

Lista de lógica Aplicada à programação

Exercício 1 – Compra Online

Uma loja virtual automatiza o processo de análise de compra com base em três critérios.

Tarefa: Identifique as proposições lógicas utilizadas no código abaixo. Formule a proposição lógica composta. Elabore a coluna final da tabela verdade.

```
def compra(p, q, r):  
    if p and q and r:  
        return "Compra aprovada"  
    else:  
        return "Compra negada"
```

Proposições:

- p: O produto está disponível em estoque.
 - q: O pagamento foi aprovado.
 - r: O endereço de entrega é válido.
-

Exercício 2 – Login no Sistema

O sistema de login exige três condições verdadeiras para autorizar o acesso de um usuário.

Tarefa: Identifique as proposições lógicas utilizadas no código abaixo. Formule a proposição lógica composta. Elabore a coluna final da tabela verdade.

```
def login(p, q, r):  
    if p and q and r:  
        return "Login bem-sucedido"  
    else:  
        return "Falha no login"
```

Proposições:

- p: O nome de usuário está correto.
- q: A senha informada está correta.

- r: A autenticação de dois fatores foi validada.
-

Exercício 3 – Acesso Condicional

Um sistema controla o acesso com base em três condições específicas.

Tarefa: Identifique as proposições lógicas utilizadas no código abaixo. Formule a proposição lógica composta. Elabore a coluna final da tabela verdade.

```
def acesso(p, q, r):  
    if p or (q and r):  
        return "Acesso permitido"  
    else:  
        return "Acesso negado"
```

Proposições:

- p: O usuário possui credencial especial.
 - q: O usuário está registrado.
 - r: O usuário confirmou a identidade.
-

Exercício 4 – Acesso Condicional 2

Um sistema alternativo avalia o acesso com base em credenciais e validação.

Tarefa: Identifique as proposições lógicas utilizadas no código abaixo. Formule a proposição lógica composta. Elabore a coluna final da tabela verdade.

```
def acesso(p, q, r):  
    if p or (q and r):  
        return "Acesso permitido"  
    else:  
        return "Acesso negado"
```

Proposições:

- p: O usuário está na lista branca de acesso.
- q: O cadastro está completo.

- r: O número de identificação foi verificado.
-

Exercício 5 – Entrada em Evento

A entrada para um evento depende de regras combinadas envolvendo inscrição e outros critérios.

Tarefa: Identifique as proposições lógicas utilizadas no código abaixo. Formule a proposição lógica composta. Elabore a coluna final da tabela verdade.

```
def entrada(p, q, r, s):  
    if p or (q and r and not s):  
        return "Entrada liberada"  
    else:  
        return "Entrada negada"
```

Proposições:

- p: O participante tem convite impresso.
 - q: A inscrição online foi realizada.
 - r: O pagamento foi confirmado.
 - s: Há pendência cadastral.
-

Exercício 6 – Desbloqueio de Celular

Um sistema de segurança controla o desbloqueio de um celular com base em diferentes combinações de ações.

Tarefa: Identifique as proposições lógicas utilizadas no código abaixo. Formule a proposição lógica composta. Elabore a coluna final da tabela verdade.

```
def desbloquear(p, q, r, s):  
    if (p or q) and r and s:  
        return "Celular desbloqueado"  
    else:  
        return "Acesso negado"
```

Proposições:

- p: A digital foi reconhecida.
 - q: A senha foi digitada corretamente.
 - r: O aparelho está conectado à internet.
 - s: A verificação facial foi aprovada.
-

Exercício 7 – Acesso Seguro

O acesso a uma área segura depende da verificação de diversos dados.

Tarefa: Identifique as proposições lógicas utilizadas no código abaixo. Formule a proposição lógica composta. Elabore a coluna final da tabela verdade.

```
def acesso_seguro(p, q, r, s):  
    if p and q and (r or s):  
        return "Acesso liberado"  
    else:  
        return "Acesso negado"
```

Proposições:

- p: A senha foi validada.
 - q: O token foi inserido.
 - r: A biometria foi reconhecida.
 - s: O cartão de acesso foi apresentado.
-

Exercício 8 – Controle de Iluminação

Um sistema inteligente de iluminação decide quando acender a luz com base em sensores.

