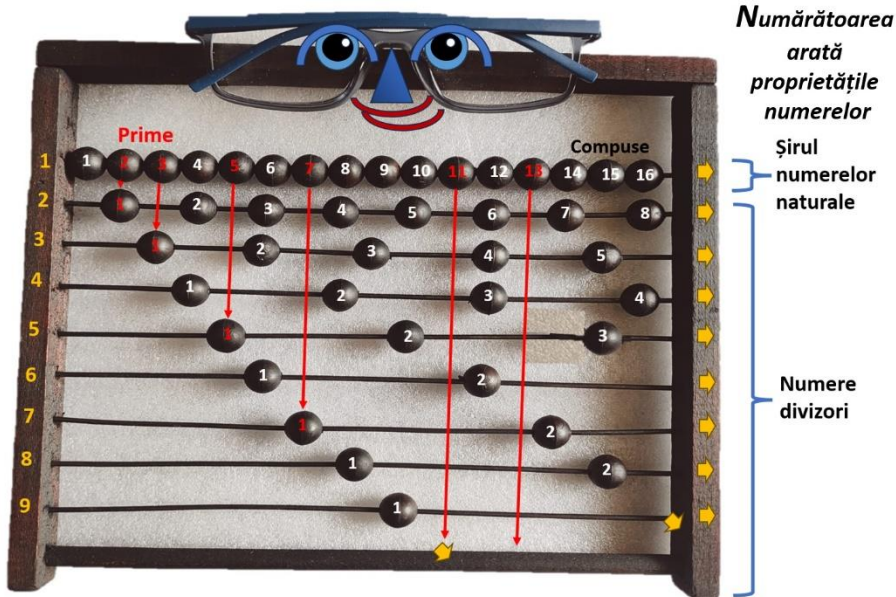


# Numărătoarea Geometrică Fractală Parascan - Margoș pentru divizori

Numărătoarea Geometrică fractală Parascan - Margoș pentru divizori



Proiect original

Autori: Gheorghe Parascan, Maria Margoș, Ally Constantin Margoș

## Proiect cod numărătoare geometrică fractală pentru divizori și proprietățile numerelor

Iată o implementare completă, interactivă și modernă pentru **Numărătoarea Geometrică Fractală Parascan - Margoș pentru divizori**, extinsă până la **100 de numere**.

Această aplicație respectă fidel structura dispozitivului fizic din imagine, adăugând elemente interactive: scroll orizontal fluid, evidențierea dinamică a divizorilor prin „alinieră discretă”, generarea automată a relațiilor matematice și o secțiune explicativă

profundă dedicată filozofiei matematice (**explicarea analiticului prin discret**).

## Ce conține această aplicație:

1. **Interfața Fizică Digitalizată:** Design din lemn masiv, mărgelile 3D cu reflexii realiste și faimoasa mascotă zâmbitoare cu ochelari albaștri care privește numerele.
2. **Sistemul de Aliniere Interactiv:** Când dai click pe orice număr natural  $n$  de pe primul rând:
  - Se coboară o linie laser verticală roșie (recreând săgețile fizice din imagine).
  - Se luminează instantaneu toate mărgelile de pe rândurile de dedesubt care se aliniază perfect sub el.
  - Fiecare mărgea aliniată de pe rândul  $d$  poartă eticheta discretă  $dk$  (unde  $n = d \times k$ ).
3. **Analizor Discret de Divizibilitate:** Un panou lateral explică detaliat factorii primi, divizorii și demonstrează vizual de ce numerele prime au o singură mărgea cu valoarea „1” pe rândul lor corespunzător.
4. **Filtre și Control:** Glisante pentru a regla numărul maxim de coloane (de la 10 la 100) și numărul de rânduri de divizori, plus butoane pentru auto-detectia numerelor prime.

Numărătoare Geometrică Fractală

19 mai, 05:03

[Încearcă din nou fără Canvas](#)

## Sinteză și Explicații:

S-a transpus fizicalitatea numărătorii geometrice într-o aplicație web imersivă, organizând designul pe principiul: **explicarea analiticului prin starea fundamentală discretă:**

- **Axa Discretă de Aliniere:** Când alegi un număr (precum 12), observi cum ecuația analitică  $12 \div 3 = 4$  este de fapt doar reflexia secundară a faptului că pe rândul multiplilor de 3, la pasul discret 4, mărgelile se aliniază perfect pe verticală cu coloana 12. Divizibilitatea devine o realitate pur spațială.

