

index.html

```
<!DOCTYPE html>
<html lang="th">
<head>
<meta charset="UTF-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0"/>
<title>COUNT QUEST – ກາຮົງຈິນັກນັບ</title>
<style>
body{
  font-family: system-ui, -apple-system, "Segoe UI", sans-serif;
  background: linear-gradient(135deg,#0f172a,#1e293b);
  color:#e2e8f0;
  margin:0;
  padding:0;
}
.wrap{
  max-width: 900px;
  margin: 0 auto;
  padding: 18px;
}
.card{
  background: rgba(255,255,255,0.06);
  border: 1px solid rgba(255,255,255,0.12);
  border-radius: 16px;
  padding: 16px;
  margin: 12px 0;
  box-shadow: 0 10px 30px rgba(0,0,0,0.25);
}
h1{
  margin: 8px 0 0;
  font-size: 26px;
}
.sub{
```

```
  opacity:0.9;  
  margin-top: 6px;  
  line-height: 1.4;  
}  
  
.row{  
  display:flex;  
  gap:12px;  
  flex-wrap:wrap;  
}  
  
.pill{  
  background: rgba(255,255,255,0.08);  
  border: 1px solid rgba(255,255,255,0.12);  
  border-radius: 999px;  
  padding: 8px 12px;  
  display:inline-flex;  
  gap:8px;  
  align-items:center;  
  font-size: 14px;  
}  
  
.btn{  
  border:none;  
  cursor:pointer;  
  padding: 10px 14px;  
  border-radius: 12px;  
  font-weight: 700;  
  background:#38bdf8;  
  color:#0b1220;  
  transition: transform .06s ease;  
}  
  
.btn:active{ transform: scale(0.98); }  
  
.btn.secondary{  
  background: rgba(255,255,255,0.12);  
  color:#e2e8f0;  
  border:1px solid rgba(255,255,255,0.18);  
}
```

```
}

.question{
  font-size: 18px;
  line-height: 1.5;
  margin: 10px 0 12px;
}

.choices{
  display:grid;
  grid-template-columns: 1fr;
  gap:10px;
}

@media(min-width:700px){
  .choices{ grid-template-columns: 1fr 1fr; }
}

.choice{
  text-align:left;
  background: rgba(255,255,255,0.08);
  border:1px solid rgba(255,255,255,0.14);
  border-radius: 14px;
  padding: 12px;
  cursor:pointer;
  font-weight: 650;
  color:#e2e8f0;
}

.choice:hover{
  border-color: rgba(56,189,248,0.8);
  box-shadow: 0 0 0 2px rgba(56,189,248,0.15);
}

.choice.correct{
  border-color: rgba(34,197,94,0.9);
  box-shadow: 0 0 0 2px rgba(34,197,94,0.15);
}

.choice.wrong{
  border-color: rgba(239,68,68,0.9);
```

```
box-shadow: 0 0 0 2px rgba(239,68,68,0.15);  
}  
.log{  
font-size: 14px;  
opacity:0.95;  
line-height:1.45;  
}  
.small{  
font-size: 13px;  
opacity:0.85;  
}  
input{  
width:100%;  
padding: 10px 12px;  
border-radius: 12px;  
border: 1px solid rgba(255,255,255,0.18);  
background: rgba(15,23,42,0.55);  
color:#e2e8f0;  
outline:none;  
}  
.badge{  
padding: 2px 8px;  
border-radius: 999px;  
font-size: 12px;  
background: rgba(56,189,248,0.18);  
border:1px solid rgba(56,189,248,0.25);  
}  
</style>  
</head>  
<body>  
<div class="wrap">  
<div class="card">  
<div class="row" style="justify-content:space-between; align-items:center;">  
<div>
```

COUNT QUEST – ภารกิจนักนับ

<div class="sub">มินิเกม "หลักการนับ" แบบสุ่มโจทย์ไม่ซ้ำ ✨ ตอบให้ทันเวลา เก็บคะแนน เอาชนะบอส!</div>

