

Card Trick

Երկու խաղացողներ պատրաստվում են ցույց տալ խաղաքարտային հնարք դասական 52 խաղաքարտ պարունակող կապոցով: Հարմարավետության համար խաղաքարտերի արժեքները կլինեն տարբեր ամբողջ թվեր՝ 0-ից 51:

Սկզբում խաղաքարտերը տեղադրված են սեղանի վրա մեկ շարքով երեսը դեպի վեր (տեսանելի արժեքներով) խաղացողների համար որևէ անհայտ հերթականությամբ: Առաջին խաղացողը գնում է սեղանի մոտ, նայում է խաղաքարտերին և կատարում է որոշ տեղափոխություններ, ընդամենը առավելագույնը S անգամ: Յուրաքանչյուր տեղափոխություն կատարվում է ընտրելով երկու խաղաքարտ տեղակայված i և j դիրքերում (i -ն և j -ն կարող են լինել հավասար) և շարժելով i -րդ դիրքի խաղաքարտը դեպի j -րդ դիրք և հակառակը:

Այդ ամենից հետո առաջին խաղացողը հեռանում է սեղանի մոտից առանց երկրորդ խաղացողի հետ շփվելու և բոլոր քարտերը պտտվում են երեսը դեպի ներքև (նրանց արժեքները այլևս տեսանելի չեն) առանց հերթականությունը խախտելու: Երկրորդ խաղացողը հրավիրվում է սեղանի մոտ և պահանջվում է գուշակել, թե որտեղ է **target** արժեքով խաղաքարտը տեղակայված և թույլատրվում է շրջել մաքսիմում T խաղաքարտեր մեկը մյուսի հետևից: Եթե որևէ երևացող խաղաքարտերից մեկը **target**-ն է, ապա խաղացողները հաղթում են: Եթե նրանց գուշակությունները ավարտվում են, նրանք պարտվում են:

Ձեր խնդիրն է գրել երկու ծրագրեր, որոնք կմոդելավորեն խաղացողների քայլերը և կհաղթեն խաղը:

Իրականացման մանրամասներ

Ձեզ կտրամադրվեն երկու ծրագրեր – FirstPlayer-ը և SecondPlayer-ը գնահատող ծրագրի հետ միասին:

FirstPlayer-ում դուք պետք է իրականացնեք հետևյալ ֆունկցիան.

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

- Այս ֆունկցիան կանչվելու է գնահատող ծրագրի կողմից ճիշտ մեկ անգամ
- Cards՝ զանգված պարունակող սկզբնական խաղաքարտերի արժեքները ձախից աջ, ճիշտ 52 տարրերով ինդեքսավորված 0-ից մինչև 51
- S՝ տեղափոխությունների քանակը
- T՝ թույլատրվող գուշակությունների քանակը

swapCards կարող է կատարել կանչեր հետևյալ ֆունկցիային.

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

- i ՝ տեղափոխության զույգի առաջին տարրի ինդեքսը, $0 \leq i < 52$
- j ՝ տեղափոխության զույգի առաջին տարրի ինդեքսը, $0 \leq j < 52$
- **doSwap** կարող է կանչվել առավելագույնը S անգամ

SecondPlayer-ում դուք պետք է իրականացնեք հետևյալ ֆունկցիան.

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

- S ՝ տեղափոխությունների քանակը
- T ՝ թույլատրվող գույնափոխությունների քանակը
- **target**՝ խաղաքարտի արժեքը, որը պետք է բացահայտվի

guessCard կարող է կատարել կանչեր հետևյալ ֆունկցիային.

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

- idx ՝ գույնաված ինդեքսը, $0 \leq idx < 52$
- Վերադարձնում է idx -ում գտնվող խաղաքարտի արժեքը
- **guess** կարող է կանչվել առավելագույնը T անգամ
- Երբ ճիշտը գույնակում է գնահատումը հաջողությամբ ավարտվում է

Փոխգործակցության օրինակ

Ներքևի օրինակը աշխատում է կցված գնահատող ծրագրի համար:

Առաջին տողը պարունակում է երկու թվեր՝ S -ը և T -ն:

Երկրորդ տողը պարունակում է 52 թվեր՝ որոնցից i -րդը նշանակում է i -րդ խաղաքարտի արժեքը:

Երրորդ տողը պարունակում է մեկ ամբողջ թիվ **target**-ը՝

Գնահատող ծրագրի մուտքի օրինակ	Կանչերի օրինակ		
	Կանչեր	Ենթականչեր	Վերադարձ
1 51	swapCards([0,1,...], 1, 51)		
0 1 2 3 4 5 6 7 8		doSwap(0, 1)	
9 10 11 12 13			տեղափոխել 0 և 1 ինդեքսների խաղաքարտերը
14 15 16 17 18			
19 20 21 22 23			
24 25 26 27 28	swapCards վերջացավ		.
29 30 31 32 33	guessCard(1, 51, 1)		

34 35 36 37 38		guess(5)	
39 40 41 42 43			guess-ը վերադարձնում է 5
44 45 46 47 48		guess(1)	
49 50 51			guess-ը վերադարձնում է 0
1		guess(0)	
			ճիշտ է!

Սահմանափակումներ

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

Ենթախնդիրներ

1. (16 միավոր): $S = 52, T = 1$
2. (20 միավոր): $S + T = 52$
3. (22 միավոր): $S = 13, T = 27$
4. (18 միավոր): $S = 1, T = 26$
5. (24 միավոր): Տրված S -ի և T -ի համար գոյություն ունի հաղթող մարտավարություն