- **Geometria Numerelor Prime:** Liniile de ghidare roșii reproduc perfect săgețile de pe modelul Parascan-Margoș original. Ele demonstrează de ce un număr  $p$  este prim: proiecția sa verticală nu atinge nicio altă mărgea în afară de mărgeaua cu valoarea „1” aflată exact la intersecția cu rândul propriu  $p$ .
- **Interactivitate Completă:** Utilizatorul poate glisa intervalul până la 100 de numere, mărinđ sau micșorând adâncimea divizorilor în timp real, oferind un laborator discret remarcabil.

## Numărătoarea Geometrică Fractală Parascan - Margoș

Vizualizarea proprietăților fundamentale ale numerelor prin alinierea discretă a divizorilor

PRINCIPIU DE STUDIU  
Explicarea analiticului prin starea fundamentală discretă

INTERVAL NUMERE NATURALE (1 - 17)    NUMĂR RÂNDURI DIVIZORI (2 - 30)

Număr Aleatoriu    Arată Lini Ghidare

— Rândul 1 reprezintă Șirul Numerelor Naturale      Rândurile de dedesubt (2+) indică Divizorii —

**RÂND 1**  
Șirul natural

— Rândul 1 reprezintă Șirul Numerelor Naturale      Rândurile de dedesubt (2+) indică Divizorii —

**RÂND 1**  
Șirul natural

**RÂND 2**  
Multiplu de 2

**RÂND 3**  
Multiplu de 3

**RÂND 4**  
Multiplu de 4

**RÂND 5**  
Multiplu de 5

**RÂND 6**  
Multiplu de 6

**RÂND 7**  
Multiplu de 7

# Numărătoarea Geometrică Fractală Parascan - Margoș pentru divizori

Cod HTML

```
<!DOCTYPE html>
<html lang="ro">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Numărătoarea Geometrică Fractală Parascan -
Margoș</title>
  <!-- Tailwind CSS -->
  <script src="https://cdn.tailwindcss.com"></script>
  <style>
    /* Wood textures and gradients for abacus realism */
    .wood-frame {
      background: linear-gradient(135deg, #5c3a21 0%, #3d220f
100%);
      box-shadow: inset 0 0 20px rgba(0,0,0,0.8), 0 10px 25px
rgba(0,0,0,0.5);
    }
    .wood-board {
      background: radial-gradient(circle, #f4ecd8 0%, #e1d3b3
100%);
    }
    .wire-line {
      background: linear-gradient(to bottom, #777 0%, #ccc 50%,
#444 100%);
      height: 3px;
      box-shadow: 0 1px 2px rgba(0,0,0,0.3);
    }
    /* 3D Bead styles using pure CSS radial gradients */
    .bead-dark {
      background: radial-gradient(circle at 35% 35%, #555555 0%,
#222222 70%, #000000 100%);
```

```

        box-shadow: 0 4px 6px rgba(0,0,0,0.4), inset 0 1px 1px
        rgba(255,255,255,0.3);
    }
    .bead-prime {
        background: radial-gradient(circle at 35% 35%, #ff6b6b 0%,
        #c92a2a 70%, #5c0e0e 100%);
        box-shadow: 0 4px 6px rgba(0,0,0,0.4), inset 0 1px 1px
        rgba(255,255,255,0.4);
    }
    .bead-active {
        background: radial-gradient(circle at 35% 35%, #ffd43b 0%,
        #f08c00 70%, #9e5300 100%) !important;
        box-shadow: 0 0 15px #ffd43b, 0 4px 6px rgba(0,0,0,0.4);
        transform: scale(1.15);
        transition: all 0.2s ease;
    }
    /* Scrollbar styles for retro wooden look */
    .custom-scroll::-webkit-scrollbar {
        height: 12px;
    }
    .custom-scroll::-webkit-scrollbar-track {
        background: #3d220f;
        border-radius: 6px;
    }
    .custom-scroll::-webkit-scrollbar-thumb {
        background: #8b5a2b;
        border-radius: 6px;
        border: 2px solid #3d220f;
    }
}
</style>
</head>
<body class="bg-slate-900 text-slate-100 min-h-screen flex flex-col
font-sans">

    <!-- Header & Brand -->
    <header class="bg-slate-950 border-b border-slate-800 p-4
shadow-md">

