

# Trik sa kartama

Dva igrača će pokazati trik sa kartama sa standardnim špilom od 52 karte. Iz praktičnih razloga vrijednosti karti će biti različiti cijeli brojevi od 0 do 51.

Karte će se na početku postavljati na stol u jednom redu licem prema gore (sa vidljivim vrijednostima) u nekom redoslijedu nepoznatom igračima.

Prvi igrač odlazi do stola, gleda karte i vrši izmjene, ukupno najviše  $S$  puta. Svaka izmjena vrši se izborom dvije karte na pozicima  $i$  i  $j$  ( $i$  i  $j$  mogu biti jednake) i pomjeranjem karata s položaja  $i$  na položaj  $j$  i obratno.

Nakon toga prvi igrač odlazi bez da komunicira sa drugim igračem i sve karte se onda okreću (njihove vrijednosti više nisu vidljive) bez promjene njihovog redoslijeda. Drugi igrač je pozvan za stol i zamoljen je da pogodi gdje je karta s ciljanom vrijednošću (**target**) i dozvoljeno mu je da okrene najviše  $T$  karti jednu po jednu. Ako je neka od otkrivenih karata ciljna, **target**, tada igrači pobjeđuju. Ako ostanu bez pretpostavki (guesses), gube.

Vaš cilj je napisati dva programa koji će simulirati postupke igrača i pobijediti u igri.

## Detalji implementacije

Dobit ćete dva programa - FirstPlayer i SecondPlayer zajedno sa sample grejderom.

U FirstPlayer morate implementirati sljedeću funkciju:

```
void swapCards(int cards[], int S, int T)
```

- Ovu funkciju grejder poziva tačno jednom
- cards: niz koji sadrži početne vrijednosti karata s lijevo na desno, s tačno 52 elementa indeksiranih od 0 do 51
- S: broj dozvoljenih zamjena  
T: broj dozvoljenih pretpostavki (guesses)

**swapCards** može pozvati sljedeću funkciju:

```
void doSwap(int i, int j)
```

- $i$ : indeks prve zamijenjene karte,  $0 \leq i < 52$
- $j$ : indeks druge zamijenjene karte,  $0 \leq j < 52$
- **doSwap** se može pozvati najviše  $S$  puta

U SecondPlayer morate implementirati sljedeću funkciju:

```
void guessCard(int S, int T, int target)
```

- S: broj dozvoljenih zamjena
- T: broj dozvoljenih pretpostavki, guesses
- target: vrijednost karte koju treba otkriti

**guessCard** može pozvati sljedeću funkciju:

```
int guess(int idx)
```

- idx: pretpostavljeni (guessed) indeks,  $0 \leq idx < 52$
- Vraća vrijednost idx-te karte
- **guess** se može pozvati najviše T puta.
- Kada se dogodi tačna pretpostavka (guess) evaluacija se uspješno završi

## Primjer interakcije

Ispod je primjer unosa za priloženi grejder.

Prvi red bi trebao sadržavati dva cijala broja: S i T.

Drugi red bi trebao sadržavati 52 broja. i-ti koji označava vrijednost i-te karte.

Treći red bi trebao sadržavati cjelobrojni **cilj (target)**.

Primjer inputa u graderu	Primjer poziva		
	Pozivi	Potpozivi	Povratne informacije
1 51	swapCards([0,1,...], 1, 51)		
0 1 2 3 4 5 6 7 8		doSwap(0, 1)	
9 10 11 12 13			Mijenja karte sa indeksima 0 i 1
14 15 16 17 18	swapCards finishes		.
19 20 21 22 23	guessCard(1, 51, 1)		
24 25 26 27 28		guess(5)	
29 30 31 32 33			guess vraća 5
34 35 36 37 38		guess(1)	
39 40 41 42 43			guess vraća 0
44 45 46 47 48		guess(0)	
49 50 51			Tačno!
1			

## Ograničenja

- $1 \leq S \leq 52$

- $1 \leq T \leq 51$
- $0 \leq \text{target} < 52$

## Podzadaci

1. (16 bodova):  $S=52, T=1$
2. (20 bodova):  $S+T=52$
3. (22 boda):  $S=13, T=27$
4. (18 bodova):  $S=1, T=26$
5. (24 boda): Strategija dobitka postoji za date  $S$  i  $T$