

딱지놀이

두 어린이 A, B가 딱지놀이를 한다. 딱지놀이 규칙은 다음과 같다. 두 어린이는 처음에 여러 장의 딱지를 가지고 있고, 매 라운드마다 각자 자신이 가진 딱지 중 하나를 낸다. 딱지에는 별(★), 동그라미(●), 네모(■), 세모(▲), 네 가지 모양 중 하나 이상의 모양이 표시되어 있다. 두 어린이가 낸 딱지 중 어느 쪽이 더 강력한 것인지는 다음 규칙을 따른다.

- 만약 두 딱지의 별의 개수가 다르다면, 별이 많은 쪽의 딱지가 이긴다.
- 별의 개수가 같고 동그라미의 개수가 다르다면, 동그라미가 많은 쪽의 딱지가 이긴다.
- 별, 동그라미의 개수가 각각 같고 네모의 개수가 다르다면, 네모가 많은 쪽의 딱지가 이긴다.
- 별, 동그라미, 네모의 개수가 각각 같고 세모의 개수가 다르다면, 세모가 많은 쪽의 딱지가 이긴다.
- 별, 동그라미, 네모, 세모의 개수가 각각 모두 같다면 무승부이다.

예를 들어, 두 어린이 A, B가 낸 딱지가 다음 그림과 같다고 하자.



위 규칙을 따르면 A의 딱지는 별 하나

를 가지고 있고 B의 딱지는 별이 없으므로 승자는 A이다. 위의 그림이 라운드 1의 상황이었고, 라운드 2, 3, 4, 5의 상황이 아래 표와 같을 때, 라운드 2, 3, 4의 승자는 각각 B, B, A이며, 라운드 5에서는 무승부가 되어 이를 D로 표현하였다.

라운드	A의 딱지	B의 딱지	결과
1	★	●●■▲	A
2	■★●■▲	★●●▲	B
3	●■▲▲	■●■▲	B
4	★●■▲	★●■	A
5	★★■●▲	★■★▲●	D

별, 동그라미, 네모, 세모를 각각 숫자 4, 3, 2, 1로 표현한다. 예를 들어, 라운드 1의 경우 어린이 A가 낸 딱지의 그림 ★는 4로 표현할 수 있고, 어린이 B가 낸 딱지의 그림 ●●■▲는 3 3 2 1로 표현할 수 있다.

라운드의 수 N 과 두 어린이가 순서대로 내는 딱지의 정보가 주어졌을 때, 각 라운드별로 딱지놀이의 결과를 구하는 프로그램을 작성하시오.

소스파일의 이름은 card.c 또는 card.cpp를 권장하지만, 서버에 제출하는 데는 다른 이름도 상관없다.

입력 형식

표준 입력으로 다음 정보가 주어진다. 첫 번째 줄에는 딱지놀이의 총 라운드 수를 나타내는 자연수 N 이 주어진다. N 은 1 이상 1,000 이하이다. 다음 줄에는 라운드 1에서 어린이 A가 내는

딱지에 나온 그림의 총 개수 a 가 주어진다. a 는 1 이상 100 이하이다. 뒤따라 나오는 a 개의 정수는 어린이 A가 낸 딱지의 그림을 나타내는데, 각각 4, 3, 2, 1 중 하나의 값이다. 4, 3, 2, 1의 순서대로 주어지지 않을 수 있음에 주의하라. 다음 줄에는 라운드 1에서 어린이 B가 내는 딱지에 나온 그림의 총 개수 b 가 주어진다. b 도 1 이상 100 이하이다. 뒤따라 나오는 b 개의 정수는 어린이 B가 낸 딱지의 그림을 나타내는데, 역시 4, 3, 2, 1 중 하나의 값이다. 역시 4, 3, 2, 1의 순서대로 주어지지 않을 수 있음에 주의하라. 다음 두 줄에는 라운드 2에서 어린이 A, B가 낸 딱지의 그림들을 같은 식으로 표현한다. 위와 같은 식으로 매 라운드마다 두 어린이가 낸 딱지의 정보는 두 줄에 표현되며, N 라운드의 딱지 정보는 차례대로 총 $2N$ 개의 줄에 주어진다.

출력 형식

표준 출력으로 총 N 줄을 출력한다. 출력의 i 번째 ($1 \leq i \leq N$) 줄에 정확히 한 글자를 출력하는데, 출력하는 글자는 A, B, D 중 하나로 라운드 i 의 결과를 나타낸다. 각 라운드의 결과는 A가 승자라면 A, B가 승자라면 B, 무승부라면 D이다.

부분문제의 제약 조건

- **부분문제 1:** 전체 점수 100점 중 2점에 해당하며 입력 예시로 주어진 입력만 존재한다.
- **부분문제 2:** 전체 점수 100점 중 13점에 해당하며 $a=1$ 이고 $b=1$ 로 가정한다.
- **부분문제 3:** 전체 점수 100점 중 17점에 해당하며 모든 딱지에는 별모양 그림만 나온다.
- **부분문제 4:** 전체 점수 100점 중 68점에 해당하며 원래의 제약조건 이외에 아무 제약조건이 없다.

입력과 출력의 예

입력(1)

```
5
1 4
4 3 3 2 1
5 2 4 3 2 1
4 4 3 3 1
4 3 2 1 1
4 2 3 2 1
4 4 3 2 1
3 4 3 2
5 4 4 2 3 1
5 4 2 4 1 3
```

출력(1)

```
A
B
B
A
D
```

입력(2)

```
4
4 4 3 2 1
4 1 4 3 2
4 3 3 2 1
4 4 3 3 3
4 4 3 3 3
4 3 4 3 2
4 3 2 1 1
3 3 2 1
```

출력(2)

```
D
B
A
A
```