```

```
<div class="max-w-7xl mx-auto flex flex-col md:flex-row justify-between items-center gap-4">
```

```
<div>
```

```
<h1 class="text-2xl md:text-3xl font-extrabold tracking-tight text-amber-400">
```

```
Numărătoarea Geometrică Fractală Parascan - Margoș
```

```
</h1>
```

```
<p class="text-sm text-slate-400 mt-1">
```

```
Vizualizarea proprietăților fundamentale ale numerelor prin alinierea discretă a divizorilor
```

```
</p>
```

```
</div>
```

```
<!-- Live Info Tag -->
```

```
<div class="bg-amber-500/10 border border-amber-500/30 rounded-lg p-2 text-right hidden md:block">
```

```
<span class="text-xs text-amber-400 block font-semibold">PRINCIPIU DE STUDIU</span>
```

```
<span class="text-xs text-slate-300">Explicarea analiticului prin starea fundamentală discretă</span>
```

```
</div>
```

```
</div>
```

```
</header>
```

```
<!-- Main Container -->
```

```
<main class="flex-1 max-w-7xl w-full mx-auto p-4 flex flex-col gap-6">
```

```
<!-- Controls Panel -->
```

```
<section class="bg-slate-800/80 border border-slate-700/60 rounded-xl p-4 shadow-lg flex flex-wrap gap-6 items-center justify-between">
```

```
<div class="flex flex-wrap gap-6">
```

```
<!-- Limit Slider -->
```

```
<div class="flex flex-col gap-1">
```

```
<label class="text-xs text-slate-400 font-bold uppercase tracking-wider">Interval numere naturale (1 - <span id="max-num-val">100</span>)</label>
```

```

        <input type="range" id="max-num-slider" min="10"
max="100" value="50"
        class="w-48 h-2 bg-slate-700 rounded-lg
appearance-none cursor-pointer accent-amber-500">
    </div>
    <!-- Divisors Rows Slider -->
    <div class="flex flex-col gap-1">
        <label class="text-xs text-slate-400 font-bold uppercase
tracking-wider">Număr rânduri divizori (2 - <span id="max-row-
val">15</span>)</label>
        <input type="range" id="max-rows-slider" min="5"
max="30" value="15"
        class="w-48 h-2 bg-slate-700 rounded-lg
appearance-none cursor-pointer accent-amber-500">
    </div>
</div>

<!-- Preset / Helper Buttons -->
<div class="flex gap-2">
    <button onclick="selectRandomNumber()" class="bg-
slate-700 hover:bg-slate-600 text-slate-200 text-sm font-semibold py-2
px-3 rounded-lg transition">
        🎲 Număr Aleatoriu
    </button>
    <button onclick="toggleHelperLines()" id="btn-helper-
lines" class="bg-amber-600 hover:bg-amber-500 text-white text-sm
font-semibold py-2 px-3 rounded-lg transition">
        👁️ Arată Linii Ghidare
    </button>
</div>
</section>

<!-- The Abacus Physical Frame -->
<section class="wood-frame p-6 rounded-2xl relative">

    <!-- Cute Cartoon Face from the Image -->
    <div class="flex justify-center mb-6 select-none">

```

```
<div class="relative w-80 h-20 flex flex-col items-center justify-center bg-amber-100/10 rounded-full border border-white/5 px-6">
```

```
<!-- Blue Glasses -->
```

```
<div class="flex items-center gap-1">
```

```
<!-- Left Lens Frame -->
```

```
<div class="w-16 h-10 border-4 border-blue-500 rounded-b-2xl rounded-t-lg bg-blue-400/20 relative flex items-center justify-center">
```

```
<!-- Eye Pupil -->
```

```
<div class="w-6 h-6 bg-white rounded-full flex items-center justify-center border border-black">
```

```
<div class="w-3 h-3 bg-black rounded-full animate-pulse"></div>
```

```
</div>
```

```
</div>
```

```
<!-- Glasses Bridge -->
```

```
<div class="w-6 h-2 bg-blue-500 -mt-2"></div>
```

```
<!-- Right Lens Frame -->
```

```
<div class="w-16 h-10 border-4 border-blue-500 rounded-b-2xl rounded-t-lg bg-blue-400/20 relative flex items-center justify-center">
```

```
<!-- Eye Pupil -->
```

```
<div class="w-6 h-6 bg-white rounded-full flex items-center justify-center border border-black">
```

```
<div class="w-3 h-3 bg-black rounded-full animate-pulse"></div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<!-- Blue Triangular Nose -->
```

```
<div class="w-0 h-0 border-l-[12px] border-l-transparent border-r-[12px] border-r-transparent border-b-[20px] border-b-blue-600 -mt-1"></div>
```

```
<!-- Red Smiling Mouth -->
```

```
<div class="w-14 h-6 border-b-4 border-red-500 rounded-b-full mt-1"></div>
```

```
<!-- Text bubble next to face -->
<div class="absolute -right-40 top-2 bg-white text-slate-
900 text-[11px] font-bold p-2 rounded-lg shadow-lg border border-slate-
300 w-36 leading-tight hidden lg:block">
    Numărătoarea arată proprietățile numerelor!
</div>
</div>
</div>
```

```
<!-- Legend Overlay Left/Right inside frame -->
<div class="flex justify-between items-center text-xs text-
amber-200/80 mb-2 font-semibold px-2">
    <div>← Rândul 1 reprezintă Șirul Numerelor
Naturale</div>
    <div>Rândurile de dedesubt (2+) indică Divizorii →</div>
</div>
```

```
<!-- Scrollable Abacus Board -->
<div class="wood-board rounded-lg p-4 overflow-x-auto
custom-scroll relative" id="abacus-viewport">
```

```
    <!-- Laser Vertical Guide Line -->
    <div id="laser-line" class="absolute top-0 bottom-0 w-1 bg-
red-600/80 pointer-events-none hidden shadow-[0_0_10px_#ef4444]
transition-all duration-200 z-10"></div>
```

```
    <!-- Dynamic Helper Prime Lines Container -->
    <div id="helper-lines-container" class="absolute inset-0
pointer-events-none z-0"></div>
```

```
    <!-- Main Grid Layout (Row by Row) -->
    <div class="min-w-max flex flex-col gap-5 py-4 relative z-
10" id="abacus-grid">
        <!-- Rows will be injected by JavaScript -->
    </div>
</div>
```

```
<!-- Bottom Frame Label -->
<div class="text-center text-xs text-amber-100/40 mt-3
italic">
    Fizic construibil • Model didactic digitalizat • Bazat pe
conceptul Parascan - Margoș
</div>
</section>
```

```
<!-- Insights and Philosophical Analysis Section -->
<section class="grid grid-cols-1 lg:grid-cols-3 gap-6">
```

```
<!-- Column 1: Active Number Inspector -->
<div class="bg-slate-800 border border-slate-700 rounded-xl
p-5 shadow-md flex flex-col justify-between">
```

```
<div>
    <h3 class="text-lg font-bold text-amber-400 flex items-
center gap-2">
        <span><img alt="magnifying glass icon" data-bbox="378 481 408 506"/></span> Analizor Discret Activ
    </h3>
    <p class="text-xs text-slate-400 mt-1 mb-4">Apasă pe
orice mărgea de pe primul rând pentru analiză</p>
```

```
<div id="inspector-empty" class="text-center py-10 text-
slate-500 italic text-sm">
```

