador. Ou seja

$$\text{Soma} = (1 \cdot 11) + (2 \cdot 10) + \dots + (8 \cdot 4) + (9 \cdot 3) + (\text{Dígito1} \cdot 2)$$

O resto é semelhante ao que foi feito anteriormente.  
Resultado = Soma módulo 11.

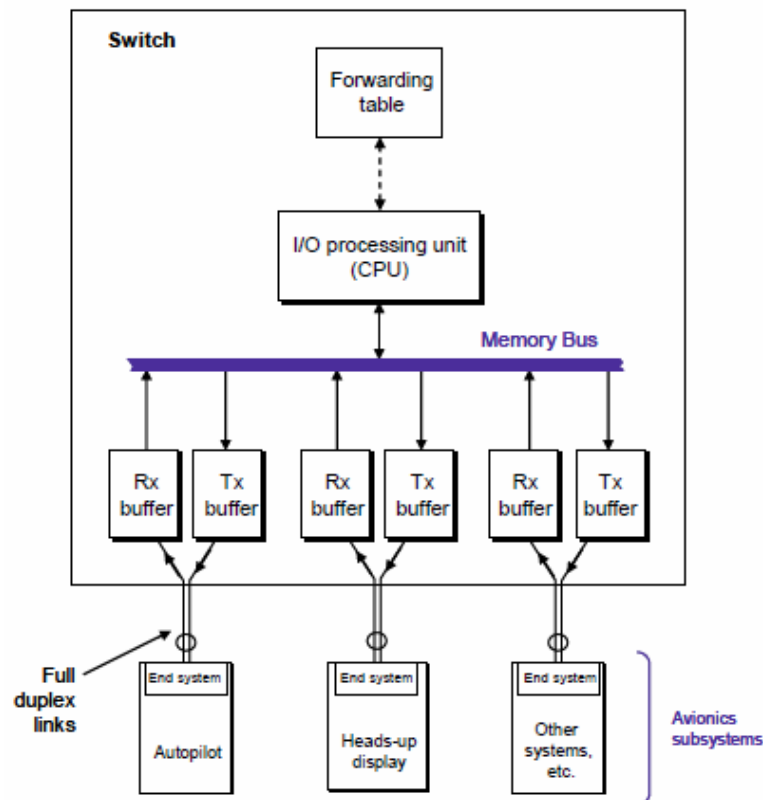
Agora analisamos Resultado:

Se Resultado for igual à 1 ou à 0, então o 2º dígito verificador é 0;  
Caso contrário, o 2º dígito verificador é o resultado da subtração de Resultado de 11.

No nosso exemplo (123.456.789-09) o número é válido.

#### Exercício 4) Interpretação de texto:

*“... To do away with contention (collisions), and hence the indeterminacy regarding how long a packet takes to travel from sender to receiver, it is necessary to move to Full-duplex Switched Ethernet. Full-duplex Switched Ethernet eliminates the possibility of transmission collisions like the ones that occur when using Half-duplex Based Ethernet. As shown in figure below, each Avionics Subsystem— autopilot, heads-up display, etc. — is directly connected to a Switch over a full-duplex link that comprises two twisted pairs — one pair for transmit (Tx) and one pair for receive (Rx). The switch comprises all the components contained in the large box. The switch is able to buffer packets for both reception and transmission...”*



**Regarding the text above, it's correct to state that:  
(more than one correct alternative is possible).**

- ( ) Full duplex communication is only possible due to the using of packet buffers.
- ( ) Collisions are the consequence of long packets travelling from transmitter to receiver.
- ( ) The use of full-duplex switch instead of a half-duplex one prevents collisions.
- ( ) The packets move from the half-duplex link to the full-duplex link through the Rx and Tx buffers.
- ( ) In this system there are full-duplex switched communication links where transmitted and received data travel and are buffered by means of exclusive physical media (twisted wire pairs) and buffering devices.

**Exercício 5) Resolva o seguinte problema descrito em ASCII:**

`byte[]` Problema = { 112, 114, 111, 118, 97, 58, 32, 116, 114, 97, 122, 101, 114, 32, 97, 108, 103, 111, 114, 105, 116, 109, 111, 32, 113, 117, 101, 32, 105, 109, 112, 114, 105, 109, 105, 32, 116, 111, 100, 111, 115, 32, 111, 115, 32, 110, 117, 109, 101, 114, 111, 115, 32, 112, 114, 105, 109, 111, 115, 32, 101, 110, 116, 114, 101, 32, 48, 32, 101, 32, 49, 48, 48, 46};

**Exercício 6)** Quando um lote de ações de uma companhia é vendido, o ganho de capital (ou, às vezes, a perda) é a diferença entre o preço de venda do lote e o preço originalmente pago por ele. Essa regra é fácil de entender para um único lote, mas se vendermos múltiplos lotes comprados em tempos diferentes temos de identificar os lotes sendo vendidos. Um método usado para identificar que lotes são vendidos é usar uma estrutura que suporte o protocolo FIFO, na qual os lotes vendidos são aqueles que temos há mais tempo (este é o método usado em vários softwares de finanças pessoais). Por exemplo, suponha que compramos 100 lotes a \$20 cada no dia 1, 20 lotes a \$24 no dia 2, 200 lotes a \$36 no dia 3 e vendemos 150 lotes no dia 4 a \$30 cada. Aplicando o protocolo FIFO, significa que dos 150 lotes vendidos, 100 foram comprados no dia 1, 20 foram comprados no dia 2 e 30 foram comprados no dia 3. O ganho de capital neste caso seria  $100 \cdot 10 + 20 \cdot 6 + 30 \cdot (-6)$ , ou \$940.

Escreva um programa que recebe como entrada uma sequência de transações da forma

•compre X lotes a \$Y cada

ou

•venda X lotes a \$Y cada

assumindo que as transações ocorrem em dias consecutivos e que os valores x e y são inteiros. Dada esta sequência de entradas, a saída deve ser o ganho total (ou perda) de capital para a sequência completa, usando o protocolo FIFO para identificar os lotes.

Obs: Utilize uma lista encadeada para armazenar os elementos.

**Considerações:**

Desenvolver utilizando preferencialmente a Linguagem C++.

Você deve enviar sua resposta para o endereço de e-mail([thiago@verion.com.br](mailto:thiago@verion.com.br))

A avaliação das questões levará em consideração não somente a correção das respostas, mas também a abordagem e organização adotadas.