

Kártyatrükk

Két játékosnak egy kártyatrükköt kell bemutatnia egy 52 kártyából álló paklival. A kártyák értékei különböző egész számok, 0-tól 51-ig.

A kártyák kezdetben az asztalon vannak, egy sorban felfelé fordítva (az értékeik látszódnak), a játékosok számára előzetesen ismeretlen sorrendben.

Az első játékos odamegy az asztalhoz, ránéz a kártyákra, és csinálhat néhány cserét (akár egyet sem), összesen legfeljebb **S**-et. Minden cserét úgy hajt végre, hogy választ két kártyát, az *i*-edik és *j*-edik pozícióban, majd az *i*-edik pozícióból az *j*-edik pozícióba teszi a kártyát, és fordítva.

Ezután az első játékos elhagyja az asztalt anélkül, hogy bármilyen kommunikációt folytatna a második játékosal. Ezután az összes kártyát lefordítjuk (az értékeik nem látszódnak) a sorrend megváltoztatása nélkül. A második játékos odamegy az asztalhoz, és ki kell találnia, hogy egy, ekkor közölt **target** értékű kártya hol van. Ennek érdekében egyesével felfordíthat legfeljebb **T** darab kártyát. Ha bármelyik felfordított kártya értéke **target**, akkor a játékosok nyernek. Ha elfogynak a próbálkozások a kívánt kártya felfordítása nélkül, akkor veszítenek.

A te feladatod, hogy írsz két programot, amelyek a játékosok szerepében szimulálják a kártyatrükköt, és nyernek.

Megvalósítás részletei

Biztosítunk neked két programot – **FirstPlayer** és **SecondPlayer** – és egy minta értékelőprogramot, ezekben kell dolgoznod.

A **FirstPlayer** fájlban az alábbi függvényt kell kódolnod:

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

- Ezt a függvényt pontosan egyszer hívja az értékelőprogram.
- **cards**: a kártyák kezdeti sorrend szerinti értékeit tartalmazó, pontosan 52 elemű tömb, 0-tól 51-ig van indexelve.
- **S**: a megengedett cserék száma.
- **T**: a második játékosnak megengedett próbálkozások száma.

A **swapCards** függvényen belül a következő függvényt többször is hívhatod:

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

- **i**: a felcserélendő első kártya sorszáma, $0 \leq i < 52$
- **j**: a felcserélendő második kártya sorszáma, $0 \leq j < 52$
- **doSwap** függvényt legfeljebb **S**-szer lehet hívni.

A **SecondPlayer** fájlban a következő függvényt kell kódolnod:

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

- **S**: az első játékosnak megengedett cserék száma
- **T**: a megengedett próbálkozások száma
- **target**: a megtalálendő kártya értéke.

A **guessCard** függvényen belül a következő függvényt többször is hívhatod:

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

- **idx**: a felfordítandó kártya sorszáma, $0 \leq idx < 52$
- A függvény visszaadja az **idx** sorszámú kártya értékét
- **guess** függvényt legfeljebb **T**-szer lehet hívni.
- Amikor a helyes (**target** értékű) kártyára hívod meg ezt a függvényt, akkor az értékelőprogram leállítja a futtatást.

Minta interakció

Alább látható egy példabemenet, amit a csatolt értékelőprogramnak adhatsz.

Az első sor két számot tartalmaz: **S** és **T**.

A második sor 52 számot tartalmaz, az *i*-edik szám adja meg a kezdeti sorrendben az *i*-edik kártya értékét.

A harmadik sor egyetlen számot tartalmaz, a **target** értékét.

Az értékelő mintabemenete	Mintahívások		
	Hívások	Belső hívások	Visszatérési értékek
1 51	swapCards([0,1,...], 1, 51)		
0 1 2 3 4 5 6 7 8		doSwap(0, 1)	
9 10 11 12 13 14			Felcseréli 0 és 1 sorszámú kártyákat
15 16 17 18 19			
20 21 22 23 24	swapCards befejeződik		
25 26 27 28 29	guessCard(1, 51, 1)		
30 31 32 33 34		guess(5)	
35 36 37 38 39			
40 41 42 43 44			guess 5-öt ad vissza
45 46 47 48 49			
50 51		guess(1)	
1			guess 0-át ad vissza
		guess(0)	
			Helyes!

Korlátok

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

Részfeladatok

1. (16 pont): $S = 52, T = 1$
2. (20 pont): $S + T = 52$
3. (22 pont): $S = 13, T = 27$
4. (18 pont): $S = 1, T = 26$
5. (24 pont): A megadott S és T számokra létezik nyerő stratégia.