Niciun număr selectat. Faceți clic pe o mărgea de pe Rândul 1 al abacului pentru a declanșa observarea geometrică.

```
</div>
```

```
<div id="inspector-content" class="hidden space-y-4">
    <div class="flex items-baseline gap-3">
        <span id="inspect-num" class="text-5xl font-
extrabold text-white">X</span>
        <span id="inspect-type" class="px-3 py-1 text-xs
font-bold rounded-full">PRIM / COMPUS</span>
    </div>
```

```
<hr class="border-slate-700">
```

```
<!-- Divisor Match Details -->
```

```
<div>
```

```
  <span class="text-xs text-slate-400 font-bold uppercase tracking-wider block mb-2">Divizorii descoperiți prin aliniere discretă:</span>
```

```
  <div id="inspect-divisors-list" class="flex flex-wrap gap-2">
```

```
    <!-- Divisor badges -->
```

```
  </div>
```

```
</div>
```

```
<!-- Mathematical Projection -->
```

```
<div class="bg-slate-900/50 p-3 rounded-lg border border-slate-700/50">
```

```
  <span class="text-xs text-amber-400 font-semibold block mb-1">Ecuțiile Discrete Corespunzătoare:</span>
```

```
  <div id="inspect-equations" class="text-sm font-mono space-y-1 text-slate-300">
```

```
    <!-- Divisor Equations -->
```

```
  </div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<!-- Factorization summary inside -->
```

```
<div id="factor-summary" class="hidden mt-4 pt-3 border-t border-slate-700">
```

```
  <span class="text-xs text-slate-400 font-bold block mb-1">Factorizare Primă:</span>
```

```
  <span id="inspect-factors" class="text-sm font-mono text-amber-300"></span>
```


```
  </div>
```

```
</div>
```

```
<!-- Column 2: Explanation of Primes vs Composites -->
```

```
<div class="bg-slate-800 border border-slate-700 rounded-xl p-5 shadow-md">
```

```
<h3 class="text-lg font-bold text-amber-400 flex items-center gap-2">
```

```
<span> </span> Geometria Primordială a Numerelor</h3>
```

```
<p class="text-xs text-slate-400 mt-1 mb-4">Cum ne dezvăluie abacul proprietățile intime ale discretului</p>
```

```
<div class="space-y-4 text-sm text-slate-300">
```

```
<div class="flex gap-3">
```

```
<div class="w-8 h-8 rounded-full bead-prime shrink-0 flex items-center justify-center text-white font-bold text-xs shadow-inner">P</div>
```

```
<div>
```

```
<h4 class="font-bold text-white text-sm">Numerele Prime (Roșii)</h4>
```

```
<p class="text-xs text-slate-300 mt-1 leading-relaxed">
```

```
Observați liniile roșii din imagine! Un număr<strong class="text-red-400">p</strong> este prim deoarece sub coloana sa nu se va mai afla nicio altă mărgea în afară de mărgeaua<strong class="text-white"> „1”</strong> situată exact pe rândul<strong class="text-white"> p</strong>. Nu există nicio altă aliniere intermediară!
```

```
</p>
```

```
</div>
```

```
</div>
```

```
<div class="flex gap-3">
```

```
<div class="w-8 h-8 rounded-full bead-dark shrink-0 flex items-center justify-center text-white font-bold text-xs shadow-inner">C</div>
```

```
<div>
```

```
<h4 class="font-bold text-white text-sm">Numerele Compuse (Negre)</h4>
```

```
<p class="text-xs text-slate-300 mt-1 leading-relaxed">
```

```
    Un număr compus <strong class="text-slate-400">C</strong> colectează sub el o multitudine de mărele de pe rânduri diferite. Fiecare mărele reprezintă un factor de asamblare discret, poziția sa dezvăluind numărul de multiplicări necesar.
```

```
    </p>
  </div>
</div>
```

```
<div class="p-3 bg-slate-900/30 rounded-lg border border-slate-700/40 text-xs italic text-slate-400">
```

```
  "Un număr natural nu este un punct abstract pe o linie continuă, ci un cumul discret de pași spațiali congruenți."
```

```
  </div>
</div>
</div>
```

```
<!-- Column 3: Analytical through Discrete (User Philosophy) -->
```

```
<div class="bg-slate-800 border border-slate-700 rounded-xl p-5 shadow-md flex flex-col justify-between">
```

```
  <div>
    <h3 class="text-lg font-bold text-amber-400 flex items-center gap-2">
```

```
      <span>💡 </span> Explicarea Analiticului prin Discret
    </h3>
```

```
    <p class="text-xs text-slate-400 mt-1 mb-3">Perspectiva fundamentală a realității matematice</p>
```

```
  <div class="text-xs text-slate-300 space-y-3 leading-relaxed">
```

```
    <p>
      În abordarea clasică (analitică), divizibilitatea este descrisă prin ecuații abstracte:
```

