<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0"/>

<title>Garage Door Spring Calculator</title>

<style>

body { font-family: sans-serif; padding: 2rem; background: #f9f9f9; }

h1 { color: #333; }

input, button { padding: 0.5rem; margin: 0.5rem 0; font-size: 1rem; width: 200px; }

button { cursor: pointer; }

.results { margin-top: 2rem; }

.spring { background: white; padding: 1rem; border: 1px solid #ddd; border-radius: 6px; margin-bottom: 1rem; }

</style>

</head>

<body>

<h1>Garage Door Spring Calculator</h1>

<label>Door Weight (kg):<br/>

<input type="number" id="weight" placeholder="e.g. 50" />

</label><br/>

<label>Door Height (mm):<br/>

<input type="number" id="height" placeholder="e.g. 2200" />

</label><br/>

<button onclick="calculate()">Calculate Springs</button>

<div class="results" id="results"></div>

<script>

const springs = [

{ partNo: "SWsect01", color: "Brown", description: "5.6 x 750", torqueNmPerRev: 4.8, ippt: 2.7, maxHeight: 2200, weight: 24.3 },

{ partNo: "SWsect02", color: "Yellow", description: "5.6 x 635", torqueNmPerRev: 4.0, ippt: 3.3, maxHeight: 2200, weight: 29 },

{ partNo: "SWsect03", color: "Lichen", description: "6.0 x 840", torqueNmPerRev: 5.3, ippt: 3.4, maxHeight: 2400, weight: 30 },

{ partNo: "SWsect04", color: "White", description: "5.0 x 630", torqueNmPerRev: 3.2, ippt: 1.9, maxHeight: 2400, weight: 16.7 },

{ partNo: "SWsect05", color: "Red", description: "5.6 x 675", torqueNmPerRev: 4.3, ippt: 3.1, maxHeight: 2400, weight: 27 },

{ partNo: "SWsect06", color: "Lite Blue", description: "6.3 x 850", torqueNmPerRev: 6.0, ippt: 4.3, maxHeight: 2400, weight: 38 },

{ partNo: "SWsect07", color: "Blue", description: "5.6 x 895", torqueNmPerRev: 5.5, ippt: 2.3, maxHeight: 2700, weight: 20 },

{ partNo: "SWsect08", color: "Black", description: "6.3 x 920", torqueNmPerRev: 6.4, ippt: 4.0, maxHeight: 2700, weight: 35 },

{ partNo: "SWsect09", color: "Orange", description: "7.1 x 1150", torqueNmPerRev: 9.5, ippt: 5.7, maxHeight: 2700, weight: 50.3 },

{ partNo: "SWsect10", color: "Pink", description: "5.6 x 830", torqueNmPerRev: 5.1, ippt: 2.5, maxHeight: 3100, weight: 22 },

{ partNo: "SWsect11", color: "Purple", description: "6.0 x 1000", torqueNmPerRev: 6.6, ippt: 2.8, maxHeight: 3100, weight: 25 },

{ partNo: "SWsect12", color: "Green", description: "6.0 x 935", torqueNmPerRev: 6.2, ippt: 3.1, maxHeight: 3100, weight: 27 },

{ partNo: "SWsect13", color: "Red/Black", description: "5.0 x 655", torqueNmPerRev: 3.6, ippt: 1.8, maxHeight: 3100, weight: 15 },

{ partNo: "SWsect14", color: "Yellow/Blue", description: "6.0 x 795", torqueNmPerRev: 5.8, ippt: 3.6, maxHeight: 2400, weight: 32.2 },

];

function calculate() {

const weight = parseFloat(document.getElementById("weight").value);

const height = parseFloat(document.getElementById("height").value);

const resultDiv = document.getElementById("results");

resultDiv.innerHTML = '';

if (isNaN(weight) || isNaN(height)) {

resultDiv.innerText = "Please enter valid weight and height.";

return;

}

const validSprings = springs.filter(s => s.maxHeight >= height);

const single = validSprings.find(s => s.weight >= weight);

let bestPair = null;

let minPairOver = Infinity;

for (let i = 0; i < validSprings.length; i++) {

for (let j = i; j < validSprings.length; j++) {

let total = validSprings[i].weight + validSprings[j].weight;

if (total >= weight && total < minPairOver) {

bestPair = [validSprings[i], validSprings[j]];

minPairOver = total;

}

}

}

let bestTriplet = null;

let minTripletOver = Infinity;

for (let i = 0; i < validSprings.length; i++) {

for (let j = i; j < validSprings.length; j++) {

for (let k = j; k < validSprings.length; k++) {

let total = validSprings[i].weight + validSprings[j].weight + validSprings[k].weight;

if (total >= weight && total < minTripletOver) {

bestTriplet = [validSprings[i], validSprings[j], validSprings[k]];

minTripletOver = total;

}

}

}

}

if (single) {

resultDiv.innerHTML += `<h2>Single Spring</h2>`;

resultDiv.innerHTML += springHtml(single);

}

if (bestPair) {

resultDiv.innerHTML += `<h2>Pair Spring</h2>`;

bestPair.forEach(s => resultDiv.innerHTML += springHtml(s));

resultDiv.innerHTML += `<strong>Total Capacity:</strong> ${(bestPair[0].weight + bestPair[1].weight).toFixed(1)} kg<br/><br/>`;

}

if (bestTriplet) {

resultDiv.innerHTML += `<h2>Triplet Spring</h2>`;

bestTriplet.forEach(s => resultDiv.innerHTML += springHtml(s));

resultDiv.innerHTML += `<strong>Total Capacity:</strong> ${(bestTriplet[0].weight + bestTriplet[1].weight + bestTriplet[2].weight).toFixed(1)} kg<br/>`;

}

if (!single && !bestPair && !bestTriplet) {

resultDiv.innerHTML = `<p>No suitable springs found for this height and weight.</p>`;

}

}

function springHtml(s) {

return `<div class="spring">

<strong>Part:</strong> ${s.partNo}<br/>

<strong>Color:</strong> ${s.color}<br/>

<strong>Description:</strong> ${s.description}<br/>

<strong>Torque:</strong> ${s.torqueNmPerRev} Nm/Rev<br/>

<strong>IPPT:</strong> ${s.ippt}<br/>

<strong>Max Height:</strong> ${s.maxHeight} mm<br/>

<strong>Rated Weight:</strong> ${s.weight} kg

</div>`;

}

</script>

</body>

</html>