**[제한 사항]**

|  |  |
| --- | --- |
| 시간 | 25개 테스트케이스를 합쳐서 C++ 의 경우 3초 / Java 의 경우 3초 |
| 메모리 | 힙, 정적 메모리 합쳐서 256MB 이내, 스택 메모리 1MB 이내 |

**※** 본 문제는 임직원의 S/W 문제해결 역량 향상을 위한 **Professional 실전형 연습문제**이며

함께 제공되는 **Testcase 또한 학습을 위한 연습용**입니다.

|  |
| --- |
| **①   C 또는 C++로 답안을 작성하시는 응시자께서는 검정시스템에 제출 시, Language 에서 C++ 를 선택하신 후 제출하시기 바랍니다.**  **②   Main 과 User Code 부분으로 구성되어 있습니다.**  **ㅇ   A.**Main         : 수정할 수 없는 코드이며, 채점 시 비 정상적인 답안 검출 등 평가를 위한 로직이 추가 될 수 있습니다.  **ㅇ   B.**User Code  : 실제 응시자가 작성해야 하는 코드이며, 제출 시에는 표준 입출력 함수가 포함되어 있으면 안 됩니다.  **③   Local PC 에서 프로그래밍 시 유의 사항**  **ㅇ   A.**2개의 파일을 생성하셔야 합니다. ( main.cpp / solution.cpp 또는 Solution.java / UserSolution.java )  **ㅇ   B.**Main 부분의 코드를 main.cpp 또는 Solution.java 에 복사해서 사용하시기 바랍니다.  **ㅇ   C.**sample\_input.txt 를 사용하시기 위해서는 Main 부분의 코드 내에  **ㅇ   C.**표준 입력을 파일로 전환하는 코드 ( 주석처리 되어 있음 ) 의 주석을 풀어서 사용하시면 됩니다.  **ㅇ   D.**User Code 부분의 코드를 작성하신 후 서버에 제출하실 때,  **ㅇ   D.**디버깅을 위한 표준 입출력 함수를 모두 삭제 또는 주석 처리해 주셔야 합니다.  **④   문제 내에 제약조건을 모두 명시하지 않으므로 주어지는 코드를 분석하셔야 합니다.**  **⑤   코드는 개발 언어에 따라 상이할 수 있으므로, 작성할 언어를 기준으로 분석하셔야 합니다.** |

**[문제 설명]**

영화제가 열리는 지역에 영화관이 N 개 있다. 각 영화관은 1 번부터 N 번까지 고유 번호를 가진다

N 개 영화관의 좌석을 예약하는 서비스를 시뮬레이션 하려고 한다.

예약을 같이한 친구들은 같은 영화관에서 상하좌우 중 한 방향 이상 연결된 좌석에 자리를 붙여서 예약을 하고 싶어한다. 좌석 예약할 때 이를 고려해야 한다.

한 개의 영화관에는 10 \* 10 개의 좌석이 있다. 좌석 번호는 왼쪽에서 오른쪽 방향으로, 앞에서 뒤쪽 방향으로 순서대로 번호가 붙어있다. 각 좌석의 번호는 [Fig. 1] 과 같다.

테이블이(가) 표시된 사진

자동 생성된 설명

영화관의 좌석 예약은 예약 ID 와 예약하는 좌석 개수 K 로 주어지고, 예약 방법은 다음과 같다.

1.**1 번 영화관부터 N 번 영화관까지 차례대로 아래 내용을 확인하여 예약을 한다.**( 한 개의 영화관이 선택되어 자리 예약이 된다. )

2. 예약하는 좌석들은 모두 상하좌우 중 한 방향 이상으로 서로 연결되어 있어야 한다.  
    만약, 영화관의 상하좌우 연결되는 좌석이 K 개 이상이 없어 좌석 예약을 할 수 없을 경우, 예약은 다음 영화관으로 넘어간다. ( 예제 Order 19. 참고 )

3. 상하좌우 연결되고, K 개 이상 비어 있는 좌석들을 하나로 묶어 빈좌석 묶음이라 한다.

4. 빈좌석 묶음이 여러 개 일 경우, 좌석 번호가 가장 작은 좌석이 포함된 빈좌석 묶음을 선택한다.

5. 첫번째 예약 좌석은 선택한 빈좌석 묶음에서 번호가 가장 작은 좌석이다.

