

# Kart oyunu

İki oyuncu standart 52 kartlık bir iskambil destesi ile bir kart gösterisi yapacaklardır. Size kolaylık olması için kartlar 0'dan 51'e birbirlerinden farklı tamsayılar ile belirtilmiştir. Kartlar ilk olarak bir masa üzerine tek bir sırada yüzleri görülecek şekilde (kart değerleri görülebilecek şekilde) sıralanırlar. Kartların sıralaması oyuncular tarafından bilinmemektedir.

İlk oyuncu masaya gider ve kartlara bakar ve sonra kartları ikili ikili yer değiştirir (swap). En fazla  $S$  swap yapar. Her bir swap için  $i$  ve  $j$  pozisyonlarında iki kart seçilir ( $i$  ve  $j$  eşit olabilir) ve  $i$ . pozisyonundaki kart  $j$ . pozisyona ve  $j$ . pozisyonundaki kart  $i$ . pozisyona getirilir. İlk oyuncu ikinci oyuncu ile hiç konuşmadan oyun alanında ayrılır ve bütün kartlar sıraları bozulmadan ters çevirilir (kart değerleri artık görünmez). İkinci oyuncu masaya davet edilir ve belirli bir **hedef** değerindeki kartın nerede olduğunu tahmin etmesi istenir ve en fazla  $T$  tane kartı tek tek sırayla ters çevirebilmesine izin verilir. Açılan kartlardan herhangi birisi **hedef** kart ise o zaman oyuncular oyunu kazanır.  $T$  tahminde hedef kart bulunamazsa oyuncular oyunu kaybederler.

Sizin hedefiniz oyuncuların hamlelerini simüle eden iki program yazarak oyunu kazanmaktır.

## Implementasyon Detayları

Size iki program FirstPlayer ve SecondPlayer ve örnek grader verilecektir.

FirstPlayer içinde aşağıdaki fonksiyonu implement etmelisiniz:

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

- Bu fonksiyon grader tarafından bir kez çağırılır
- cards: soldan sağa kartların ilk değerlerini içeren dizi, 0'dan 51'e indekslenmiş tam olarak 52 değer içerir
- S: izin verilen swap sayısı
- T: izin verilen tahmin sayısı

**swapCards** aşağıdaki fonksiyonu bir ya da daha fazla çağırabilir:

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

- $i$ : swap edilecek ilk kartın indeksi,  $0 \leq i < 52$
- $j$ : swap edilecek ikinci kartın indeksi,  $0 \leq j < 52$
- **doSwap** fonksiyonu en fazla  $S$  kere çağırılabilir

SecondPlayer programında aşağıdaki fonksiyonu implement etmelisiniz:

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

- S: izin verilen swap sayısı
- T: izin verilen tahmin sayısı
- target: yeri bulunması gereken hedef kartın değeri

**guessCard** aşağıdaki fonksiyonu bir ya da daha fazla çağırabilir:

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

- idx: tahmin yapılan indeks,  $0 \leq idx < 52$
- idx indeksindeki kartın değerini döner
- **guess** en fazla T kere çağırılabilir.
- Doğru tahminde bulunulduğunda değerlendirme başarılı bir şekilde sonlanır

## Örnek etkileşim

Verilen grader için aşağıda örnek bir girdi bulunmaktadır.

İlk satır iki tamsayı içermelidir: S ve T.

İkinci satır 52 sayı içermelidir. i. sayı i. indeksteki kartın değerini gösterir.

Üçüncü satır **hedef** değerini göstermektedir.

Grader'a örnek girdi	Örnek Çağrılar		
	Çağrılar	alt-çağrılar	Dönen değerler
1 51	swapCards([0,1,...], 1, 51)		
0 1 2 3 4 5 6 7 8		doSwap(0, 1)	
9 10 11 12 13			0 ve 1 indeksindeki kartların yeri değişir
14 15 16 17 18	swapCards biter		.
19 20 21 22 23	guessCard(1, 51, 1)		
24 25 26 27 28		guess(5)	
29 30 31 32 33			guess 5 döner
34 35 36 37 38		guess(1)	
39 40 41 42 43			guess 0 döner
44 45 46 47 48		guess(0)	
49 50 51			Doğru!
1			

## Kısıtlar

- $1 \leq S \leq 52$
- $1 \leq T \leq 51$
- $0 \leq target < 52$

## Altgörevler

1. (16 puan):  $S = 52, T = 1$
2. (20 puan):  $S + T = 52$
3. (22 puan):  $S = 13, T = 27$
4. (18 puan):  $S = 1, T = 26$
5. (24 puan): Verilen  $S$  ve  $T$  değerleri için bir kazanma stratejisi mevcuttur