



## 코딩과제 #40 (배점: 30점)

1. 설명:

어떤 수를 입력 받아서 그 수를 뒤집은 다음 뒤집어진 수를 원래의 수에 더하는 과정을 뒤집어서 더하기라고 부르자. 그 합이 회문(palindrome, 앞뒤 어느 쪽에서 읽어도 같은 말이 되는 어구, 예: eye, madam, 소주만병만주소)이 아니면 회문이 될 때까지 이 과정을 반복한다.

예를 들어 처음에 195에서 시작해서 다음과 같이 네 번 뒤집어서 더하기를 반복하면 9339라는 회문이 만들어진다.

195	786	1473	5214
<u>+591</u>	<u>+687</u>	<u>+3741</u>	<u>+4125</u>
786	1473	5214	9339

대부분의 정수는 이 방법을 몇 단계만 반복하면 회문이 된다. 하지만 예외도 있다. 회문을 찾을 수 없는 것으로 밝혀진 첫 번째 수는 196이다. 하지만 회문이 없다는 것이 증명된 적은 없다. 어떤 수가 주어졌을 때 회문이 있으면 출력하고, 그 회문을 찾기까지 뒤집어서 더하기를 반복한 횟수를 출력하는 프로그램을 작성하라. 테스트 데이터로 쓰이는 수는 모두 뒤집어서 더하기를 1,000번 미만 반복해서 회문을 찾을 수 있는 수이고, 그렇게 만들어진 회문은 4,294,967,295보다 크지 않다고 가정한다.

※ 유의사항: 본 코딩과제는 반드시 함수를 만들어야 함.

2. 입력:

자연수 n (100 ~ 4,294,967,295)

3. 출력:

반복횟수와 회문으로 판명된 수

4. 함수원형:

```
unsigned int palindromenum(unsigned int n)
{

}
```

5. 실행결과 예시:

[예시 #1]

Please enter a number: 195 ↵ // 사용자가 입력한 값  
4, 9339



[예시 #2]

Please enter a number: 265 ↵ // 사용자가 입력한 값  
5, 45254

[예시 #3]

Please enter a number: 750 ↵ // 사용자가 입력한 값  
3, 6666