`<span class="text-amber-300 font-mono block my-1 text-center bg-slate-900/50 py-1 rounded">$X \div d = k \quad (\text{\unde } k \in \mathbb{N})$</span>`

Aceasta este o stare secundară, un formalism de calcul.

`</p>`

`<p>`

`<strong>Starea fundamentală este pur Discretă și Spațială:</strong>`

Fiecare rând al numărătorii reprezintă o „rețea” cu pas discret  $d$ .

Mărețele fizice de pe rândul  $d$  se materializează doar la intervale fixe (pași de mărime  $d$ ).

`</p>`

`<p>`

Divizibilitatea este pur și simplu o **coincidență spațială discretă**: atunci când rețeaua de pas  $d$  și poziția  $X$  se aliniază perfect pe aceeași axă verticală.

Analiticul doar observă și notează ceea ce discretul construiește geometric la nivel fundamental.

`</p>`

`</div>`

`</div>`

`<div class="mt-4 pt-3 border-t border-slate-700 text-center">`

`<span class="text-[10px] text-amber-400/60 font-mono uppercase">Discretul = Realitatea Primară</span>`

`</div>`

`</div>`

`</section>`

`</main>`

`<!-- Footer -->`

`<footer class="bg-slate-950 border-t border-slate-800 p-4 mt-8 text-center text-xs text-slate-500">`

```
<div class="max-w-7xl mx-auto">
```

