

# 고장난 시계

시간 제한	1초
메모리 제한	512 MB
사용 가능 언어	C, C++
C++ 허용 STL	전체
C++의 경우 main 함수 내의 시작 지점에 다음 내용을 추가하여 cin, cout 의 입출력 속도를 개선할 수 있다. <div><code>ios_base::sync_with_stdio(false); cin.tie(nullptr); cout.tie(nullptr);</code></div> 단, 위 내용을 추가할 경우 cin, cout 만 사용해야 하며, scanf, printf 등 C 입출력과 혼용해서 사용하면 안 된다. C++ std::endl의 경우 출력 속도가 느리므로, cout<<endl; 대신 <b>cout&lt;&lt;"\n";</b> 을 사용하는 것을 권장한다.	

인하네 시계방에는 수리를 기다리고 있는 시계들이 여러 개 있다. 이 시계들은 완전히 멈추지는 않았지만, 원래의 시계보다 더 빠르게 가거나 느리게 간다. 단, 맞게 가는 시계도 있을 수 있다. 참고로, 인하네 시계방에서는 디지털 시계를 취급하지 않기 때문에, 모든 시계는 12 시간을 한 바퀴로 표시하는 아날로그 시계이다.

인하는 이 시계들을 모두 12시 정각에 맞춘 후, 모든 시계가 처음으로 12시 위치를 동시에 가리키게 되는 데 걸리는 시간을 확인해보려 한다. 실험을 위해, 인하는 각 시계의 바늘이 한바퀴를 돌아 12시 위치로 돌아오는데 걸리는 실제 시간을 미리 측정한 후, 모든 시계들을 12시 정각에 맞춰 놓았다. 그러나 이 시계들이 다시 12시 위치에 모두 모이는 순간까지 기다리는 것은 매우 지루한 일이므로, 당신에게 이 시간을 구하는 작업을 맡겼다. 각 시계의 바늘이 시계의 한바퀴를 돌아 12시 위치로 돌아오는 데 걸리는 실제 시간이 주어질 때, 모든 시계가 처음으로 12시 위치를 동시에 가리키게 되는 데 걸리는 시간을 계산하는 프로그램을 작성해보자.

예를 들어 2 개의 시계가 있고, 두 시계의 바늘이 12시 위치로 돌아오는 데 걸리는 시간이 각각 8 시간, 14 시간이라면, 56 시간동안 첫 번째 시계는 7 바퀴, 두 번째 시계는 4 바퀴 회전한 뒤 모두 12시 위치를 가리키게 되며, 이보다 빠른 시간에 12시 위치를 동시에 가리키는 일은 없다.

## 입력

첫 번째 줄에 테스트 케이스의 개수  $T$  ( $1 \leq T \leq 1,000$ )가 주어진다. 이후 각 테스트 케이스의 정보가 다음과 같이 주어진다.

- 첫 번째 줄에 인하네 시계방에 있는 시계들의 개수  $N$  ( $2 \leq N \leq 10$ )이 주어진다.
- 두 번째 줄에  $N$ 개의 자연수가 공백으로 구분되어 주어진다. 주어지는 자연수  $t_i$  ( $0 < t_i < 20$ )는 각 시계가 한바퀴 돌아 12시 위치로 돌아오는데 걸리는 실제 시간을 의미한다.

# 출력

실험을 시작한 이후로 모든 시계가 처음으로 12시 위치를 동시에 가리킬 때까지의 경과된 시간을 출력한다.

## 예제 입출력

예제 입력 1	예제 출력 1
2 3 1 2 3 2 8 14	6 56