Tarefa: Identifique as proposições lógicas utilizadas no código abaixo. Formule a proposição lógica composta. Elabore a coluna final da tabela verdade.

```
def acender_luz(p, q, r, s):  
    if p and (q or r) and s:  
        return "Luz acesa"  
    else:  
        return "Luz apagada"
```

Proposições:

- p: Movimento foi detectado.
 - q: O ambiente está escuro.
 - r: A luz natural é insuficiente.
 - s: A lâmpada está funcionando.
-

Exercício 9 – Avaliação de Candidatos

Um processo seletivo filtra candidatos com base em múltiplos critérios de exclusão e inclusão.

Tarefa: Identifique as proposições lógicas utilizadas no código abaixo. Formule a proposição lógica composta. Elabore a coluna final da tabela verdade.

```
def avaliar_candidato(p, q, r, s, t):  
    if p and q and r and not s and not t:  
        return "Convidar"  
    else:  
        return "Não convidar"
```

Proposições:

- p: O candidato tem formação adequada.
 - q: O candidato tem experiência mínima exigida.
 - r: O candidato tem disponibilidade.
 - s: O candidato tem pendências jurídicas.
 - t: O candidato apresentou documentação incompleta.
-

Exercício 10 – Processo Industrial

Um processo de produção só inicia se todas as variáveis estiverem dentro dos parâmetros exigidos.

Tarefa: Identifique as proposições lógicas utilizadas no código abaixo. Formule a proposição lógica composta. Elabore a coluna final da tabela verdade.

```
def iniciar_maquina(p, q, r, s, t):  
    if p and q and r and s and t:  
        return "Iniciar processo"  
    else:  
        return "Falha no processo"
```

Proposições:

- p: Temperatura estável.
 - q: Pressão controlada.
 - r: Todos os sensores operacionais.
 - s: Sistema calibrado.
 - t: Matéria-prima disponível.
-

Exercício 11 – Liberação de Sistema Crítico

Um sistema crítico de segurança só pode ser liberado sob total conformidade de todos os critérios.

Tarefa: Identifique as proposições lógicas utilizadas no código abaixo. Formule a proposição lógica composta. Elabore a coluna final da tabela verdade.

```
def liberar(p, q, r, s, t):  
    if p and q and r and s and t:  
        return "Sistema liberado"  
    else:  
        return "Bloqueado"
```

Proposições:

- p: Acesso autorizado.
- q: Testes de segurança realizados.
- r: Operador qualificado.
- s: Supervisão aprovada.
- t: Condições ideais no sistema.

Exercício 12 – Alarme de Segurança

Um alarme é ativado com base em sensores múltiplos, incluindo variações de movimento e tempo.

Tarefa: Identifique as proposições lógicas utilizadas no código abaixo. Formule a proposição lógica composta. Elabore a coluna final da tabela verdade.

```
def ativar_alarme(p, q, r, s, t):  
    if p and ((q or r) or s) and t:  
        return "Alarme ativado"  
    else:  
        return "Alarme desativado"
```

Proposições:

- p: O sistema está armado.
- q: Detecção de movimento.
- r: Ruído acima do normal.
- s: Porta aberta sem autorização.
- t: Período noturno.

Gabarito: Coluna Final da Tabela Verdade

Ex1 (Compra Online - 8 linhas):

F F F F F F F V

Ex2 (Login no Sistema - 8 linhas):

F F F F F F F V

Ex3 (Acesso Condicional - 8 linhas):

F F F V V V V V

Ex4 (Acesso Condicional 2 - 8 linhas):

F F F V V V V V

Ex5 (Entrada em Evento - 16 linhas):

F F F F F F V F V V V V V V V V

Ex6 (Desbloqueio de Celular - 16 linhas):

FFFFFFFFVFFFVFFFV

Ex7 (Acesso Seguro - 16 linhas):

FFFFFFFFFFFFFFFFVVV

Ex8 (Controle de Iluminação - 16 linhas):

FFFFFFFFFFFFVFVFV

Ex9 (Avaliação de Candidatos - 32 linhas):

FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFVFFF

Ex10 (Processo Industrial - 32 linhas):

FFV

Ex11 (Liberação de Sistema Crítico - 32 linhas):

FFV

Ex12 (Alarme de Segurança - 32 linhas):

FFFFFFFFFFFFFFFFFFFFVFVFVFVFVFVFVFV