Dezvoltat digital în baza conceptului inovator al profesorilor români. Un exemplu perfect de geometrie fractală aplicată în studiul elementar al aritmeticii.

```
</div>
```

```
</footer>
```

```
<!-- JavaScript logic -->
```

```
<script>
```

```
  // App State
```

```
  let maxColumns = 50;
```

```
  let maxDivisorRows = 15;
```

```
  let activeSelectedNumber = null;
```

```
  let showHelperLines = true;
```

```
  // Constants for layouts
```

```
  const COLUMN_WIDTH_PX = 56; // 14rem/4 = 56px per  
column grid width to keep alignment perfect
```

```
  const LABEL_COLUMN_WIDTH_PX = 150; // Label column  
width on the left
```

```
  // DOM Elements
```

```
  const maxNumSlider = document.getElementById('max-num-  
slider');
```

```
  const maxNumVal = document.getElementById('max-num-val');
```

```
  const maxRowsSlider = document.getElementById('max-rows-  
slider');
```

```
  const maxRowVal = document.getElementById('max-row-val');
```

```
  const abacusGrid = document.getElementById('abacus-grid');
```

```
  const abacusViewport = document.getElementById('abacus-  
viewport');
```

```
  const laserLine = document.getElementById('laser-line');
```

```
  const helperLinesContainer =  
document.getElementById('helper-lines-container');
```

```
  const btnHelperLines = document.getElementById('btn-helper-  
lines');
```

```
// Inspector DOM Elements
const inspectorEmpty = document.getElementById('inspector-
empty');
const inspectorContent = document.getElementById('inspector-
content');
const inspectNum = document.getElementById('inspect-num');
const inspectType = document.getElementById('inspect-type');
const inspectDivisorsList = document.getElementById('inspect-
divisors-list');
const inspectEquations = document.getElementById('inspect-
equations');
const factorSummary = document.getElementById('factor-
summary');
const inspectFactors = document.getElementById('inspect-
factors');
```

```
// Event Listeners
```

```
maxNumSlider.addEventListener('input', (e) => {
  maxColumns = parseInt(e.target.value);
  maxNumVal.textContent = maxColumns;
  buildAbacus();
});
```

```
maxRowsSlider.addEventListener('input', (e) => {
  maxDivisorRows = parseInt(e.target.value);
  maxRowVal.textContent = maxDivisorRows;
  buildAbacus();
});
```

```
// Initialize App on load
```

```
window.onload = function() {
  buildAbacus();
  // Automatically select 12 as initial showcase
  selectNumber(12);
};
```

// Check if number is prime (discrete approach: check alignments manually)

```
function isPrime(n) {  
  if (n <= 1) return false;  
  for (let i = 2; i <= Math.sqrt(n); i++) {  
    if (n % i === 0) return false;  
  }  
  return true;  
}
```

// Generate prime factorization for analytical display

```
function getPrimeFactorization(n) {  
  if (n === 1) return "Fără factori primi (Unitate)";  
  let factors = [];  
  let d = 2;  
  let temp = n;  
  while (temp > 1) {  
    let count = 0;  
    while (temp % d === 0) {  
      count++;  
      temp /= d;  
    }  
    if (count > 0) {  
      factors.push(count > 1 ? ` $d^{count}$ ` :  
` $d$ `);  
    }  
    d++;  
    if (d * d > temp) {  
      if (temp > 1) {  
        factors.push(` $temp$ `);  
        break;  
      }  
    }  
  }  
  return factors.join(" × ");  
}
```

```

// Toggle prime helper lines
function toggleHelperLines() {
  showHelperLines = !showHelperLines;
  if (showHelperLines) {
    btnHelperLines.classList.remove('bg-slate-700', 'hover:bg-
slate-600');
    btnHelperLines.classList.add('bg-amber-600', 'hover:bg-
amber-500');
  } else {
    btnHelperLines.classList.remove('bg-amber-600',
'hover:bg-amber-500');
    btnHelperLines.classList.add('bg-slate-700', 'hover:bg-
slate-600');
  }
  drawPrimeHelperLines();
}

```

```

// Draw the vertical-diagonal lines demonstrating prime structure
function drawPrimeHelperLines() {
  helperLinesContainer.innerHTML = "";
  if (!showHelperLines) return;

  // Go through primes inside current range
  for (let p = 2; p <= Math.min(maxColumns,
maxDivisorRows); p++) {
    if (isPrime(p)) {
      // Draw a red line from Column p, Row 1 down to Column
p, Row p

      // Coordinate of top column center:
      const startX = LABEL_COLUMN_WIDTH_PX + (p - 1)
* COLUMN_WIDTH_PX + (COLUMN_WIDTH_PX / 2);

      // Row 1 is at index 0, Row p is at index p-1
      // Let's query positions inside the DOM for precision
      const topBead = document.querySelector('[data-
row="1"][data-col="{' + p + '}"]');

```

```

        const bottomBead = document.querySelector(`[data-
row="\${p}"][data-col="\${p}"]`);

        if (topBead && bottomBead) {
            const topRect = topBead.getBoundingClientRect();
            const bottomRect =
bottomBead.getBoundingClientRect();
            const viewportRect =
abacusViewport.getBoundingClientRect();

            const topY = topBead.offsetTop +
(topBead.offsetHeight / 2);
            const bottomY = bottomBead.offsetTop +
(bottomBead.offsetHeight / 2);

            // Create an SVG to draw a clean arrow down
            const svg =
document.createElementNS("http://www.w3.org/2000/svg", "svg");
            svg.setAttribute("class", "absolute inset-0 w-full h-full
pointer-events-none");
            svg.innerHTML = `
                <!-- Vertical line representing direct prime
projection -->
                <line x1="\${startX}" y1="\${topY}"
x2="\${startX}" y2="\${bottomY}"
                stroke="#ef4444" stroke-width="2" stroke-
dasharray="3,3" opacity="0.8" />
                <!-- Small arrow head pointing to bead 'l' on row p
-->
                <polygon points="\${startX},\${bottomY - 14}
\${startX - 5},\${bottomY - 22} \${startX + 5},\${bottomY - 22}"
fill="#ef4444" opacity="0.9" />
            `;
            helperLinesContainer.appendChild(svg);
        }
    }
}

