

## Roteiro 2 – Loops (Repetições)

### Desenvolva/modifique os seguintes algoritmos no IC:

1. Modifique o algoritmo de cálculo do peso ideal feito na aula anterior para funcionar em um *loop* que será encerrado quando o usuário decidir (leitura de opção).

*Para lembrar:*

O cálculo do peso ideal de uma pessoa é dado por:

para homens:  $(72.7 * altura) - 58$

para mulheres:  $(62.1 * altura) - 44.7$

Usar sempre **pontos** em números fracionários, e não vírgula.

Ler do usuário: o **sexo** (como número ou como letra) e a **altura**;

Calcular e mostrar: o **peso ideal**.

Mostrar mensagem de erro se o usuário digitar opção errada para sexo.

2. Os sistemas de segurança simples são geralmente baseados em **senhas** pessoais, que o usuário deve digitar para ter seu acesso liberado. Escreva um algoritmo para um sistema de segurança simplificado. As diretivas são:
  - a. O sistema terá uma senha **definida internamente, dentro** do algoritmo. Esta senha pode ser **numérica** ou pode ser um **caracter**.
  - b. O algoritmo deve pedir para o usuário digitar a senha e então compará-la à senha armazenada internamente. (Lembrar que, para comparações e atribuições, caracteres devem estar entre aspas simples. Ex.1: `char senha = 'x'`; Ex.2: `if (senha != 'x') ...`).
  - i. Se a senha estiver correta, o sistema deve dar acesso ao usuário (imprimir, por exemplo, “acesso autorizado”).
  - ii. Se a senha estiver incorreta, o acesso não deve ser permitido (imprimir, por exemplo, “acesso negado”).
  - c. Após a autorização ou negação de acesso, o sistema deve voltar a pedir nova entrada de senha (para novas tentativas de acesso).
  - d. O programa deve ser encerrado quando o usuário decidir.
3. Modificar o algoritmo anterior de forma a que mais de uma senha esteja cadastrada internamente no algoritmo. Por exemplo, uma senha para o usuário 1, outra senha para o usuário 2, e assim por diante.
  - a. O sistema deve, portanto, pedir a digitação da identificação do usuário, digitação da senha e então comparar com a senha armazenada para aquele usuário.
  - b. Se o usuário não estiver cadastrado, dar aviso de erro (ex.: “usuário inexistente”).
  - c. Após o aviso de erro, ou o aviso de acesso liberado ou não, o algoritmo deve voltar a pedir nova entrada de identificação de usuário e senha (para novas tentativas de acesso).
  - d. O programa deve ser encerrado quando o usuário decidir.