

Flash Encode

Skill: Checkerboard Lookup -- plaintext letter to codegroup and back

OVERVIEW

Flash Encode drills the straddling checkerboard -- the variable-length substitution table at the heart of the Soviet-era hand cipher. Some letters produce a single digit; others require two (an escape digit followed by a row digit). Fluency here is the prerequisite for every other OTP skill.

PLAYERS & MATERIALS

- 2 players
- 1 shared straddling checkerboard (printed or drawn on paper)
- Optional: stopwatch

STANDARD STRADDLING CHECKERBOARD

A common training layout (adapt to your operational checkerboard):

	0	1	2	3	4	5	6	7	8	9
	E	T	A	O	N		I	S		R
6	B	C	D	F	G	H	J	K	L	M
8	P	Q	U	V	W	X	Y	Z	0	/

The blank row positions (0, 5, 8 -- adjust to your board) are single-digit. Letters in rows 6 and 8 require two digits (6x or 8x).

HOW TO PLAY

- Player A calls a letter at random. Player B states the codegroup without pausing.
- Alternate: Player A calls a 1- or 2-digit codegroup. Player B names the letter (or NULL if the digit is an escape marker mid-stream).
- Score one point per correct answer. First to 20 points wins the round.
- Timed variant: set a 60-second timer. Count correct answers. Track personal bests.

RULES

- No looking at the checkerboard after the first 5 minutes of a session.
- A two-digit codegroup must be stated as a unit -- no pausing between digits.
- Calling an escape digit alone (6 or 8 without its follow-on) is an error.
- Spaces and numerals must also be handled correctly (numerals use a figure shift).

EXAMPLE ROUND

Caller says	Correct response	Notes
H	67	Row 6, column 7
A	2	Direct single-digit
83	Y	Row 8, column 3 = Y
R	9	Direct single-digit
61	B	Row 6, column 1 = B

Expected outcome: Players can encode 10+ letters per minute without the reference card after 3 to 4 practice sessions. Decode (digits to letters) typically takes 2x as long to reach the same fluency.

Tens Complement Snap

Skill: Mod-10 Subtraction via additive complement -- the foundation of fast decryption

OVERVIEW

Decryption requires subtracting the key digit from the cipher digit, mod 10. Under operational stress, subtraction is error-prone. The correct technique is to add the tens complement: instead of $C - K$, compute $C + (10 - K) \bmod 10$. This game makes complement lookup reflexive.

PLAYERS & MATERIALS

- 2 players (or solo with a timer)
- 1 standard d10 (or a deck of cards, Ace=1 through 10=0)
- Optional: stopwatch

TENS COMPLEMENT TABLE

Digit K	0	1	2	3	4	5	6	7	8	9
Complement $(10-K) \bmod 10$	0	9	8	7	6	5	4	3	2	1

HOW TO PLAY -- SNAP MODE

- Player A calls a digit 0-9 (or rolls d10). Player B must immediately state the tens complement.
- Any hesitation over 2 seconds is scored as an error.
- Play to 30 consecutive correct answers without error. Any error resets the streak.
- Once comfortable, Player A increases pace: roughly one call per second.

HOW TO PLAY -- SUBTRACT WAR

- Both players roll a d10 simultaneously. Cipher = Player A's roll. Key = Player B's roll.
- Both players independently compute $C - K \bmod 10$ (using complement addition).
- First to call the correct result wins the trick. Most tricks after 20 rounds wins.
- Verify by checking: $\text{result} + K \bmod 10$ must equal C.

RULES

- Complement of 0 is 0, not 10. This trips most beginners.
- You may not use paper. Mental arithmetic only.
- In Subtract War, calling an answer before computing it is a forfeit of that trick.

