

고장난 청소 로봇

시간 제한	1초
메모리 제한	512 MB
사용 가능 언어	C, C++
C++ 허용 STL	전체

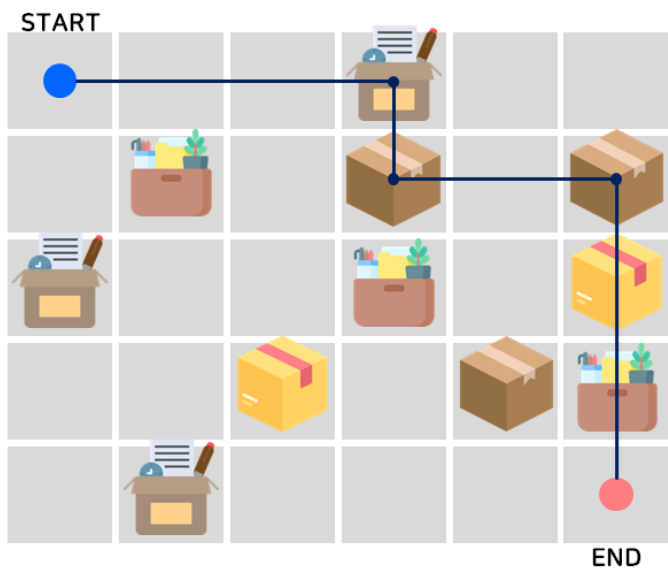
C++의 경우 main 함수 내의 시작 지점에 다음 내용을 추가하여 cin, cout 의 입출력 속도를 개선할 수 있다.

```
ios_base::sync_with_stdio(false);
cin.tie(nullptr);
cout.tie(nullptr);
```

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 **cout<<"\n";**을 사용하는 것을 권장한다.

인하는 물품 보관 창고로 쓰이던 공간을 새로운 작업실로 쓰기 위하여 중고 청소 로봇을 구매하였다. 창고는 세로 N , 가로 M 크기의 격자 모양으로 구성되어 있고, 각 격자는 기존에 보관 중이던 물품이 채워져 있거나 비어 있다. 인하는 청소 로봇을 이용하여 창고의 모든 격자에 있는 물건을 치우려고 한다.

인하가 구매한 청소 로봇은 시작 위치와 종료 위치를 정해주면 시작 위치부터 자동으로 청소를 하며 종료 위치까지 이동하는 로봇이다. 인하는 청소 로봇의 시작 위치를 창고의 가장 왼쪽 위 공간(1, 1)으로, 종료 위치를 가장 오른쪽 아래 공간(N , M)으로 설정하고 작동 버튼을 눌렀다. 그런데 중고 청소 로봇은 고장이 난 상태였고, 인하는 로봇 작동과 동시에 오른쪽 또는 아래쪽으로 밖에 이동을 못하는 문제점을 발견하게 되었다. 어쩔 수 없이 인하는 청소 로봇이 최대한 많은 물품을 치우도록 수동으로 경로를 설정하고, 나머지 물품들은 직접 치우기로 결정하였다. 물품이 배치되어 있는 위치의 정보가 주어졌을 때, 인하가 직접 치워야 하는 물품의 수의 최솟값을 구해보자.



예를 들어, 위 그림과 같이 5×6 크기의 창고에 총 11 개의 물품이 있을 때, 청소 로봇이 오른쪽과 아래로만 이동하여 위 그림의 경로와 같이 이동할 경우 5 개의 물품을 치울 수 있고, 이때 인하가 직접 치워야 하는 물품은 6 개이며 이 경우가 인하가 치워야 하는 물품의 수가 최소인 경우이다.

입력

첫 번째 줄에는 테스트 케이스의 수 T ($1 \leq T \leq 100$)가 주어진다. 이후 각 테스트 케이스의 정보가 다음과 같이 주어진다.

- 첫 번째 줄에 창고의 세로 길이 N ($1 \leq N \leq 500$), 가로 길이 M ($1 \leq M \leq 500$)이 공백으로 구분되어 주어진다.
- 두 번째 줄부터 N 개의 줄에 걸쳐 물품의 존재 여부를 나타내는 M 개의 정수가 공백으로 구분되어 주어진다. 정수는 0 또는 1로 주어지며, 0인 경우 해당 격자에 물품이 존재하지 않고, 1인 경우 물품이 존재함을 의미한다.

출력

각 테스트 케이스마다 인하가 직접 치워야 하는 물품 수의 최솟값을 한 줄씩 출력한다.

예제 입출력

예제 입력	예제 출력
3 5 6 0 0 0 1 0 0 0 1 0 1 0 1 1 0 0 1 0 1 0 0 1 0 1 1 0 1 0 0 0 0 2 2 1 1 1 0 4 4 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	6 1 9