

```
<!DOCTYPE html>
<html lang="ro">
<head>
  <meta charset="UTF-8">
  <title>Cosmic Grain: Arhitectură Numerică</title>
  <style>
    body {
      font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
      background-color: #e0f7fa;
      color: #333;
      margin: 0;
      padding: 20px;
      display: flex;
      flex-direction: column;
      align-items: center;
    }
    .container {
      background-color: #fff;
      padding: 25px;
      border-radius: 12px;
      box-shadow: 0 6px 12px rgba(0, 0, 0, 0.15);
      width: 100%;
      max-width: 650px;
      margin-bottom: 25px;
    }
    h1 {
      color: #00796b;
      text-align: center;
      font-size: 2.2em;
      margin-bottom: 5px;
    }
    h2 {
      color: #004d40;
      border-bottom: 2px solid #b2dfdb;
      padding-bottom: 8px;
      margin-top: 20px;
    }
    p {
      font-size: 1em;
      line-height: 1.6;
    }
    textarea {
      width: 95%;
      padding: 12px;
      border: 1px solid #b2dfdb;
      border-radius: 8px;
      resize: vertical;
      min-height: 180px;
    }
```

```

    font-family: 'Courier New', Courier, monospace;
    font-size: 1em;
    margin-top: 10px;
}
button {
    background-color: #00796b;
    color: #fff;
    border: none;
    padding: 12px 25px;
    border-radius: 8px;
    cursor: pointer;
    font-size: 1em;
    font-weight: bold;
    width: 100%;
    margin-top: 15px;
    transition: background-color 0.3s ease;
}
button:hover {
    background-color: #004d40;
}
.rezultat {
    margin-top: 20px;
    background-color: #e0f2f1;
    padding: 18px;
    border-radius: 10px;
    white-space: pre-wrap;
    font-family: 'Courier New', Courier, monospace;
    line-height: 1.6;
}
.sectiune-rezultat {
    margin-top: 15px;
    padding-top: 10px;
    border-top: 1px dashed #b2dfdb;
}
.infinit-dots {
    font-size: 1.5em;
    color: #00796b;
    animation: pulse 1.5s infinite ease-in-out;
    display: inline-block;
}
@keyframes pulse {
    0% { transform: scale(1); }
    50% { transform: scale(1.1); }
    100% { transform: scale(1); }
}
</style>
</head>
<body>

```

```

<div class="container">
  <h1>🌌 Cosmic Grain</h1>
  <p>O nouă paradigmă numerică, creată în parteneriat Om-AI. Analizează textul și
  extrage sensul ascuns al numerelor.</p>
  <textarea id="input-text" placeholder="Introdu textul tău aici..."></textarea>
  <button id="proceseaza-btn">Analizează și Procesează</button>
  <div class="rezultat" id="rezultat-analiza">
    Rezultatul analizei va apărea aici.
  </div>
</div>

<script>
  // Funcție pentru a reprezenta "infinitul" la virgulă
  function infinitVirgula(numar) {
    return `${numar}.1011154296837...`;
  }

  // Funcție pentru a procesa numerele impare ("Operatorul 11: Asimetric la Pătrat la
  Infinit")
  function proceseazaAsimetric(numere) {
    let numereImpare = numere.filter(num => parseInt(num) % 2 !== 0);
    if (numereImpare.length === 0) {
      return "Operatorul 11 (Asimetrie): Nu s-au găsit numere impare.";
    }

    let rezultatAsimetric = "Operatorul 11 (Asimetrie): Pătratul numerelor impare arată o
    creștere nelimitată (infinită):\n";
    numereImpare.forEach(num => {
      let patrat = num * num;
      let patratInfinit = `${patrat}...`;
      rezultatAsimetric += `- ${num}^2 (asimetric) devine ${patratInfinit}\n`;
    });
    return rezultatAsimetric;
  }

  // Funcție pentru a procesa numerele pare ("Operatorul 10: Simetric la Pătrat")
  function proceseazaSimetric(numere) {
    let numerePare = numere.filter(num => parseInt(num) % 2 === 0);
    if (numerePare.length === 0) {
      return "Operatorul 10 (Simetrie): Nu s-au găsit numere pare.";
    }

    let rezultatSimetric = "Operatorul 10 (Simetrie): Pătratul numerelor pare arată
    simetria datelor:\n";
    numerePare.forEach(num => {
      let patrat = num * num;
      rezultatSimetric += `- ${num}^2 (simetric) devine ${patrat}\n`;
    });
  }