6. 다음 예약 좌석은 새롭게 예약된 좌석들과 상하좌우 중 한 방향 이상으로 연결되어야 하며, 그 중 번호가 가장 작은 비어 있는 좌석이다. ( [Fig. 2] 번호 순서 참고 )

7. K 개의 좌석을 모두 예약할 때까지 6번을 반복한다.

[Fig. 2] 와 같이 한 개의 영화관은 10 \* 10 좌석으로 되어 있다. (각각의 좌석 번호는 [Fig. 1] 참고)

회색은 이미 예약이 되어 예약 불가한 자리이고, 흰색은 예약 가능한 자리이다.

15개의 좌석을 예약할 경우 아래와 같은 순서대로 15개의 좌석이 예약된다. (첫번째 예약 좌석의 번호는 18 이다.)

**좌석 번호가 1 ~ 4 인 좌석은 상하좌우 연결된 예약 가능한 좌석이 4 개만 있기 때문에 제외한다.**

테이블이(가) 표시된 사진

자동 생성된 설명

위와 같은 좌석 예약 서비스 시뮬레이션 프로그램을 구현하라.

아래 API 설명을 참조하여 각 함수를 구현하라.

※ 아래 함수 signature는 C/C++에 대한 것으로 Java에 대해서는 제공되는 Solution.java와 UserSolution.java를 참고하라.

아래는 User Code 부분에 작성해야 하는 API 의 설명이다.

|  |
| --- |
| **void init(int N)** |
| 각 테스트 케이스의 처음에 호출된다.    1 번부터 N 번까지 N 개의 영화관이 있다. (한 개의 영화관은 10 \* 10 개의 좌석이 있다.)    N 개의 영화관은 모두 비어 있어 예약 가능하다.      ***Parameters***    N : 영화관의 개수 ( 1 ≤ N ≤ 2,000 ) |
| **Result reserveSeats(int mID, int K)** |
| mID 예약 번호로 K 개의 좌석을 예약하고, 아래의 값을 반환한다.  Result 구조 내의 id = 예약된 영화관 번호, num = 예약 좌석 중에서 가장 작은 좌석 번호를 반환한다.    모든 영화관이 K 개 이상의 상하좌우 연결되는 좌석이 없어 예약이 실패하는 경우, Result 구조 내의 id = 0, num = 0 을 반환한다.    좌석 예약 방법은 본문 설명과 같다.    mID 예약 번호는 이전에 해당 함수에서 호출된 적이 없다.      ***Parameters***    mID : 예약 번호 ( 1 ≤ mID ≤ 50,000 )    K    : 좌석 예약 개수 ( 1 ≤ K ≤ 50 )    ***Returns***    Result 구조 내의 id = 예약된 영화관 번호, num = 예약 좌석 중에서 가장 작은 좌석 번호    예약 실패할 경우 Result 구조 내의 id = 0, num = 0 |
| **Result cancelReservation(int mID)** |
| mID 예약 번호로 예약된 좌석을 모두 취소하고, 아래의 값을 반환한다.  Result 구조 내의 id = 취소된 영화관 번호, num = 취소된 좌석 번호를 모두 더한 값을 반환한다.    mID 예약 번호로 예약된 좌석이 있음을 보장한다.      ***Parameters***    mID  : 취소하려는 예약 번호 ( 1 ≤ mID ≤ 50,000 )    ***Returns***    Result 구조 내의 id = 취소된 영화관 번호, num = 취소된 좌석 번호를 모두 더한 값 |

**[예제]**

아래에는 테스트케이스 1번을 처음부터 순서대로 보여준다.

|  |  |  |  |
| --- | --- | --- | --- |
| **Order** | **Function** | **Return** | **Figure** |
| 1 | init(3) |  |  |
| 2 | reserveSeats(1, 17) | id = 1, num = 1 |  |
| 3 | reserveSeats(21, 2) | id = 1, num = 18 |  |
| 4 | reserveSeats(33, 2) | id = 1, num = 20 |  |
| 5 | reserveSeats(14, 2) | id = 1, num = 21 |  |
| 6 | reserveSeats(15, 4) | id = 1, num = 23 |  |
| 7 | reserveSeats(65, 3) | id = 1, num = 27 |  |
| 8 | reserveSeats(27, 2) | id = 1, num = 31 |  |
| 9 | reserveSeats(58, 2) | id = 1, num = 33 |  |
| 10 | reserveSeats(39, 4) | id = 1, num = 35 |  |
| 11 | cancelReservation(21) | id = 1, num = 37 |  |
| 12 | cancelReservation(14) | id = 1, num = 43 |  |
| 13 | cancelReservation(65) | id = 1, num = 84 |  |
| 14 | cancelReservation(58) | id = 1, num = 67 |  |
| 15 | reserveSeats(22, 15) | id = 1, num = 18 | [Fig. 3] |

**[Table 1]**

Order 15. reserveSeats(22, 15) 에서 좌석 예약은 [Fig. 3] 과 같다. (검은색 숫자는 15 개 좌석 예약할 때 예약 순서이다.)

