

## 1. Замки

1 секунда

100 очков

Искатель кладов оказался в пещере, единственный выход из которой закрыт несколькими замками. Чтобы выйти из пещеры, надо иметь ключи от всех замков. К счастью, искатель кладов имеет с собой карту пещеры, на которой отмечены места нахождения выхода и всех ключей.

Написать программу, которая, пользуясь картой пещеры и исходным положением искателя кладов, находит кратчайший путь, по которому он может собрать все ключи и дойти до выхода пещеры.

Каждым шагом искатель кладов может передвинуться на одну клетку карты на север, юг, восток или запад. Во время прохождения лабиринта любое доступное место (в том числе место нахождения выхода) можно посетить несколько раз, но проходить сквозь стены или выходить за пределы карты нельзя. Известно, что из любого прохода имеется к выходу ровно один самонепересекающийся путь.

**Входные данные.** В первой строке текстового файла `lukud.sis` размеры карты,  $N$  и  $M$  ( $1 \leq N, M \leq 100$ ). В каждой из следующих  $N$  строк ровно  $M$  символов: карта пещеры, на которой 0 обозначает проход, 1 стену, K место нахождения ключа, S исходную позицию искателя кладов и F место нахождения выхода. Известно, что все ключи находятся в проходах (не в стенах) и их не больше 1000.

**Выходные данные.** В единственную строку текстового файла `lukud.val` вывести инструкции для выхода из пещеры, в которых N обозначает шаг на север, S на юг, E на восток и W на запад. Если путей с минимальным числом шагов несколько, вывести любой из них.

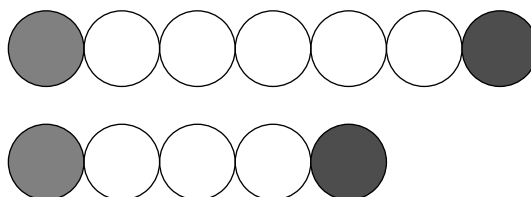
<b>Пример.</b>	<code>lukud.sis</code>	<code>lukud.val</code>
	3 3	NWESSWWN
	1K0	
	F1S	
	000	

## 2. Монетный бокс

1 секунда

100 очков

Монетный бокс — это настольная игра, в которую играют вдвоем на доске, состоящей из двух, трёх или четырёх полос. Каждая полоса состоит из нескольких кружков для монет. У игроков по одну монету на полосу, которые в начале игры поставлены в противоположные концы полосы, как видно (для двух полос) на рисунке.



Далее игроки ходят по очереди. За один ход игрок может передвинуть одну из своих монет на любое положительное число шагов либо в сторону монеты противника, либо от нее. Игрок не имеет права “прыгать” через монету противника, ставить свою монету на нее или выходить за пределы полосы. Если у игрока нет возможных ходов, он считается проигравшим.

Требуется написать программу для игры в монетный бокс, которая будет зачитывать состояние игры, а затем играть от этой позиции до конца партии.

При запуске программы на первой строке стандартного устройства ввода дано число полос,  $R$  ( $2 \leq R \leq 4$ ), затем — длины всех полос доски,  $N_1 \dots N_R$  ( $3 \leq N_i \leq 100$ ), а затем — число 1, если программа будет делать первый ход, либо число 2, если первый ход делает соперник.

После ввода начальной позиции первый игрок должен вывести на стандартное устройство вывода свой ход, затем ввести ход оппонента, вывести свой следующий ход и т.д. до конца партии. Второй игрок должен ввести первый ход оппонента, затем вывести свой ответ.

Ход “ $k$  шагов к сопернику на полосе  $N_i$ ” вывести в формате  $i\ k$ , ход “ $k$  шагов от соперника на полосе

$N_i$  вывести в формате  $i - k$ . Если ходить невозможно, вывести 0 0 и завершить работу программы. Если оппонент сделал ход 0 0, партия выиграна и программа должна немедленно завершить работу.

Пример.	Ввод	Вывод	Позиция
	2 7 5 1		1-----2 / 1---2
		1 5	-----12 / 1---2
	2 1		-----12 / 1--2-
		2 2	-----12 / --12-
	2 -1		-----12 / --1-2
		2 1	-----12 / ---12
	0 0		

**Оценивание.** Решения этой задании оценивают на турнире: за недопустимый ход, перерасход времени или завершение работы ошибкой программе дают 0 баллов, за корректно проигранную партию — 1 балл, а за выигранную партию — 3 балла. Лимит времени (1 секунда) ограничивает длину одной партии. Участник получает очки пропорционально результату своей программы на турнире, причём победитель турнира получает количество очков равное стоимости задания (100 очков).

### 3. Крыса

1 секунда

100 очков

Исследуем поведение крыс в лабиринте, построенном из единичных кубиков на клетчатом поле. Экспериментами установлено, что если поместить крысу в левую верхнюю клетку лабиринта, а кусок сыра — в правую нижнюю клетку, то крыса, следуя за запахом сыра, будет двигаться только направо (что мы обозначаем буквой E) или вниз (обозначаем буквой S).

Крыса может достигнуть сыра несколькими путями. Рассматриваем список всех возможных путей в алфавитном порядке. Написать программу, которая из карты лабиринта и порядкового номера пути находит описание этого пути.

**Входные данные.** В первой строке текстового файла `rott.sis` размеры лабиринта,  $N$  и  $M$  ( $1 \leq N, M \leq 30$ ). В каждой из следующих  $N$  строк ровно  $M$  символов: карта лабиринта, на которой 0 обозначает пустую клетку, а X — кубик (т.е. непроходимую клетку). В последней строке файла порядковый номер искомого пути,  $T$  ( $1 \leq T \leq 1\,000\,000\,000$ ). Можно полагать, что крыса имеет не менее  $T$  разных путей к сыру.

**Выходные данные.** В единственную строку текстового файла `rott.val` вывести описание пути номер  $T$ .

Пример.	rott.sis	rott.val
	4 5	ESSESEE
	00000	
	00X00	
	000X0	
	0X000	
	3	

Все возможные пути в альфавитном порядке: EEEESSS, EEESESS, ESSESEE, SESESEE, SSESEEE.