</div>

<div class="pill">Unique Mode  สุ่มโจทย์</div>

</div>

<div class="row" style="margin-top:12px;">

<div class="pill"> เวลา: <b id="time">60 วิ</div>

<div class="pill"> ชีวิต: <b id="life">3</div>

<div class="pill"> คะแนน: <b id="score">0</div>

<div class="pill"> เลvel: <b id="level">1</div>

</div>

<div style="margin-top:12px;" class="row">

<button class="btn" id="startBtn">เริ่มเกม</button>

<button class="btn secondary" id="resetBtn">รีเซ็ต</button>

</div>

</div>

<div class="card" id="gameCard" style="display:none;">

<div class="row" style="justify-content:space-between; align-items:center;">

<div class="pill"> โจทย์ที่ <b id="qNo">1 / <b id="qMax">10</div>

<div class="pill"> ประเภท: <b id="qType">-</div>

</div>

<div class="question" id="questionText">...</div>

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

<div style="margin-top:12px;" class="log" id="feedback"></div>

<div class="small" id="hint"></div>

</div>

<div class="card" id="endCard" style="display:none;">

<h2>สรุปผลภารกิจ 

```
<div class="sub" id="summary"></div>
<div style="margin-top:12px;" class="row">
  <button class="btn" id="playAgainBtn">เล่นอีกครั้ง</button>
  <button class="btn secondary" id="shareBtn">คัดลอกข้อความแชร์</button>
</div>
<div class="small" style="margin-top:10px;">
  *ถ้าคุณเอาเกมขึ้นเว็บแล้ว จะส่ง "ลิงก์เล่นเกม" ให้เพื่อนได้เลย
</div>
</div>
```

```
<div class="card">
  <h3>วิธีเล่น (สั้นๆ)</h3>
  <div class="sub">
    - ตอบโจทย์หลักการนับให้ถูกต้อง<br/>
    - ตอบเร็วๆได้โบนัสคะแนน<br/>
    - ตอบผิดเสียชีวิต<br/>
    - เล่นครบ 10 ข้อ หรือหมดเวลา เกมจบ
  </div>
</div>
</div>
```

```
<script>
/* =====
 COUNT QUEST – UNIQUE ENGINE
 โจทย์ถูกสร้างแบบ "Seeded Random"
 ทำให้เกมของคุณมีเอกลักษณ์ + ไม่ซ้ำ
===== */
```

```
const $ = (id)=>document.getElementById(id);
```

```
let state = {
  started:false,
  time:60,
  life:3,
```

```
score:0,  
level:1,  
qNo:1,  
qMax:10,  
usedSignatures:new Set(),  
timer:null,  
current:null,  
seed: Date.now() % 1000000,  
streak:0  
};
```