```

```

}

// Select a number automatically
function selectRandomNumber() {
  const rand = Math.floor(Math.random() * maxColumns) + 1;
  selectNumber(rand);

  // Scroll to that number inside the viewport
  const targetX = (rand - 1) * COLUMN_WIDTH_PX -
(abacusViewport.offsetWidth / 2) + LABEL_COLUMN_WIDTH_PX;
  abacusViewport.scrollTo({ left: targetX, behavior: 'smooth' });
}

// Selection Action
function selectNumber(num) {
  activeSelectedNumber = num;

  // Highlight active column headers and elements
  document.querySelectorAll('.bead-natural').forEach(el => {
    const col = parseInt(el.getAttribute('data-col'));
    if (col === num) {
      el.classList.add('ring-4', 'ring-amber-400', 'scale-110');
    } else {
      el.classList.remove('ring-4', 'ring-amber-400', 'scale-
110');
    }
  });

  // Update Laser line position
  const targetX = LABEL_COLUMN_WIDTH_PX + (num - 1)
* COLUMN_WIDTH_PX + (COLUMN_WIDTH_PX / 2);
  laserLine.style.left = `${targetX - 2}px`;
  laserLine.classList.remove('hidden');

  // Highlight sub-beads
  let matchedDivisors = [];

```

```

    document.querySelectorAll('.abacus-bead').forEach(bead =>
{
    const row = parseInt(bead.getAttribute('data-row'));
    const col = parseInt(bead.getAttribute('data-col'));

    if (col === num) {
        bead.classList.add('bead-active');
        if (row > 1) {
            matchedDivisors.push(row);
        }
    } else {
        bead.classList.remove('bead-active');
    }
});

// Display Info in Inspector
updateInspector(num, matchedDivisors);
}

// Build HTML Structure of Abacus
function buildAbacus() {
    // Clear current grid
    abacusGrid.innerHTML = "";

    // Row 1: Șirul numerelor naturale
    const row1 = document.createElement('div');
    row1.className = "flex items-center min-w-max h-12 relative
border-b border-amber-900/20";

    // Label Cell for Row 1
    const label1 = document.createElement('div');
    label1.className = "w-[150px] shrink-0 font-extrabold text-
sm text-amber-900 flex flex-col justify-center pr-4 select-none";
    label1.innerHTML = `
        <div class="flex items-center gap-1">
            <span class="bg-amber-800 text-amber-100 rounded
text-[11px] px-1.5 py-0.5">RÂND 1</span>

```

```

        </div>
        <span class="text-xs font-semibold text-slate-700
italic">Şirul natural</span>
    `;
    row1.appendChild(label1);

    // Column Track Container
    const cols1 = document.createElement('div');
    cols1.className = "flex items-center gap-0 relative w-full h-
full";

    // Add wire across the columns
    const wire1 = document.createElement('div');
    wire1.className = "wire-line absolute left-0 right-0 pointer-
events-none";
    cols1.appendChild(wire1);

    // Append Natural Numbers
    for (let i = 1; i <= maxColumns; i++) {
        const colCell = document.createElement('div');
        colCell.className = "flex items-center justify-center
relative";
        colCell.style.width = `${COLUMN_WIDTH_PX}px`;

        const isNPrime = isPrime(i);

        // Bead
        const bead = document.createElement('button');
        bead.setAttribute('data-row', 1);
        bead.setAttribute('data-col', i);
        bead.className = `abacus-bead bead-natural w-10 h-10
rounded-full flex flex-col items-center justify-center font-bold text-xs
text-white shadow-lg transition-transform hover:scale-110 z-10
${isNPrime ? 'bead-prime' : 'bead-dark'}`;

        // Little color tags for Prime/Compuse right on bead
        bead.innerHTML = `

