

## 문제해결프로젝트 프로젝트 II

### <첫 번째 문제 제시> A to B 정수 바꾸기 (5점)

정수 A를 B로 바꾸려고 한다. 가능한 연산은 다음과 같은 두 가지밖에 없다.

- 2를 곱한다.
- 1을 숫자의 가장 오른쪽에 추가한다.

A를 B로 바꾸는 데 필요한 연산의 최소 횟수를 출력하는 프로그램을 작성하시오. (실행시간 0.1초 이하 @i7 3.8GHz CPU, 입력 받는 시간 제외)

#### <입력 조건>

입력의 첫 줄에는 A, B ( $1 < A < B < 10^9$ ) 값이 띄어쓰기로 구분되어 주어진다.

#### <출력 조건>

첫 번째 줄에 A를 B로 바꾸는 데 필요한 연산의 최소값에 1을 더한 값을 출력한다. 바꿀 수 없는 경우에는 -1을 출력한다.

두 번째 줄에 입력 받은 이후부터 결과 출력까지 걸린 실행시간을 초 단위로 출력한다.

#### <입력/출력 예시>

##### 예제 입력 1 복사

```
2 162
```

2 → 4 → 8 → 81 → 162

##### 예제 입력 2 복사

```
4 42
```

##### 예제 입력 3 복사

```
100 40021
```

100 → 200 → 2001 → 4002 → 40021

##### 예제 출력 1 복사

```
5
실행시간: xxx초
```

##### 예제 출력 2 복사

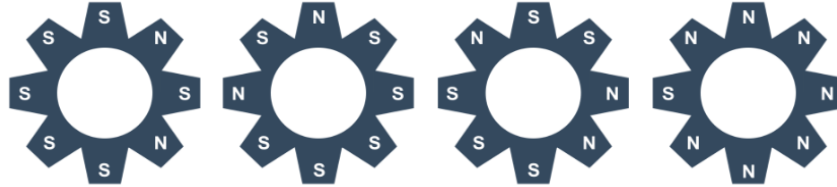
```
-1
실행시간: xxx초
```

##### 예제 출력 3 복사

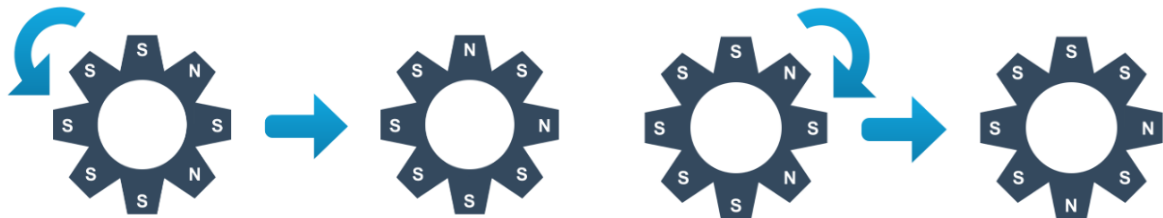
```
5
실행시간: xxx초
```

## <두 번째 문제 제시> 톱니바퀴 (5점)

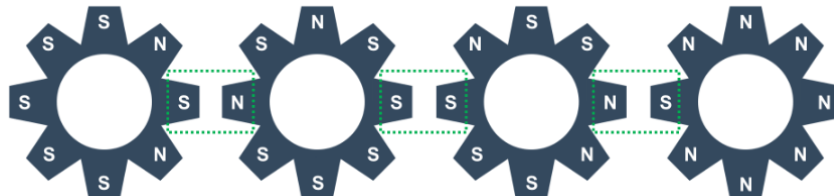
총 8개의 톱니를 가지고 있는 톱니바퀴 4개가 아래 그림과 같이 일렬로 놓여져 있다. 또 톱니는 N극 또는 S극 중 하나를 나타낸다. 톱니바퀴에는 번호가 매겨져 있는데, 가장 왼쪽 톱니바퀴가 1번, 그 오른쪽은 2번, 그 오른쪽은 3번, 가장 오른쪽 톱니바퀴는 4번이다.



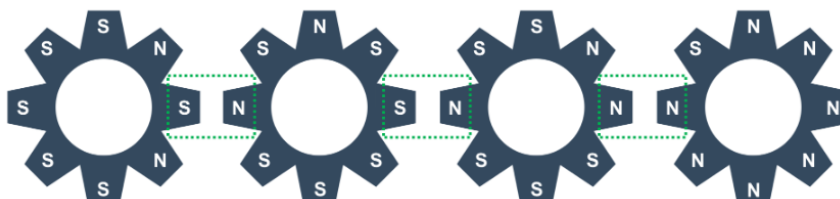
이 때, 톱니바퀴를 K번 회전시키려고 한다. 톱니바퀴의 회전은 한 칸을 기준으로 하고, 반시계방향과 시계방향으로 아래 그림과 같이 회전이 가능하다.



톱니바퀴를 회전시키려면, 회전시킬 톱니바퀴와 회전시킬 방향을 결정해야 한다. 톱니바퀴가 회전할 때, 서로 맞닿은 극에 따라서 옆에 있는 톱니바퀴를 회전시킬 수도 있고, 회전시키지 않을 수도 있다. 톱니바퀴 A를 회전할 때, 그 옆에 있는 톱니바퀴 B와 서로 맞닿은 톱니의 극이 다르면, B는 A가 회전한 방향과 반대방향으로 회전하게 된다. 아래의 경우를 살펴보자.