// Seeded RNG (ให้สุ่มแบบเป็นเอกลักษณ์ตามเครื่อง/เวลา)

```
function mulberry32(a){  
    return function(){  
        let t = a += 0x6D2B79F5;  
        t = Math.imul(t ^ t >>> 15, t | 1);  
        t ^= t + Math.imul(t ^ t >>> 7, t | 61);  
        return ((t ^ t >>> 14) >>> 0) / 4294967296;  
    }  
}  
let rand = mulberry32(state.seed);
```

```
function rInt(min,max){  
    return Math.floor(rand()*(max-min+1))+min;
```

```
}
```

```
function shuffle(arr){
```

```
    for(let i=arr.length-1;i>0;i--){
```

```
        const j=Math.floor(rand()*(i+1));
```

```
        [arr[i],arr[j]]=[arr[j],arr[i]];
```

```
}
```

```
    return arr;
```

```
}
```

```
function pick(arr){ return arr[Math.floor(rand()*arr.length)]; }
```

```
function factorial(n){  
    let x=1;  
    for(let i=2;i<=n;i++) x*=i;  
    return x;  
}  
function nPr(n,r){  
    if(r>n) return 0;  
    return factorial(n)/factorial(n-r);  
}  
function nCr(n,r){  
    if(r>n) return 0;  
    return factorial(n)/(factorial(r)*factorial(n-r));  
}  
  
function formatNum(x){  
    // กันเลขไทยด้วย  
    return x.toLocaleString("th-TH");  
}  
  
function updateHUD(){  
    $("time").textContent = state.time;  
    $("life").textContent = state.life;  
    $("score").textContent = state.score;  
    $("level").textContent = state.level;  
    $("qNo").textContent = state.qNo;  
    $("qMax").textContent = state.qMax;  
}  
  
function startTimer(){  
    if(state.timer) clearInterval(state.timer);  
    state.timer = setInterval(()=>{  
        if(!state.started) return;  
        state.time--;  
        $("time").textContent = state.time;  
    }, 1000);  
}
```

```
if(state.time<=0){  
    state.time=0;  
    endGame("หมดเวลา ");  
}  
},1000);  
}
```

```
function difficulty(){  
    // เพิ่มความยากตามเวลา  
    if(state.level<=1) return "easy";  
    if(state.level<=2) return "normal";  
    return "hard";  
}
```

```
function signature(obj){  
    // ลายเซ็นกันโจทย์ข้า  
    return JSON.stringify(obj);  
}
```

```
/* =====  
 Question Generators  
 1) หลักการคูณ (AND)  
 2) หลักการบวก (OR)  
 3) เรียงสับเปลี่ยน (Permutation)  
 4) จัดหมู่ (Combination)  
 5) เงื่อนไขห้ามติดกัน / ต้องติดกัน (Unique style)  
===== */
```

```
function genQuestion(){  
    let tries=0;  
    while(tries<60){  
        tries++;  
  
        const types = ["AND","OR","PERM","COMB","RULE"];
```

```

const type = pick(types);

const diff = difficulty();
let q = null;

if(type === "AND"){
    // เลือกชุดแต่งตัว/เมนู: a ทางเลือก x b ทางเลือก
    let a = rInt(3, diff === "easy" ? 6 : 8);
    let b = rInt(3, diff === "easy" ? 6 : 9);
    let theme = pick([
        "เลือกชุดไปโรงเรียน",
        "เลือกเครื่องดื่มในร้าน",
        "เลือกสีแพ็กเกจสินค้า",
        "เลือกสติ๊กเกอร์แต่งสมุด",
        "เลือกเมนูในคาเฟ่"
    ]);
    const ans = a * b;
    q = {
        type: "หลักการคูณ (AND)",
        text: `${theme}: มีตัวเลือกแบบที่ 1 จำนวน ${a} แบบ และแบบที่ 2 จำนวน ${b} แบบ ถ้าต้องเลือก "ทั้งสองอย่าง" จะมีวิธีเลือกได้กี่วิธี?`,
        answer: ans,
        explain: `ใช้หลักการคูณ: ${a} \times ${b} = ${ans}`
    };
}

if(type === "OR"){
    let a = rInt(4, diff === "easy" ? 8 : 12);
    let b = rInt(3, diff === "easy" ? 7 : 10);
    let overlap = rInt(0, Math.min(a, b, diff === "easy" ? 2 : 4));
    let theme = pick([
        "ชุมชนในโรงเรียน",
        "กิจกรรมวันกีฬา",
        "คอร์สสอนไลน์",
        "เมนูของหวาน"
    ]);
}

```

"เกมที่เลือกเล่น"

```
]);
const ans = a + b - overlap;
q = {
  type:"หลักการบวก (OR)",
  text:`${theme}: เลือกได้ "อย่างใดอย่างหนึ่ง" ระหว่างตัวเลือก A มี ${a} แบบ และตัวเลือก B มี ${b} แบบ แต่เมื่อตัวเลือกที่ซ้ำกัน ${overlap} แบบ รวมแล้วเลือกได้กี่แบบ?`,
  answer: ans,
  explain: `ใช้หลักการบวกแบบมีซ้ำ: ${a} + ${b} - ${overlap} = ${ans}`
};
```

