



INSTITUTO FEDERAL

Goiano

Campus Morrinhos

Desenvolvimento para Dispositivos Móveis - DART

Curso Tecnologia em Sistemas para Internet

Professor Msc. Marcel Melo

1. Dada uma palavra, calcule a pontuação Scabble desta palavra. Leve em consideração o valor de cada letra conforme a tabela seguir:

Letras	Valor
A, E, I, O, U, L, N, R, S, T	1
D, G	2
B, C, M, P	3
F, H, V, W, Y	4
K	5
J, X	8
Q, Z	10

Exemplo:

Banana deve ser pontuado:

B = 3 pontos

A = 1 ponto / cada

N = 1 pontos

Resultado = $3 + (1 \cdot 3) + (1 \cdot 2) = 8$ pontos

2. Se você precisa construir um componente usando o Raspberry Pi, provavelmente usará resistores. Para este exercício, você precisa saber duas coisas sobre eles:

- Cada resistor tem um valor de resistência.
- Os resistores são pequenos - tão pequenos que, se você imprimir o valor da resistência neles, seria difícil de ler.

Para resolver esse problema, os fabricantes imprimem faixas codificadas por cores nos resistores para definir seus valores de resistência. Cada faixa tem uma posição e um valor numérico.

As 2 primeiras faixas de um resistor têm um esquema de codificação simples: cada cor é mapeada para um número único. Por exemplo, se eles imprimissem uma faixa marrom (valor 1) seguida por uma faixa verde (valor 5), seria traduzido para o número 15.

Neste exercício, você criará um programa em Java para não precisar se lembrar dos valores das faixas. O programa receberá nomes de cores como entrada e produzirá um número de dois dígitos, mesmo que a entrada tenha mais de duas cores! As cores das faixas são codificadas da seguinte forma:

- Preto: 0
- Marrom: 1

- Vermelho 2
- Laranja: 3
- Amarelo: 4
- Verde: 5
- Azul: 6
- Violeta: 7
- Cinza: 8
- Branco: 9

Do exemplo acima: marrom-verde deve retornar 15 marrom-verde-violeta também deve retornar 15, ignorando a terceira cor.

3. Dado um número, determine se ele é ou não válido de acordo com a fórmula de Luhn. O algoritmo Luhn é uma fórmula de soma de verificação simples usada para validar uma variedade de números de identificação, como números de cartão de crédito. O objetivo deste exercício é determinar se uma String de números é ou não válida.

Validando um número

- Strings de comprimento 1 ou menos não são válidas. Espaços são permitidos na entrada, mas devem ser removidos antes da verificação. Todos os outros caracteres que não sejam dígitos não são permitidos.

Exemplo: Número de cartão de crédito válido

4539 3195 0343 6467

A primeira etapa do algoritmo de Luhn é remover os espaços e dobrar cada segundo dígito, começando pela direita. nós estaremos dobrando

4_3_3_9_0_4_6_6_

Se ao dobrar o número resultar em um número maior que 9, subtraia 9 do produto. Os resultados da nossa duplicação:

8569 6195 0383 3437

Em seguida, some todos os dígitos:

8+5+6+9+6+1+9+5+0+3+8+3+3+4+3+7 = 80

Se a soma for divisível por 10, o número é válido. Este número é válido! Caso contrário, o número é inválido.