11-Faça um programa simplificado para ver se o usuário pode pegar um empréstimo no banco. O critério é que ele tem que ser maior de idade (18 anos ou mais) e não pode ter saldo bancário negativo. Se o problema for só idade, avise isso; se o problema for só o saldo, avise também. Se o problema for os dois, imprima "Não né, não". E se o usuário entrar com idade negativa, simplesmente imprima "Erro" e não faça a classificação.

13-Sendo dois valores de resistores dados pelo usuário, nós temos quatro possibilidades de associação resistiva: só o 1°, só o 2°, associação paralela dos dois ou associação em série. Sendo um valor final também fornecido pelo usuário, determine qual das 4 configurações possíveis é a configuração com o valor mais próximo do desejado.

1. Você já está planejando onde vai morar depois de ganhar na loteria e quer fazer um programa que ajude na decisão. O programa deverá avaliar a temperatura média anual **tm** de uma cidade proposta e a distância que ela está da comidinha de mamãe, pelos seguintes critérios: cidades aceitáveis estão em até 10.000 km da sua mãe; se **tm** for abaixo de 10 ou acima de 30 graus, é inaceitável. Se **tm** estiver entre 20 e 25 graus, "perfeito"; casos remanescentes são "okay". O programa pede ao usuário a distância e a **tm** e imprime na tela o parecer final. Se a distância entrada for negativa, apenas dê um xingão no usuário.
2. "O exercício mais chato do mundo": determine a porta lógica. O usuário vai entrar com as 4 linhas de saída de uma tabela verdade (são 4 variáveis que valem 0 ou 1) e o programa deverá imprimir a função booleana correspondente. Ex.1: se os dados são 0 0 0 1, isso representa uma porta AND (tipo A.B). Ex.2: se os dados são 0 0 1 1 isso é só um buffer (uma "porta não-inversora", tipo A). Se algum dado for diferente de 0 e 1, o programa deverá informar que há um erro. Note que são dezesseis possibilidades.
3. Faça um programa que lê 3 variáveis booleanas A, B e C e imprime o resultado da equação A̅.B+B̅.C. Ou seja, se A=1, B=0 e C=1, o programa deverá imprimir 1 (pois 1̅.0+0̅.1 = 1).
4. Dados dois números x e y pelo usuário, calcule se 5\*x é menor ou não do que y ao cubo e imprima se é ou não é. Se algum dos números for negativo, imprima "Atenção! Um ou mais números são negativos!" antes do resultado; se algum dos números for zero, imprima "Uêpa! Um ou mais números são zero!" antes do resultado (se um número for zero e outro for negativo, a mensagem dos negativos não deve ser impressa).
5. Para montar um projeto (tá, é um armário) você precisa de 12 parafusos grandes com porcas, 8 pequenos de madeira e 4 dobradiças. O usuário vai entrar com os preços de 1 parafuso grande, de 1 pequeno e de 1 dobradiça para duas lojas diferentes. O programa deverá imprimir o preço final da loja mais barata. Adicionalmente, se algum preço for negativo (indicando que o produto está em falta), a mensagem "A loja 1 não tem os suprimentos" (se for a loja 1 com falta; se for a 2, "a loja 2...", claro) deverá ser impressa antes do preço final da loja mais barata. Se ambas as lojas não tiverem algum produto, informe isso e não imprima preço nenhum.
6. Qual a diferença dos programas 1 e 2 mostrados abaixo? Ou seja, como eles funcionam? Explique a diferença testando os valores -333, 0, 15 e 48 em ambos os programas. OPCIONAL: o programa 3 é igual ao programa 1 ou ao programa 2?
7. Faça um programa que imprime uma contagem de 0 a 20, inclusive.
8. Faça um programa que imprime uma contagem regressiva de 999 a 1, inclusive.
9. Faça um programa que lê**n**notas de prova de 1 aluno e imprime a média destas provas. O número**n** de provas deve ser perguntado ao usuário no início do programa.
10. Faça um programa que determina se um número **n** entrado pelo teclado é primo ou não. *Dica:* faça uma contagem de 1 até n para verificar quantos divisores esse n possui (se n%x==0, então x é divisor de n -- por exemplo, 12%5 dá 2, portanto não divide; 21%7 dá 0, portanto divide). Se o n só for divisível por 1 e por n, ele é primo.
11. Faça um programa que lê um número positivo do teclado e calcula o cubo dele. Se o usuário digitar um número negativo, o programa deverá pedir repetidamente que ele entre um número válido, até que um número positivo seja digitado.
12. Faça um programa que repetidamente lê temperaturas em Celsius e converte elas pra Kelvin. O programa só vai ser encerrado quando o usuário digitar -1000 como entrada. Exemplo de interação (a digitação do usuário está em negrito)