}

```
if(type==="PERM"){
  let n = rInt(diff==="easy"?4:5, diff==="hard"?9:7);
```

```
  let r = rInt(2, Math.min(5,n));
  let theme = pick([
```

```
  "จัดคิวถ่ายรูป",
  "จัดอันดับผู้ชนะ",
  "จัดรหัสล็อกเกอร์",
  "เรียงตัวอักษร",
  "เรียงคนขึ้นเวที"
```

```
]);
```

```
  const ans = nPr(n,r);
```

```
  q = {
```

```
    type:"เรียงลับเปลี่ยน (Permutation)",
```

```
    text:`${theme}: มีของทั้งหมด ${n} ชิ้น ต้อง "เรียงลำดับ" เลือกมา ${r} ชิ้น จะทำได้กี่วิธี?`,
```

```
    answer: ans,
```

```
    explain: `Permutation: ${n}P${r} = ${n}! / (${n}-${r})! = ${ans}`
```

```
};
```

}

```
if(type==="COMB"){

  let n = rInt(diff==="easy"?5:6, diff==="hard"?11:9);
```

```
  let r = rInt(2, Math.min(5,n-1));
  let theme = pick([
```

```

    "เลือกทีมแข่งขัน",
    "เลือกเพื่อนทำโครงการ",
    "เลือกหนังสืออ่าน",
    "เลือกผลไม้ทำสลัด",
    "เลือกคนเป็นตัวแทนห้อง"
]);
const ans = nCr(n,r);
q = {
  type:"จัดหมู่ (Combination)",
  text: `${theme}: มีทั้งหมด ${n} ตัวเลือก ต้อง "เลือกโดยไม่สนใจลำดับ" จำนวน ${r} ตัวเลือก จะเลือกได้กี่วิธี?`,
  answer: ans,
  explain: `Combination: ${n}C${r} = ${n}! / (${r}!(${n}-${r})!) = ${ans}`
};
}

```

```

if(type==="RULE"){
  // เงื่อนไขพิเศษ: ห้ามติดกัน / ต้องติดกัน
  // ตัวเลขเล็กเพื่อไม่ให้หนักเกินไป แต่ดู "ไม่เหมือนใคร"
  let n = rInt(5, diff==="easy"?7:9);
  let theme = pick([
    "จัดที่นั่งในโรงหนัง",
    "เรียงคนถ่ายรูปหมู่",
    "เรียงหนังสือบนชั้น",
    "จัดแต่งเดินพาเหรด",
    "เรียงการ์ดในมือ"
  ]);
  let mode = pick(["no_adjacent","together"]);
}

```

```

if(mode==="together"){
  // ให้ 2 คนต้องติดกัน: n! / 2
  const ans = factorial(n) / 2;
  q = {
    type:"กฎพิเศษ (ต้องติดกัน)",
    text: `${theme}: มีคน ${n} คนเรียงเป็น列 โดยมี "คู่เพื่อนสนิท 2 คน" ต้องยืนติดกันเสมอ จะเรียงได้กี่วิธี?`,
    answer: ans,
    explain: `排列: ${n}P${r} = ${n}! / (${r}!(${n}-${r})!) = ${ans}`
  };
}

```

answer: ans,

explain: 'มองคู่เพื่อนเป็น 1 ก้อน \Rightarrow มี $n-1$ ก้อนเรียงได้ $(n-1)!$ และในก้อนลับได้ 2 วิธี $\Rightarrow 2 \times (n-1)! = \text{ans}$ '

};

} else {

// ห้ามติดกัน: ใช้วิธีแทรก (จ่ายๆ แบบ n ไม่ใหญ่)