EXAMPLE EXCHANGES

K called	Correct complement	Verify: $K + \text{comp} \bmod 10$
0	0	$0 + 0 = 0 \bmod 10 = 0$

3	7	$3 + 7 = 10 \text{ mod } 10 = 0$
7	3	$7 + 3 = 10 \text{ mod } 10 = 0$
9	1	$9 + 1 = 10 \text{ mod } 10 = 0$
5	5	$5 + 5 = 10 \text{ mod } 10 = 0$

SUBTRACT WAR EXAMPLE

A rolls (C)	B rolls (K)	Complement of K	C + comp mod 10	Result
8	3	7	$8 + 7 = 15$	5
2	9	1	$2 + 1 = 3$	3
4	4	6	$4 + 6 = 10$	0

Expected outcome: 30 consecutive complements without error in under 45 seconds. At this level decryption column work no longer requires the complement table as a crutch.

Column Race

Skill: Mod-10 Addition -- encryption arithmetic across a full message block

OVERVIEW

Column Race simulates the core encryption step: adding plaintext codegroups to key digits, mod 10, to produce ciphertext. Each player generates their own key stream, encodes the same plaintext, then decrypts the other's work. This drills both directions of the arithmetic and surfaces errors at the column level.

PLAYERS & MATERIALS

- 2 players
- 2 d10 dice (one per player), or a shared source rolled alternately
- Paper and pencil for each player
- 1 agreed plaintext message (10-15 characters, pre-written)
- 1 shared straddling checkerboard

HOW TO PLAY

- Both players independently roll and record a 15-digit key stream.
- Each player encodes the agreed plaintext using their checkerboard, producing a plaintext digit stream.
- Each player adds their key stream to their plaintext stream, mod 10, digit by digit, producing their ciphertext.
- Players exchange ciphertext strips and key streams.
- Each player decrypts the other's ciphertext using the provided key, then reverse-lookups through the checkerboard to recover plaintext.
- First player to correctly recover the original message and call it aloud wins.

RULES

- No carrying. $7 + 6 = 3$, not 13. This must become instinctive.
- No borrowing on decrypt. Use complement addition, never subtraction.
- Key streams must be written before encoding begins -- no generating key on the fly.
- Grouping: write digits in blocks of 5 to match standard 5-group format.

EXAMPLE -- ENCODING 'HELLO'

Step	H	E	L	L	O
Plaintext letter	H	E	L	L	O
Checkerboard out	67	1	62	62	3

Key stream	4	8	3	5	7
$P + K \text{ mod } 10$	01	9	95	17	0

Note: each checkerboard digit is added to its own key digit independently. A two-digit codegroup (e.g. 67 for H) consumes two key digits (e.g. 4 and 8), producing two cipher digits (0 and 5). The key stream must be long enough to cover the full plaintext digit stream, not just the character count.

WORKED COLUMN ADDITION

Position	1	2	3	4	5	6	7	8	9	10
Plaintext digits	6	7	1	6	2	6	2	3		
Key digits	4	8	3	5	0	9	1	7		
Cipher (mod 10)	0	5	4	1	2	5	3	0		

Expected outcome: Both players recover the correct plaintext within 3 minutes. A clean run with zero arithmetic errors across a 15-digit block is the passing standard.

Running Sum

Skill: Sustained digit-stream attention -- maintaining position across a long key block

OVERVIEW

Sustained attention across a long column of digits is where most beginners lose position. A single transposed digit or skipped row corrupts every subsequent group. Running Sum isolates this skill: no encoding, no lookup -- just pure mod-10 accumulation under time pressure with random challenges.

PLAYERS & MATERIALS

- 2 players (Caller and Tracker)
- 1 d10 or a prepared strip of 40+ random digits
- Paper (Caller only -- Tracker works mentally)

HOW TO PLAY

- Caller reads digits aloud at a steady pace -- approximately one per second.
- Tracker maintains a running mod-10 sum mentally: start at 0, add each digit mod 10.
- At any point Caller may say 'Sum?' -- Tracker must immediately state current sum.
- An incorrect answer or hesitation over 3 seconds is an error. Record errors.
- After 30 digits, Caller reveals the correct running sum at each challenge point.
- Swap roles. Lowest error count across a full session wins.