두 톱니바퀴의 맞닿은 부분은 녹색 점선으로 표기되어 있다. 여기서, 3번 톱니바퀴를 반시계방향으로 회전하면, 4번 톱니바퀴는 시계방향으로 회전한다. 2번 톱니바퀴는 맞닿은 부분의 극이 같기 때문에 회전하지 않게 되고, 1번 톱니바퀴는 2번이 회전하지 않기 때문에 회전하지 않게 된다. 따라서 아래의 그림과 같은 모양을 만든다.



톱니바퀴의 초기 상태와 톱니바퀴를 회전시키는 방법이 주어졌을 때, 최종 톱니바퀴의 상태를 구하는 프로그램을 작성하시오. (실행시간 0.2초 이하 @i7 3.8GHz CPU, 입력 받는 시간 제외)

### <입력 조건>

입력의 첫째 줄에 1번 톱니바퀴의 상태, 둘째 줄에 2번 톱니바퀴의 상태, 셋째 줄에 3번 톱니바퀴의 상태, 넷째 줄에 4번 톱니바퀴의 상태가 주어진다. 상태는 8개의 정수로 이루어져 있고, 12방향부터 시계방향 순서대로 주어진다. N극은 0, S극은 1로 주어진다.

입력의 다섯째 줄에는 회전 횟수(1~10)가 주어진다. 다음 K개 줄에는 회전시킨 방법이 순서대로 주어진다. 각 방법은 두 개의 정수가 공백으로 구분되어 있고, 첫 번째 정수는 회전시킨 톱니바퀴의 번호, 두 번째 정수는 회전방향을 나타낸다. 방향이 1인 경우는 시계방향이고, -1인 경우는 반시계방향이다.

### <출력 조건>

첫 번째 줄에 총 K번 회전시킨 이후에 네 톱니바퀴의 점수의 합을 출력한다. (아래 계산 참조)

- 1번 톱니바퀴의 12시 방향이 N극이면 0번, S극이면 1점
- 2번 톱니바퀴의 12시 방향이 N극이면 0번, S극이면 2점
- 3번 톱니바퀴의 12시 방향이 N극이면 0번, S극이면 4점
- 4번 톱니바퀴의 12시 방향이 N극이면 0번, S극이면 8점

두 번째 줄에 입력 받은 이후부터 점수 출력까지 걸린 실행시간을 초 단위로 출력한다.

### <입력/출력 예시>

#### 예제 입력 1 복사

```
10101111
01111101
11001110
00000010
2
3 -1
1 1
```

#### 예제 출력 1 복사

```
7
실행시간: xxx초
```

#### 예제 입력 2 복사

```
11111111
11111111
11111111
11111111
3
1 1
2 1
3 1
```

#### 예제 출력 2 복사

```
15
실행시간: xxx초
```

#### 예제 입력 3 복사

```
10001011
10000011
01011011
00111101
5
1 1
2 1
3 1
4 1
1 -1
```

#### 예제 출력 3 복사

```
6
실행시간: xxx초
```

#### <주의사항>

- Cpp 파일 2개(정수바꾸기.cpp, 톱니바퀴.cpp)를 과제 게시판에 업로드할 것 (그 외 파일은 허용 안됨, 압축파일 형태로 제출하지 말 것)
- Cpp 파일의 코드에 주석을 상세히 기입할 것 (예시: //변수명, 반복문, 조건문, 알고리즘 절차 설명)
- 필요한 모든 헤더 파일 및 함수를 cpp 파일에 포함시킬 것
- 띄어쓰기나 줄 바꿈에 주의할 것
- 수강생들간의 Copy 발견 시 모두 0점 처리함
- GNU Compiler Collection (g++ 9.2, clang++ 10.0) 범용 컴파일러에서 에러 없이 실행되어야 함.
- 실행시간 측정 방법, 컴파일러 설치 및 설정 방법은 첫 번째 프로젝트 문제 첨부파일 확인할 것.

#### <평가기준>

- 정수바꾸기, 톱니바퀴 문제에 대한 배점은 각 5점 만점으로 한다. (총 10점)
- 각 문제에 대한 평가 기준은 다음과 같이 정한다.
- 다양한 입력 테스트케이스에 대해서 프로그램 실행 시 출력 값이 모두 맞고, 실행시간 조건을 만족할 경우 5점 만점 처리함
- 다양한 테스트케이스에 대해서 프로그램 실행 시 출력 값이 모두 맞고, 실행시간 조건을 위반하거나 실행시간 출력이 없는 경우 4점 처리함
- 다양한 테스트케이스에 대해서 프로그램 실행 시 출력 값이 한 번이라도 틀린 경우 3점 처리함
- 컴파일 에러, 런타임 에러 등으로 인해 프로그램 실행이 안 될 경우 2점 처리함
- 주석 설명이 없거나 불충분하면 2점 처리함
- 핵심 구현 내용 부재 시 1점 처리함 (예시: 입력만 받고 처리에 대한 구현이 없는 경우)
- 기한 내 미제출하거나 Copy 발견시 0점 처리함