

Card Trick

Два игрока собираются продемонстрировать карточный трюк со стандартной колодой из 52 карт. Для удобства значения карт будут заданы различными целыми числами от 0 до 51.

Карты изначально выложены на столе в один ряд лицом вверх (так, чтобы было видно значение) в каком-то порядке, неизвестном игрокам.

Первый игрок подходит к столу, смотрит на карты и делает обмены, в целом не более S раз. Каждый обмен заключается в выборе двух карт на позициях i и j (i и j могут быть равны) и перемещении карты с позиции i на позицию j и наоборот.

После этого первый игрок уходит, не общаясь со вторым игроком, и все карты поворачиваются рубашкой вверх (их значения больше не видны) не меняя порядок карт. Второй игрок приглашается к столу и у него просят попытаться угадать, где находится карта с **заданным** значением. Разрешается открыть не более T карт, одну за другой. Если одна из открытых карт является **заданной**, то игроки выиграли. Если попытки закончились, то они проиграли.

Вашей целью является написать две программы, имитирующие действия игроков, и выиграть игру.

Детали реализации

Вам будут предоставлены две программы - FirstPlayer and SecondPlayer вместе с примером грейдера.

В FirstPlayer Вы должны реализовать следующую функцию:

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

- Эта функция вызывается грейдером ровно один раз
- cards: массив, содержащий начальные значения карт слева направо, содержащий точно 52 элемента пронумерованных от 0 до 51
- S: разрешенное количество обменов
- T: разрешенное количество попыток угадать

swapCards может вызывать следующую функцию:

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

- i : индекс первой карты для обмена, $0 \leq i < 52$
- j : индекс второй карты для обмена, $0 \leq j < 52$
- **doSwap** должна вызываться не более S раз

В SecondPlayer Вам необходимо реализовать следующую функцию:

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

- S: количество обменов
- T: разрешенное количество попыток угадать
- target: заданное значение карты, которую необходимо открыть

guessCard может вызывать следующую функцию:

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

- idx: индекс угадываемой карты, $0 \leq idx < 52$
- функция возвращает значение карты с индексом idx
- **guess** может быть вызвана не более T раз.
- когда карта угадана верно, процесс угадывания успешно завершается

Пример взаимодействия

Ниже приводится пример работы прилагаемого грейдера.

Первая строка содержит два целых числа: S и T.

Вторая строка содержит 52 числа. i-е число задает значение i-й карты.

Третья строка содержит целое число, являющееся **заданным** значением карты.

Пример ввода для грейдера	Примеры вызовов		
	Вызовы	под-вызовы	Возвраты
1 51 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 1	swapCards([0,1,...], 1, 51)		
		doSwap(0, 1)	
			меняет карты с индексами 0 и 1
			swapCards закончила работу.
	guessCard(1, 51, 1)		
		guess(5)	
			guess возвращает 5

		guess(1)	
			guess возвращает 0
		guess(0)	
			Верно!

Ограничения

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

Подзадачи

1. (16 баллов): $S = 52, T = 1$
2. (20 баллов): $S + T = 52$
3. (22 балла): $S = 13, T = 27$
4. (18 баллов): $S = 1, T = 26$
5. (24 балла): Выигрышная стратегия существует для заданных S и T