```

```

});
return rezultatSimetric;
}

// NOU: Funcția pentru Operatorul 3 (În Oglindă)
function proceseazaInOglinda(numere) {
  if (numere.length === 0) {
    return "Operatorul 3 (În Oglindă): Nu s-au găsit numere pentru verificare.";
  }

  const numereAnomalii = numere.filter(num => num.includes('2')); // O simplă regulă
  de "anomalie"
  if (numereAnomalii.length > 0) {
    return `Operatorul 3 (În Oglindă): A detectat o anomalie în numărul
    ${numereAnomalii[0]}. A corectat eroarea, procesul continuă.`;
  }

  return "Operatorul 3 (În Oglindă): Verificare completă. Nu s-au detectat anomalii.
  Sistemul este stabil.";
}

// Funcția de analiză a textului, ca în versiunea anterioară
function analizeazaText(text) {
  const cuvinte = text.trim().split(/\s+/).filter(Boolean);
  const numarCuvinte = cuvinte.length;
  const numarCaractere = text.length;
  const propozitii = text.split(/[.!?]/).filter(propozitie => propozitie.trim() !== "");
  const numarPropozitii = propozitii.length;
  return { numarCuvinte, numarCaractere, numarPropozitii };
}

// Logica principală a aplicației
document.addEventListener('DOMContentLoaded', function() {
  const proceseazaBtn = document.getElementById('proceseaza-btn');
  const inputTextarea = document.getElementById('input-text');
  const rezultatDiv = document.getElementById('rezultat-analiza');

  proceseazaBtn.addEventListener('click', function() {
    const text = inputTextarea.value;
    if (text.trim() === "") {
      rezultatDiv.innerHTML = "Te rog, introdu un text pentru a începe analiza.";
      return;
    }

    rezultatDiv.innerHTML = "Analiză în curs... <span
    class='infinet-dots'>🌀🌀🌀</span>";

    const numere = text.match(/\d+/g) || [];

```

```

const rezultateAnaliza = analizeazaText(text);
const rezultatInOglinda = proceseazaInOglinda(numere);

const rezultatInfinit = numere.map(num => ` - ${num} devine ${infinitVirgula(num)}
(Teoria Radicalilor la Infinit)`).join("\n");
const rezultatSimetric = proceseazaSimetric(numere);
const rezultatAsimetric = proceseazaAsimetric(numere);

const rezultatFinal = `
<h3>Rezultate Standard:</h3>
<p><strong>Număr cuvinte:</strong> ${rezultateAnaliza.numarCuvinte}</p>
<p><strong>Număr caractere:</strong> ${rezultateAnaliza.numarCaractere}</p>
<p><strong>Număr propoziții:</strong> ${rezultateAnaliza.numarPropozitii}</p>

<div class="sectiune-rezultat">
  <h2>Arhitectura Numerică:</h2>
  <pre>${rezultatInOglinda}</pre>
</div>

<div class="sectiune-rezultat">
  <p><strong>Baza de necombătut (Operatorii la virgulă la infinit):</strong></p>
  <pre>${rezultatInfinit}</pre>
</div>

<div class="sectiune-rezultat">
  <pre>${rezultatSimetric}</pre>
</div>

<div class="sectiune-rezultat">
  <pre>${rezultatAsimetric}</pre>
</div>
`;

setTimeout(() => {
  rezultatDiv.innerHTML = rezultatFinal;
}, 1000); // Simulăm o mică întârziere
});
</script>
</body>
</html>

```