RULES

- Tracker may not use paper, fingers, or any physical counting aid.
- Caller sets a consistent tempo -- no deliberately irregular pauses to trick Tracker.
- A 'Sum?' challenge within the first 3 digits of a new run is not allowed.
- If Tracker loses position entirely, they may say 'Reset' -- scored as 2 errors.

EXAMPLE -- FIRST 10 DIGITS

Position	1	2	3	4	5	6	7	8	9	10
Digit called	3	7	1	5	9	2	6	4	8	0
Running sum (mod 10)	3	0	1	6	5	7	3	7	5	5

Verification: $3+7=10 \text{ mod } 10=0$, $+1=1$, $+5=6$, $+9=15 \text{ mod } 10=5$, $+2=7$, $+6=13 \text{ mod } 10=3$, $+4=7$, $+8=15 \text{ mod } 10=5$, $+0=5$. Final sum: 5. Challenges at positions 4 and 9 would expect answers 6 and 5 respectively.

PROGRESSIVE DIFFICULTY

- Level 1: Pace of 1 digit per 2 seconds. Challenges every 5 digits.

- Level 2: Pace of 1 digit per second. Challenges every 3-4 digits.
- Level 3: Pace of 2 digits per second. Random challenges. 40-digit runs.
- Operational standard: 30 correct digits at Level 2 pace with no errors.

SOLO VARIANT

Prepare a printed strip of 40 digits. Set a metronome or ticking clock. Work through the strip at pace, recording your running sum every 5 digits. Check against the answer key you prepare before starting. Track your first-error position across sessions -- improvement is visible quickly.

Expected outcome: Tracker maintains correct running sum through a 30-digit sequence with no more than one error. At operational readiness, zero errors across a full 50-digit run.

Dead Drop

Skill: Full OTP cycle -- key generation, encode, transmit, decode, verify

OVERVIEW

Dead Drop is the capstone game. It simulates a complete one-time pad operation from key generation through plaintext recovery, using all prior skills in sequence. Errors are isolated by phase, allowing players to identify exactly where their process breaks down. Rotate roles each round.

PLAYERS & MATERIALS

- 2 players (Sender and Receiver)
- 2 d10 dice per player
- Paper and pencil for each player
- 1 shared straddling checkerboard
- 1 pre-written plaintext message (10-15 characters) -- agreed before play
- A privacy screen (book or folder) to prevent peeking during key generation

PHASE 1 -- KEY GENERATION

- Both players independently roll 20 d10 dice and record the results as a key strip in 5-group format (e.g. 73819 42056 11734 92801).
- Players exchange key strips and verify they match exactly. Any discrepancy must be resolved before proceeding -- in a real OTP this would be a catastrophic failure.
- Record the matched key. This is the shared one-time pad.

PHASE 2 -- ENCODE

- Sender encodes the plaintext through the checkerboard, producing a digit stream.
- Sender adds the key stream digit-by-digit, mod 10, to produce ciphertext.
- Sender records: plaintext, plaintext digit stream, key stream, ciphertext.
- Sender passes only the ciphertext to the Receiver.

PHASE 3 -- DECODE

- Receiver subtracts the key stream from the ciphertext, mod 10 (using complement addition), producing a plaintext digit stream.
- Receiver reverse-lookups through the checkerboard to recover the message.
- Receiver announces the recovered plaintext aloud.

PHASE 4 -- VERIFY AND SCORE

Phase	Error type	Penalty
Key generation	Mismatch between strips	3 points

Encode -- checkerboard	Wrong codegroup for a letter	1 point each
Encode -- arithmetic	Incorrect mod-10 addition	1 point each
Decode -- arithmetic	Incorrect mod-10 subtraction	1 point each
Decode -- checkerboard	Wrong letter for codegroup	1 point each
Full clean round	Zero errors, under 5 minutes	-2 points (bonus)

Lowest score after 5 rounds wins. Penalty points accumulate -- a clean round bonus can offset prior errors.

