Card Trick

Two players are going to demonstrate a card trick with a standard 52-card deck. For convenience card values will be distinct integers from 0 to 51.
Cards are initially placed on a table in a single row face up (with values visible) in some order unknown to the players.
The first player goes to the table, looks at the cards and does swaps, at most S times in total. Each swap is performed by choosing two cards on positions i and j (i and j can be equal) and moving the card from position i to position j and vice versa.
After that the first player leaves without communicating with the second player and all the cards are turned over (their values are no longer visible) without changing their order. The second player is invited to the table and is asked to guess where the card with **target** value is and is allowed to turn over at most Tcards one by one. If any of the revealed cards is **target**, then the players win. If they run out of guesses they lose.
Your goal is to write two programs that will simulate the actions of the players and win the game.

Implementation Details

You will be provided with two programs - FirstPlayer and SecondPlayer along with sample grader.

In FirstPlayer you have to implement the following function:

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

- This function is called exactly once by the grader
- cards: an array containing initial card values from left to right, with exactly 52 elements indexed from 0 through 51
- S: the number of allowed swaps
- T: the number of allowed guesses

`swapCards` can make calls to the following function:

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

- i: index of first card to swap, $0 \le i < 52$
- j: index of second card to swap, $0 \le j < 52$
- `doSwap` can be called at most S times

In SecondPlayer you have to implement the following function:

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

- S: the number of allowed swaps
- T: the number of allowed guesses
- target: the card value that should be revealed

`guessCard` can make calls to the following function:

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

- idx: guessed index, $0 \leq idx < 52$
- It returns the value of the idx-th card
- `guess` can be called at most T times.
- when the correct guess happens the evaluation terminates successfully

Sample interaction

Below is an example of input for the attached grader.
The first line should contain two integers: S and T.
The second line should contain 52 numbers. i-th one denoting the value of the i-th card.
The third like contains an integer **target**.

Sample Input to grader	Sample Calls		
	Calls	sub-calls	Returns
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)	
			swaps cards with indexes 0 and 1
	swapCards finishes		.
	guessCard(1, 51, 1)		
		guess(5)	
			guess returns 5
		guess(1)	
			guess returns 0
		guess(0)	
			Correct!

Constraints

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

Subtasks

1. (16 points): $\boldsymbol{S} = 52, \boldsymbol{T} = 1$
2. (20 points): $\boldsymbol{S} + \boldsymbol{T} = 52$
3. (22 points): $\boldsymbol{S} = 13, \boldsymbol{T} = 27$
4. (18 points): $\boldsymbol{S} = 1, \boldsymbol{T} = 26$
5. (24 points): Winning strategy exists for given $\boldsymbol{S}$ and $\boldsymbol{T}$