회색은 이미 예약이 되어 예약 불가한 자리이고, 흰색은 예약 가능한 자리이다.

테이블이(가) 표시된 사진

자동 생성된 설명

 [Table 2] 는 [Table 1] 이후의 함수 호출이다.

|  |  |  |  |
| --- | --- | --- | --- |
| **Order** | **Function** | **Return** | **Figure** |
| 16 | reserveSeats(111, 13) | id = 1, num = 41 |  |
| 17 | cancelReservation(15) | id = 1, num = 98 |  |
| 18 | cancelReservation(39) | id = 1, num = 146 |  |
| 19 | reserveSeats(100, 48) | id = 2, num = 1 |  |
| 20 | reserveSeats(51, 2) | id = 1, num = 21 | [Fig. 4] |

**[Table 2]**

Order 16. reserveSeats(111, 13) 에서 13개의 좌석을 예약하기 때문에 1 번 영화관의 21 ~ 22 좌석은 상하좌우 연결된 예약 가능한 좌석이 2개밖에 없어 제외되고, 41 좌석이 첫번째 예약 좌석이 된다.

Order 19. reserveSeats(100, 48) 에서 48개의 좌석을 예약하기 때문에 1 번 영화관은 비어있는 자리가 51 개 이지만, 상하좌우 연결된 좌석이 최대 40개이기 때문에 제외된다.

            48 개 이상의 상하좌우 연결된 예약 가능한 좌석이 있는 2번 영화관 1 좌석이 첫번째 예약 좌석이 된다.

Order 20. reserveSeats(51, 2) 실행 후 3개 영화관의 좌석은 [Fig. 4] 과 같다. (회색 좌석이 예약된 좌석이다.)

쇼지, 낱말맞추기게임이(가) 표시된 사진

자동 생성된 설명

 [Table 3] 은 [Table 2] 이후의 함수 호출이다.

|  |  |  |  |
| --- | --- | --- | --- |
| **Order** | **Function** | **Return** | **Figure** |
| 21 | reserveSeats(110, 18) | id = 1, num = 61 |  |
| 22 | reserveSeats(511, 50) | id = 2, num = 49 |  |
| 23 | reserveSeats(32, 50) | id = 3, num = 1 |  |
| 24 | reserveSeats(1133, 1) | id = 1, num = 23 |  |
| 25 | reserveSeats(447, 37) | id = 3, num = 51 |  |
| 26 | cancelReservation(100) | id = 2, num = 1176 |  |
| 27 | reserveSeats(69, 50) | id = 0, num = 0 |  |
| 28 | reserveSeats(59, 19) | id = 1, num = 79 | [Fig. 5] |

**[Table 3]**

Order 28. reserveSeats(59, 19) 실행 후 3개 영화관의 좌석은 [Fig. 5] 와 같다. (회색 좌석이 예약된 좌석이다.)

텍스트, 쇼지이(가) 표시된 사진

자동 생성된 설명

**[제약사항]**

1. 각 테스트 케이스 시작 시 init() 함수가 호출된다.

2. reserveSeats() 에서 좌석 예약은 한 개의 영화관에서 예약 되어야 한다.

3. 각 테스트 케이스에서 reserveSeats() 함수의 호출 횟수는 20,000 이하이다.

4. 각 테스트 케이스에서 cancelReservation() 함수의 호출 횟수는 10,000 이하이다.

**[입출력]**

입출력은 제공되는 Main 부분의 코드에서 처리하므로 User Code 부분의 코드에서는 별도로 입출력을 처리하지 않는다.

Sample input 에 대한 정답 출력 결과는 “#TC번호 결과” 의 포맷으로 보여지며 결과가 100 일 경우 정답, 0 일 경우 오답을 의미한다.