// เลือก 2 คน A,B ห้ามติดกัน: ทั้งหมด $n!$ - ติดกัน

const total = factorial(n);

const adjacent = factorial(n-1) * 2;

const ans = total - adjacent;

q = {

type: "กฎพิเศษ (ห้ามติดกัน)",

text: `\${theme}: มีคน \${n} คนเรียงเป็นแทว โดยมี "คู่แข่ง 2 คน" ห้ามยืนติดกัน จะเรียงได้กี่วิธี?",

answer: ans,

explain: 'ทั้งหมด $n! = \text{total}$ วิธี\ทกรณีติดกัน: มองเป็น 1 ก้อน $\Rightarrow (n-1)!$ แล้วลับในก้อนได้ 2 $\Rightarrow 2 \times (n-1)! = \text{adjacent}$ \ทดังนั้นห้ามติดกัน = $\text{total} - \text{adjacent} = \text{ans}$ '

};

}

}

if(!q) continue;

// กันโจทย์ซ้ำ

const sig = signature({t:q.type, text:q.text, ans:q.answer});

if(state.usedSignatures.has(sig)) continue;

state.usedSignatures.add(sig);

// สร้างข้อยล์

const correct = q.answer;

const wrongs = new Set();

while(wrongs.size<3){

let w;

// สุ่มคำตอบหลอกให้ดูสมจริง

const bump = rInt(1,4);

const factor = pick([0.5, 0.75, 1.25, 1.5, 2]);

w = Math.round(correct*factor + (rand()<0.5? bump : -bump));

```
if(w<0) w = correct + rInt(2,9);
if(w==correct) continue;
wrongs.add(w);
}

const choices = shuffle([correct, ...Array.from(wrongs)]);

return {
...q,
choices
};

}
```

```
// fallback
return {
type:"หลักการคูณ (AND)",
text:"มีตัวเลือก 4 แบบ และอีกชุด 5 แบบ เลือกทั้งสองอย่าง ได้กี่วิธี?",
answer: 20,
explain:" $4 \times 5 = 20$ ",
choices: shuffle([20, 9, 18, 24])
};

}
```

```
function renderQuestion(){
state.current = genQuestion();
$("#qType").textContent = state.current.type;
$("#questionText").textContent = state.current.text;
$("#feedback").textContent = "";
$("#hint").textContent = "";
```

```
const box = $("#choices");
box.innerHTML = "";

state.current.choices.forEach((val, idx)=>{
const btn = document.createElement("button");

