

01 Quais as questões de projeto para:

- a) Comuns a todas as sentenças de seleção e de controle de iteração?
- b) Estruturas de seleção?
- c) Sentenças de seleção múltipla?
- d) Todas as sentenças de controle iterativa?
- e) Sentenças de laços controlados por contador?

02 O que é uma sentença de laço com pré-teste? E com pós-teste?

03 Qual é a diferença entre a sentença **for** de C++ e a de Java?

04 De que maneira a sentença **for** de C é mais flexível do que a de muitas outras linguagens?

05 Quais são os argumentos, a favor e contra, do uso de endentação em Python para especificar sentenças compostas em construções de controle?

06 Reescreva o seguinte segmento de pseudocódigo usando uma estrutura de laço nas linguagens: C, Python e Ruby

```
k = (j + 13) / 27
loop:
    if k > 10 then goto out
    k = k + 1
    i = 3 * k - 1
    goto loop
out:
    print i, k
```

07 Reescreva o seguinte segmento de código usando uma sentença de seleção múltipla nas linguagens C++, Python e Ruby:

```
if ((k == 1) || (k == 2)) j = 2 * k - 1
if ((k == 3) || (k == 5)) j = 3 * k - 1
if (k == 4) j = 4 * k - 1
if ((k == 6) || (k == 7) || k == 8)) j = k - 2
print k, j
```

08 Em uma carta para o editor da CACM, Rubin (1987) usa o seguinte segmento de código como evidência de que a legibilidade de algum código com **gotos** é melhor do que o código equivalente sem **gotos**. Esse código encontra a primeira linha de uma matriz inteira n por n chamada x que não tem nada além de valores iguais a zero. Reescreva esse código sem **gotos** na linguagem C++. Compare a legibilidade de seu código com o do exemplo:

```
for (i = 1; i <= n; i++){
    for (j = 1; j <= n; j++){
        if (x[i][j] != 0)
            goto reject;
        print ('Primeira linha de zeros: ', i);
        break;
    }
reject:
}
```

09 Considere o seguinte segmento de programa em C. Reescreva-o sem usar **breaks**.

```
j = -3;
for (i == 0; i < 3; i++){
    switch (j + 2){
        case 3:
        case 2:
            j--;
            break;
        case 0:
            j += 2;
            break;
        default:
            j = 0;
    }
    if (j > 0)
        break;
    j = 3 - 1;
}
```