RULES

- Key strips must be written and verified before any encoding begins.
- The Sender's plaintext record is the authoritative source for verification.
- Receiver must decode without being shown the plaintext digit stream.
- After verification, both players review errors by phase and discuss root cause.

Expected outcome: A clean round (zero errors, sub-5 minutes) after 6 to 8 sessions. Early sessions will reveal whether errors cluster in arithmetic, checkerboard lookup, or key discipline -- the phase breakdown guides further practice with earlier games.

Farkle Mod-10

Skill: Mod-10 arithmetic under distraction -- embedded in a real dice game

OVERVIEW

Farkle Mod-10 embeds cipher arithmetic inside a familiar press-your-luck dice game. Because Farkle demands strategic decisions, the mod-10 tracking runs in the background -- simulating the divided attention of real operational conditions where the cryptographer may be interrupted, tired, or managing multiple tasks.

PLAYERS & MATERIALS

- 2 to 4 players
- 6 standard d6 dice
- Paper and pencil for scoring
- Optional: 1 d10 for tiebreakers

STANDARD FARKLE RULES (SUMMARY)

- On your turn, roll all 6 dice. You must set aside at least one scoring die.
- You may re-roll remaining dice or bank your score and end your turn.
- If all dice score, you may pick up all 6 and continue rolling (hot dice).
- If no dice score on a roll, you Farkle -- lose all points accumulated that turn.
- Standard scoring: single 1 = 100pts, single 5 = 50pts, three of a kind = face x 100 (three 1s = 1000), four or more of a kind doubles with each additional die.

MOD-10 SCORING MODIFICATION

- All scores are tracked as mod-10 running sums, not standard totals.
- A score of 100 points adds 1 to your running sum (mod 10).
- A score of 50 points adds 5 to your running sum (mod 10).
- A score of 200 points adds 2. A score of 300 adds 3. And so on.
- You BUST (lose your turn's accumulation) if your running sum reaches exactly 0 mid-turn (not at the start of a turn).
- First player to reach a running sum of 7 (and hold it at end of turn) wins. Adjust target to 5 for shorter games.

MOD-10 SCORE REFERENCE

Dice result	Standard pts	Mod-10 add	Example: sum was 6, now...
Single 1	100	+1	$6 + 1 = 7$
Single 5	50	+5	$6 + 5 = 11 \text{ mod } 10 = 1$

Three 1s	1000	+1	$6 + 1 = 7$ (1000 \rightarrow +1 same as single 1 rule)
Three 2s	200	+2	$6 + 2 = 8$
Three 5s	500	+5	$6 + 5 = 1$
Three 6s	600	+6	$6 + 6 = 2$
Straight 1-6	1500	+5	$6 + 5 = 1$

EXAMPLE TURN

Player's running sum starts at 4. They roll and score: three 3s (300 pts, +3). Sum moves to 7. They decide to keep rolling (greedy). Next roll scores a single 5 (+5). Sum: $7 + 5 = 12 \bmod 10 = 2$. They are back down. They roll again and Farkle -- lose the turn. Sum reverts to 4 (the value at start of turn).

RULES

- Running sum reverts to its value at the start of a turn if you Farkle.
- You may not bank a sum of 0 -- you must keep rolling or you are scored as Farkled.
- The bust rule (sum hits 0 mid-turn) applies only after the first scoring die is banked.
- In case of dispute, write down each partial sum on paper and reconstruct.
- For a cooperative variant, all players contribute to a shared running sum and attempt to reach the target without any player Farkling.

WHY THIS WORKS

The strategic pressure of Farkle (bank vs. roll decisions, counting remaining dice, watching opponents) simulates the real cognitive load of cipher work in non-ideal conditions. If a player can maintain accurate mod-10 tracking through a full Farkle game, they are ready for sustained column work.

Expected outcome: Players track mod-10 sums accurately through a full game (typically 20 to 40 turns) with no external verification needed. Disagreements should be rare and resolvable by reconstruction from written partial sums.