문제해결프로젝트 프로젝트 II

<첫 번째 문제 제시> 먹방 라이브 (5점)

평소 식욕이 왕성한 무지는 자신의 재능을 뽐내고 싶어 졌고 고민 끝에 카카오 TV 라이브로 방송을 하기로 마음먹었다.

그냥 먹방을 하면 다른 방송과 차별성이 없기 때문에 무지는 아래와 같이 독특한 방식을 생각해냈다. 회전판에 먹어야 할 N개의 음식이 있다. 각 음식에는 1부터 N까지 번호가 붙어있으며, 각 음식을 섭취하는데 일정 시간이 소요된다. 무지는 다음과 같은 방법으로 음식을 섭취한다.

- 무지는 1번 음식부터 먹기 시작하며, 회전판은 번호가 증가하는 순서대로 음식을 무지 앞으로 가져다 놓는다.
- 마지막 번호의 음식을 섭취한 후에는 회전판에 의해 다시 1번 음식이 무지 앞으로 온다.
- 무지는 음식 하나를 1초 동안 섭취한 후 남은 음식은 그대로 두고, 다음 음식을 섭취한다.
  - ■ 다음 음식이란, 아직 남은 음식 중 다음으로 섭취해야 할 가장 가까운 번호의 음식을 말한다.
- 회전판이 다음 음식을 무지 앞으로 가져오는데 걸리는 시간은 없다고 가정한다.

무지가 먹방을 시작한 지 k 초 후에 네트워크 장애로 인해 방송이 잠시 중단되었다. 무지가 네트워크 정상화 후 다시 방송을 이어갈 때, 몇 번 음식부터 섭취해야 하는지 출력하는 프로그램을 작성하시오. (실행시간 1초 이하 @i7 3.8GHz CPU, 입력 받는 시간 제외)

<입력 조건>
- 첫 번째 줄에 음식의 개수(N)이 주어진다. N은 1 이상 200,000 이하이다.
- 두 번째 줄에 각 음식을 모두 먹는데 필요한 시간이 음식의 번호 순서대로 공백으로 구분 되어 주어진다. 각 음식을 먹는 시간은 1 이상 100,000,000 이하의 자연수이다.
- 세 번째 줄에 먹방을 시작한 후 몇 초(k) 후에 방송이 중단되었는지 k 값이 주어진다. k는 1 이상 2×10^{13} 이하의 자연수이다.

<출력 조건>

- 첫 번째 줄에 네트워크 정상화 후 다시 방송을 이어갈 때, 몇 번 음식부터 먹어야 하는지 출력한다. 만약 더 섭취해야 할 음식이 없다면 -1을 출력한다.
- 두 번째 줄에 입력 받은 이후부터 결과 출력까지 걸린 실행시간을 초 단위로 출력한다.

<예제 입력>

```
3
3 1 2
5
```

<예제 출력>

```
1
실행시간: xxx초
```

<입출력 예 설명>

0~1초 동안에 1번 음식을 섭취한다. 남은 시간은 2 1 2 이다.

1~2초 동안 2번 음식을 섭취한다. 남은 시간은 2 0 2 이다.

2~3초 동안 3번 음식을 섭취한다. 남은 시간은 2 0 1 이다.

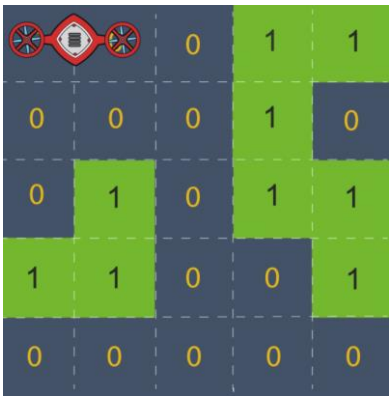
3~4초 동안 1번 음식을 섭취한다. 남은 시간은 1 0 1 이다.

4~5초 동안 (2번 음식은 다 먹었으므로) 3번 음식을 섭취한다. 남은 시간은 1 0 0 이다.

