

수학과 프로그래밍을 좋아하는 A와 B 두 사람이 다음과 같은 게임을 하고 있다.

둘은 각각 1 이상 30,000 미만의 수 하나를 고른다.

이 수를 가지고 점수를 계산하여 큰 쪽이 이기는 게임이다.

어떤 수의 점수는, 이 수부터 시작해서 한 자리를 골라 이 자리의 숫자를 지워서 소수를 만들고, 이 과정을 연속해서 최대로 많이 만들 수 있는 소수의 개수이다.

이 과정에는 입력받은 원래의 수도 포함된다.

예를 들어, 127은 자신이 소수이다. 만약 7을 지우면 12가 되고, 이는 소수가 아니다.

그러나, 7을 지우는 대신 2를 지워서 17, 다시 1을 지워서 7을 만들면 총 3개의 소수를 연속해서 만들 수 있으며,

127의 점수는 3, 즉 127에서 규칙을 지키면서 소수를 최대로 많이 만들 수 있는 경우라는 것을 쉽게 보일 수 있다.

어떤 숫자를 지우더라도 소수를 만들 수 없다면 종료한다.

A와 B가 제시한 수에서 규칙에 의해 소수의 수열을 만들었을 때, 더 높은 점수를 얻는 수를 제시한 쪽이 이긴다.

둘이 제시한 수의 점수가 같다면 무승부이다.

규칙상, 소수가 아닌 수를 낸다면 이 수를 낸 쪽이 지거나, 비기는 경우 두 가지만 가능하다

예를 들어, 128을 냈다면 이 수는 소수가 아니므로 이 수의 점수는 0점이고, 상대방도 합성수를 낸다면 비기고, 그렇지 않다면 지게 된다.

또한, 1은 소수가 아니다. A와 B가 제시하는 수의 어느 자리에도 0이 들어 있지 않다.

즉, 107과 같은 수를 제시할 수 없다.

A, B가 고른 두 수를 가지고 승패를 결정하는 프로그램을 작성하시오.

- 제한시간: 전체 테스트 케이스는 20개 이하이며, 전체 수행 시간은 1초 이내. (Java 2초 이내)

제한 시간을 초과하면 제출한 소스코드의 프로그램이 즉시 종료되며,

그때까지 수행한 결과에서 테스트 케이스를 1개 그룹 이상 통과하였더라도 점수는 0점이 됩니다.

그러나, 제한 시간을 초과하더라도 테스트 케이스를 1개 그룹 이상 통과하였다면 '부분 점수(0; 점수; 만점)'를 받을 수 있으며,

이를 위해서는, C / C++ 에서 "printf 함수" 사용할 경우, 프로그램 시작부분에서 "setbuf(stdout, NULL);"를 한번만 사용하십시오.

C++에서는 "setbuf(stdout, NULL);"와 "printf 함수" 대신 "cout"를 사용하고, Java에서는 "System.out.println"을 사용하시면,

제한 시간을 초과하더라도 '부분 점수'를 받을 수 있습니다.

※ 언어별 기본 제공 소스코드 내용 참고

만약, 제한 시간을 초과하지 않았는데도 '부분 점수'를 받았다면, 일부 테스트 케이스를 통과하지 못한 경우입니다.

- 메모리 사용 제한 : heap, global, static 총계 256MB, stack 100MB

- 제출 제한 : 최대 10회 (제출 횟수를 반영하여 순위 결정 → 동점자의 경우 제출 횟수가 적은 사람에게 높은 순위 부여)

메모리 사용 제한

heap, global, static (총계) : 256MB

stack : 100MB

입력

입력 파일에는 여러 테스트 케이스가 포함될 수 있다.

파일의 첫째 줄에 테스트 케이스의 개수를 나타내는 자연수 T 가 주어지고,

이후 차례로 T 개의 테스트 케이스가 주어진다. ($1 \leq T \leq 20$)

각 테스트 케이스의 첫 줄에는 두 정수 A, B ($1 \leq A, B < 30,000$)이 주어지는데, 이는 각각 A 와 B 가 고른 수를 나타낸다.

- 점수 : 최대 10회 제출하여 취득한 각각의 점수 중에서 최대 점수 (만점 100점)

모든 테스트 케이스를 맞추었을 때 만점을 받는다.

출력

각 테스트 케이스의 답을 순서대로 표준출력으로 출력하여야 하며,

각 테스트 케이스마다 첫 줄에는 “Case #C”를 출력하여야 한다. 이때 C 는 테스트 케이스의 번호이다.

그 다음 줄에, 이 테스트 케이스에서 A 가 이기면 1, B 가 이기면 2, 무승부이면 3을 출력하여야 한다.

입출력예

입력

2

127 128

131 127

출력

Case #1

1

Case #2

3