```

```
btn.className = "choice";
btn.textContent = `${String.fromCharCode(65+idx)}. ${formatNum(val)}`;
btn.onclick = ()=>checkAnswer(val, btn);
box.appendChild(btn);
});
```

```
updateHUD();
```

```
}
```

```
function checkAnswer(selected, btn){
  const correct = state.current.answer;
  const all = document.querySelectorAll(".choice");
  all.forEach(b=>b.disabled=true);
```

```
if(selected === correct){
  btn.classList.add("correct");
  state.streak++;
```

```
// คะแนน: ถูก + โบนัสตามเวลา + streak
```

```
const timeBonus = Math.max(0, Math.floor(state.time/10));
const streakBonus = Math.min(10, state.streak);
const gain = 10 + timeBonus + streakBonus;
```

```
state.score += gain;
```

```
$("#feedback").textContent = `✅ ถูกต้อง! +${gain} คะแนน`;
$("#hint").textContent = `เฉลย: ${state.current.explain}`;
```

```
// เพิ่มเลเวลทุก 3 ข้อที่ถูกติด
```

```
if(state.streak % 3 === 0) state.level++;
```

```
}else{
```

```
  btn.classList.add("wrong");
  state.life--;
```

```
state.streak = 0;
```

```
// แสดงปุ่มที่ถูก
```

```
all.forEach(b=>{
```

```
  const num = Number(b.textContent.replace(/\^d/g,""));
```

```
  // กันเลข format
```

```
  const raw = b.textContent.split(" ")[1].replace(/,/g,"");
```

```
  if(Number(raw) === correct) b.classList.add("correct");
```

```
});
```

```
  $("feedback").textContent = `✖ ยังไม่ถูก! คำตอบที่ถูกคือ ${formatNum(correct)} (เลี้ยง 1 ชีวิต)`;
```

```
  $("hint").textContent = `วิธีคิด: ${state.current.explain}`;
```

```
}
```

```
updateHUD();
```

```
setTimeout(()=>{
```

```
  if(state.life<=0){
```

```
    endGame("ชีวิตหมด ❤️");
```

```
    return;
```

```
}
```

```
  state.qNo++;
```

```
  if(state.qNo>state.qMax){
```

```
    endGame("ครบทุกข้อแล้ว 🎉");
```

```
    return;
```

```
}
```

```
  renderQuestion();
```

```
}, 1200);
```

```
}
```

```
function startGame(){
```

```
  state.started = true;
```

```
  state.time = 60;
```

```
  state.life = 3;
```

```
state.score = 0;  
state.level = 1;  
state.qNo = 1;  
state.qMax = 10;  
state.usedSignatures = new Set();  
state.streak = 0;  
  
$("gameCard").style.display = "block";  
$("endCard").style.display = "none";
```

```
updateHUD();  
startTimer();  
renderQuestion();  
}
```

```
function resetGame(){  
state.started = false;  
if(state.timer) clearInterval(state.timer);  
$("gameCard").style.display = "none";  
$("endCard").style.display = "none";  
state.time = 60;  
state.life = 3;  
state.score = 0;  
state.level = 1;  
state.qNo = 1;  
state.qMax = 10;  
updateHUD();  
}
```

```
function endGame(reason){  
state.started = false;  
if(state.timer) clearInterval(state.timer);  
  
$("gameCard").style.display = "none";
```

```

        $("endCard").style.display = "block";

        const rank = state.score >= 160 ? "👑 นักนับระดับตำนาน"
            : state.score >= 110 ? "🔥 นักนับขั้นโปร"
            : state.score >= 70 ? "✨ นักนับฝีกหัด"
            : "🌱 มือใหม่หัดนับ";

        $("summary").innerHTML = `
            เหตุผลจบเกม: <b>${reason}</b><br/>
            คะแนนรวม: <b>${state.score}</b><br/>
            เลเวลสูงสุด: <b>${state.level}</b><br/>
            ฉายา: <b>${rank}</b><br/><br/>
            <span class="small">Seed เกมของคุณ: <b>${state.seed}</b> (ทำให้เกมนี้ "ไม่เหมือนใคร")</span>
        `;
    }

    $("startBtn").onclick = startGame;
    $("resetBtn").onclick = resetGame;
    $("playAgainBtn").onclick = ()=>{
        // เปลี่ยน seed ให้รอบใหม่ไม่ซ้ำ
        state.seed = (Date.now() + Math.floor(Math.random()*99999)) % 1000000;
        rand = mulberry32(state.seed);
        startGame();
    };

    $("shareBtn").onclick = async ()=>{
        const msg = `มาเล่น COUNT QUEST – ภารกิจนักนับกัน! 🎯✨
        เกมคณิตศาสตร์หลักการนับ (สูมโจทย์+เก็บคะแนน)
        คะแนนที่ฉันทำได้: ${state.score}
        (ถ้าอัปขึ้นเว็บแล้ว ใส่ลิงก์ของคุณตรงนี้)`;

        try{
            await navigator.clipboard.writeText(msg);
            alert("คัดลอกข้อความเช็คแล้ว ✅");
        }
    }
}

```

```
 }catch(e){  
     alert("คัดลอกไม่สำเร็จ (อาจต้องใช้ https) ลองคัดลอกเองจากหน้าจอแทน");  
 }  
};  
  
updateHUD();  
</script>  
</body>  
</html>
```