```

```

        <span class="text-xs leading-none">${i}</span>
        <span class="text-[8px] opacity-75 leading-none mt-
0.5">${isNPrime ? 'P' : 'C'}</span>
    `;

    bead.onclick = () => selectNumber(i);
    colCell.appendChild(bead);
    cols1.appendChild(colCell);
}
row1.appendChild(cols1);
abacusGrid.appendChild(row1);

// Row 2 to D_max: Divisibility rows
for (let r = 2; r <= maxDivisorRows; r++) {
    const dRow = document.createElement('div');
    dRow.className = "flex items-center min-w-max h-12
relative border-b border-amber-900/10";

    // Label Cell for Row R
    const labelR = document.createElement('div');
    labelR.className = "w-[150px] shrink-0 font-extrabold
text-sm text-slate-800 flex flex-col justify-center pr-4 select-none";
    labelR.innerHTML = `
        <div class="flex items-center gap-1">
            <span class="bg-slate-700 text-slate-200 rounded text-
[11px] px-1.5 py-0.5">RÂND ${r}</span>
        </div>
        <span class="text-[10px] font-semibold text-slate-500
italic">Multipli de ${r}</span>
    `;
    dRow.appendChild(labelR);

    const colsR = document.createElement('div');
    colsR.className = "flex items-center gap-0 relative w-full
h-full";

    // Add wire across the columns

```

```

const wireR = document.createElement('div');
wireR.className = "wire-line absolute left-0 right-0
pointer-events-none";
colsR.appendChild(wireR);

// Populate with divisor beads at aligned positions
for (let i = 1; i <= maxColumns; i++) {
  const colCell = document.createElement('div');
  colCell.className = "flex items-center justify-center
relative";
  colCell.style.width = `${COLUMN_WIDTH_PX}px`;

  if (i % r === 0) {
    // There is a multiple here! Label is k = i/r
    const k = i / r;
    const bead = document.createElement('div');
    bead.setAttribute('data-row', r);
    bead.setAttribute('data-col', i);
    bead.className = "abacus-bead w-9 h-9 rounded-full
bead-dark flex items-center justify-center font-bold text-sm text-
white/90 shadow-md z-10 select-none cursor-pointer";
    bead.textContent = k;

    // Clicking a sub-bead also triggers selection of the
main natural column
    bead.onclick = () => selectNumber(i);

    colCell.appendChild(bead);
  } else {
    // Empty space, only wire visible
  }
  colsR.appendChild(colCell);
}
dRow.appendChild(colsR);
abacusGrid.appendChild(dRow);
}

```

```

// Draw line overlay and triggers
setTimeout(() => {
    drawPrimeHelperLines();
    if (activeSelectedNumber) {
        selectNumber(activeSelectedNumber);
    }
}, 50);
}

// Update Inspector Output
function updateInspector(num, divisors) {
    inspectorEmpty.classList.add('hidden');
    inspectorContent.classList.remove('hidden');
    factorSummary.classList.remove('hidden');

    inspectNum.textContent = num;

    const isNPrime = isPrime(num);
    if (isNPrime) {
        inspectType.textContent = "NUMĂR PRIM";
        inspectType.className = "px-3 py-1 text-xs font-bold
rounded-full bg-red-600/20 text-red-400 border border-red-500/30";
    } else if (num === 1) {
        inspectType.textContent = "UNITATE";
        inspectType.className = "px-3 py-1 text-xs font-bold
rounded-full bg-slate-600/20 text-slate-400 border border-slate-500/30";
    } else {
        inspectType.textContent = "NUMĂR COMPUS";
        inspectType.className = "px-3 py-1 text-xs font-bold
rounded-full bg-amber-600/20 text-amber-400 border border-amber-
500/30";
    }
}

// List of divisors
inspectDivisorsList.innerHTML = ";

```

// Divisibility always includes 1 (conceptualized on Row 1 as the unit step)

```
const allDivisors = [1, ...divisors];

allDivisors.sort((a,b) => a-b).forEach(d => {
  const isDivisorPrime = isPrime(d);
  const badge = document.createElement('span');
  badge.className = `px-3 py-1 rounded-lg text-xs font-bold
shadow flex items-center gap-1 ${isDivisorPrime ? 'bg-red-950/40 text-
red-400 border border-red-800' : 'bg-slate-700 text-slate-200'}`;
  badge.innerHTML = `
    <span>${d}</span>
    <span class="text-[9px] opacity-65">${isDivisorPrime ?
'(prim)' : ""}</span>
  `;
  inspectDivisorsList.appendChild(badge);
});
```

// Write Equations showing step-by-step spatial construction  
inspectEquations.innerHTML = "";

```
// 1 x num = num
const eq1 = document.createElement('div');
eq1.innerHTML = `<span class="text-amber-400">Pas de
1:</span> 1 × ${num} = ${num} (Șirul de bază)`;
inspectEquations.appendChild(eq1);
```

```
divisors.forEach(d => {
  const k = num / d;
  const eq = document.createElement('div');
  eq.innerHTML = `<span class="text-amber-400">Pas de
${d}</span> ${d} × <span class="text-white bg-slate-800 px-1.5
rounded">${k}</span> = ${num}`;
  inspectEquations.appendChild(eq);
});
```

// Priming Factorization

```
    inspectFactors.innerHTML = getPrimeFactorization(num);
  }
</script>
</body>
</html>
```