



**Valor:** 2,5 pontos

**Correção:** ganha os pontos totais se passar no BOCA e implementar as funções como foram pedidas, ou até 1,5 dependendo do que foi implementado considerando o uso das funções pedidas e dos comandos necessários para a solução.

**Problema:** (BOCA: P2\_2018\_Q4\_a) Um programador resolveu codificar a sua senha de 6 dígitos como números de um texto. Os valores relevantes para a montagem da senha são armazenados em números que estão entre parênteses no texto. Cada número entre parênteses representa um dígito secreto. Um dígito secreto é dado pela soma dos algarismos do número entre parênteses, sucessivamente até que só reste um dígito (por exemplo,  $29 = 2 + 9 = 11$ ;  $11 = 1 + 1 = 2$ , portanto, 29 codifica o dígito secreto 2). A senha é montada inserindo os dígitos secretos pela direita de acordo com a ordem de ocorrência (por exemplo, se apareceram no texto os dígitos secretos 3, depois o 2 e depois o 6, isso formaria a senha 326). Faça um programa que converta os números entre parênteses para o dígito secreto, e ao final imprima a senha codificada.

- Entrada: um texto terminado com ponto '.' e com alguns números dados entre parênteses. Atentar para o fato do número do texto poder ser maior do que o limite de armazenamento de um inteiro. Portanto, não será possível ler o número todo de uma vez para uma variável inteira. Assuma que os parênteses sempre virão em pares, "(" e ")", e que o conteúdo entre eles sempre será um número, ou será vazio.
- Saída: o texto original com os números entre parênteses reduzidos ao dígito secreto. Caso o conteúdo de um par de parênteses seja vazio, imprimir "ERRO", ao invés do dígito reduzido para o par de parênteses em questão e esse par não deverá ser considerado para a montagem da senha. Ao final, imprimir "SENHA:#1", em que #1 representa a senha codificada. Ver exemplos abaixo!

O programa deve ser entregue modularizado, para isso crie:

- a) Uma função que recebe um inteiro  $n$  e retorne a sua versão reduzida a um dígito, i.e., seguindo a soma sucessiva descrita acima. Cabeçalho: *int ReduzInteiro(int n);*
- b) Uma função responsável por montar a senha, ou seja, incluir mais um dígito na senha. A função recebe dois inteiros, um representando a senha até o momento e o outro representando o dígito a ser incluído na parte menos significativa do número, ou seja, no lado direito da senha. A função retorna um inteiro representando a *senha* com o dígito *dig* na posição correta (ou seja, se a função recebe *senha=23* e *dig=7*, ela retorna o inteiro 237). Cabeçalho: *int MontaSenha(int senha, int dig);*
- c) Uma função que lê os dígitos de um número a ser reduzido e retorna a sua versão reduzida a um dígito. Caso não haja número a ser lido, a função deverá retornar -1. Assuma que a função será chamada no início da leitura dos dígitos, ou seja, logo após a leitura da abertura de um parênteses. Cabeçalho: *int ReduzCodigo();*

- Exemplo de Entrada:

Lembra das questoes L2 22 e L2 23 (99999999999999999999)? Entao, (99999999999989999999)(99999999999989999998)() eh soh implementar (12345678909832) o que for diferente (1) (12).
Olha a lista ai gente!!! Essa questao () 123 (777) 777 (12345) 0 (0)1(6)(11111111111111111111)(56) 123.
(98)(645)(222)(77777777777777777777)(423)(2234).

- Exemplo de Saída:

Lembra das questoes L2 22 e L2 23 [9]? Entao, [8][7][ERRO] eh soh implementar [4] o que for diferente [1] [3]. SENHA:987413
Olha a lista ai gente!!! Essa questao [ERRO] 123 [3] 777 [6] 0 [0]1[6][1][2] 123. SENHA:360612
[8][6][6][7][9][2]. SENHA:866792