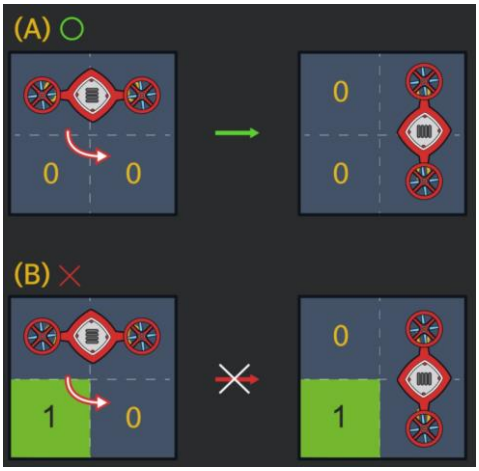
5초에서 네트워크 장애가 발생했다. 1번 음식을 섭취해야 할 때 중단되었으므로, 장애 복구 후에 1번 음식부터 다시 먹기 시작하면 된다.

<두 번째 문제 제시> 이동 로봇 (5점)

로봇개발자 전남이는 한 달 앞으로 다가온 로봇경진대회에 출품할 로봇을 준비하고 있다. 준비 중인 로봇은 2 x 1 크기의 로봇으로 전남이는 "0"과 "1"로 이루어진 N x N 크기의 지도에서 2 x 1 크기인 로봇을 움직여 (N, N) 위치까지 이동할 수 있도록 프로그래밍을 하려고 한다. 로봇이 이동하는 지도는 가장 왼쪽, 상단의 좌표를 (1, 1)로 하며 지도 내에 표시된 숫자 "0"은 빈칸을 "1"은 벽을 나타낸다. 로봇은 벽이 있는 칸 또는 지도 밖으로는 이동할 수 없다. 로봇은 처음에 아래 그림과 같이 좌표 (1, 1) 위치에서 가로방향으로 놓여있는 상태로 시작하며, 앞뒤 구분없이 움직일 수 있다.



로봇이 움직일 때는 현재 놓여있는 상태를 유지하면서 이동한다. 예를 들어, 위 그림에서 오른쪽으로 한 칸 이동한다면 (1, 2), (1, 3) 두 칸을 차지하게 되며, 아래로 이동한다면 (2, 1), (2, 2) 두 칸을 차지하게 된다. 로봇이 차지하는 두 칸 중 어느 한 칸이라도 (N, N) 위치에 도착하면 된다.
로봇은 다음과 같이 조건에 따라 회전이 가능하다.



위 그림과 같이 로봇은 90도씩 회전할 수 있다. 단, 로봇이 차지하는 두 칸 중, 어느 칸이든 축이 될 수 있지만, 회전하는 방향(축이 되는 칸으로부터 대각선 방향에 있는 칸)에는 벽이 없어야 한다. 로봇이 한 칸 이동하거나 90도 회전하는 데는 걸리는 시간은 정확히 1초이다.

"0"과 "1"로 이루어진 지도인 board가 주어질 때, 로봇이 (N, N) 위치까지 이동하는데 필요한 최소 시간을 출력하는 프로그램을 작성하시오. (실행시간 0.2초 이하 @i7 3.8GHz CPU, 입력 받는 시간 제외)

<입력 조건>
첫 번째 줄에 board의 한 변의 길이 N(5 ≤ N ≤ 100)이 주어진다.
그 다음 줄에 N줄에 걸쳐 N x N 개의 board 원소가 공백으로 구분되어 주어진다. 각 board 원소는 0 또는 1이다. 로봇이 처음에 놓여 있는 칸 (1, 1), (1, 2)는 항상 0으로 주어진다.

<출력 조건>
첫 번째 줄에 로봇이 (N, N) 위치까지 이동하는 데 필요한 최소 시간(초)을 출력한다. 만약 로봇이 (N, N) 위치로 이동할 수 없을 경우 -1을 출력한다.
두 번째 줄에 입력 받은 이후부터 결과 출력까지 걸린 실행시간을 초 단위로 출력한다.

<예제 입력>

```
5
0 0 0 1 1
0 0 0 1 0
0 1 0 1 1
1 1 0 0 1
0 0 0 0 0
```

<예제 출력>

```
7
실행시간: xxx초
```

<입출력 예 설명>
상기 그림 예시 참조바람.

<주의사항>

- Cpp 파일 2개(먹방라이브.cpp, 이동로봇.cpp)를 과제 게시판에 업로드할 것 (그 외 파일은 허용 안됨, 압축파일 형태로 제출하지 말 것)
- Cpp 파일의 코드에 주석을 상세히 기입할 것
- 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) 컴파일러에서 에러 없이 실행되어야 함.
- 실행시간 측정 방법, 컴파일러 설치 및 설정 방법은 첫 번째 프로젝트 첨부파일 참조

<평가기준>

- 먹방라이브, 이동로봇 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