※ 문제 및 풀이에 대해 **사외 온라인/오프라인에 게시/공유 하는 것은 허용**되어 있습니다.

SW Expert 아카데미의 유저 문제는 **모든 사람**에게 오픈 되어 있습니다.

KSFS: KennySoft File System

**[제한 사항]**

|  |  |
| --- | --- |
| 시간 | 9개 테스트케이스를 합쳐서 C/C++의 경우 5초 |
| 메모리 | 힙, 정적 메모리 합쳐서 256MB 이내, 스택 메모리 1MB 이내 |

KENNYSOFT에 다니는 현민이는 이번에 새로운 파일 시스템을 만들어 보기로 했다: 그 이름은 KennySoft의 File System이라 하여 KSFS로 정했다.

그런데 정작 파일의 용량이 커지면 어떻게 섹터를 나누고 파편화에 대해서는 어떻게 처리해야 할지 등의 생각을 하지 않은 현민이는 먼저 폴더 구조만을 구축해 보기로 했다.

한편 그 파일 시스템조차도 구축하는 것이 너무 귀찮았던 현민이는 관심을 돌려 파일 시스템을 테스트하기 위한 간단한 셸을 먼저 만들었다. 그리고 4가지 종류의 명령을 준비하려고 한다.

* mkdir은 폴더를 생성하기 위한 명령어이다. 인자로는 폴더의 이름이 주어진다. 문자열은 알파벳 소문자로만 구성되어 있으며, 12자리 이하이다. 현재 위치한 경로에서 해당 폴더의 이름을 만들되, 같은 부모 아래에 동일한 이름의 폴더가 이미 있다면 생성하지 않는다. 이 때, 폴더를 생성하는 작업이 성공했는지 여부를 1(성공) 또는 0(실패)의 형태로 반환해야 한다. 이 명령은 최대 20,000회 실행된다.
* cd는 현재 위치를 이동하기 위한 명령어이다. 인자로는 세 가지 종류가 주어질 수 있다. 각 종류의 인수에 따라 현재 위치를 이동하는 작업이 성공했는지 여부를 1(성공) 또는 0(실패)의 형태로 반환해야 한다. 이 명령은 최대 10,000회 실행된다.
  + / : 루트로 이동한다. 현재 어디에 있었든지 성공한 것으로 간주한다.
  + .. : 상위 폴더로 이동한다. 이미 루트에 있었다면 실패한 것으로 간주한다.
  + (문자열)\t : 주어진 문자열로 시작하는 하위 폴더들 중 사전 순으로 가장 빠른 폴더를 선택하여 해당 폴더로 이동한다. 문자열은 비어 있을 수도 있는데, 이는 모든 하위 폴더들 중 사전 순으로 가장 빠른 폴더로 이동하라는 뜻이다. 주어진 문자열로 시작하는 이름을 가진 폴더가 없다면 이동하지 않고, 이 명령은 실패한 것이다.
* rm은 폴더를 삭제하기 위한 명령어이다. 인자로는 (문자열)\* 형태가 주어진다. 이는 주어진 문자열로 시작하는 모든 하위 폴더와 그들의 모든 자식 폴더를 지우라는 뜻이다. 역시 문자열은 비어 있을 수 있다. 이 명령은 동작 후 지워진 폴더의 개수를 반환해야 한다. 이 명령은 최대 10,000회 실행된다.
* ls는 폴더 내의 내용을 확인하기 위한 명령어이다. 한편 하위 폴더가 굉장히 많을 수 있으므로 단순히 하위 폴더의 개수만을 구해 보기로 하였다. 인자로는 (문자열)\* 형태가 주어지고, 동일하게 문자열은 비어 있을 수 있으며, 주어진 문자열로 시작하는 모든 하위 폴더와 그들의 모든 자식 폴더의 개수를 합한 값을 반환해야 한다. 이 명령은 최대 200회 실행된다.

FAT, NTFS, ReFS, HFS+, APFS 등을 누르고 KSFS가 새로운 파일 시스템의 시대를 열 수 있도록 현민이가 파일 시스템을 구축하는 데에 도움을 주자.

**[입력]**

첫 번째 줄에 테스트 케이스의 수 T가 주어진다.

각 테스트 케이스의 첫 번째 줄에는 두 정수 S, N(0 ≤ S < 231, 1 ≤ N ≤ 40200)이 공백 하나로 구분되어 주어진다. S는 0이 아닌 경우에는 랜덤 데이터를 만들기 위한 시드 값이다.

S가 0인 경우에는 미리 제작된 데이터를 사용하는 경우로, 두 번째 줄부터 N개의 줄에 각 명령이 주어진다. 각 명령의 형태는 아래와 같다. X는 다른 것으로 처리되지 않은 임의의 문자열(알파벳 소문자, 12자리 이하)을 의미한다.

* 1 X : mkdir X
* 2
  + 2 / : cd /
  + 2 .. : cd ..
  + 2 0 : cd \t
  + 2 X : cd X\t
* 3
  + 3 0 : rm \*
  + 3 X : rm X\*
* 4
  + 4 0 : ls \*
  + 4 X : ls X\*

**[출력]**

각 테스트 케이스마다 "#X" (X는 테스트 케이스 번호를 의미하며 1부터 시작한다)를 출력하고 다음 줄부터 N개의 줄에 각 명령의 실행 결과를 출력한다.

|  |  |
| --- | --- |
| 입력 예제 | |
| 9  0 37  1 selc  1 selc  2 selc  1 kenny  1 soft  1 softicecream  1 software  1 softball  1 softech  1 sofrano  1 sofa  1 sofia  1 softener  4 soft  2 sec  2 soft  1 sec  1 samsung  2 ..  4 s  4 softe  2 softe  3 sec  1 sec  4 0  3 sec  4 0  1 sec  4 0  2 ..  4 softe  3 softe  2 softe  4 s  2 /  2 ..  3 se  0 37  1 selc  1 selc  2 selc  1 kenny  1 soft  1 softicecream  1 software  1 softball  1 softech  1 sofrano  1 sofa  1 sofia  1 softener  4 soft  2 sec  2 soft  1 sec  1 samsung  2 ..  4 s  4 softe  2 softe  3 sec  1 sec  4 0  3 sec  4 0  1 sec  4 0  2 ..  4 softe  3 softe  2 softe  4 s  2 /  2 ..  3 se  20180528 50  … | // 테스트 케이스 개수  // 첫 번째 테스트 케이스의 S N  // mkdir selc (성공: 1)  // mkdir selc (실패: 0)  // cd selc\t (성공: 1, 현재 경로: /selc/)  // mkdir kenny (성공: 1)  // mkdir soft (성공: 1)  // mkdir softicecream (성공: 1)  // mkdir software (성공: 1)  // mkdir softball (성공: 1)  // mkdir softech (성공: 1)  // mkdir sofrano (성공: 1)  // mkdir sofa (성공: 1)  // mkdir sofia (성공: 1)  // mkdir softener (성공: 1)  // ls soft\* (결과: 9[[1]](#endnote-1))  // cd sec\t (실패: 0)  // cd soft\t (성공: 1, 현재 경로: /selc/soft/)  // mkdir sec (성공: 1)  // mkdir samsung (성공: 1)  // cd .. (성공: 1, 현재 경로: /selc/)  // ls s\* (결과: 11[[2]](#endnote-2))  // ls softe\* (결과: 2[[3]](#endnote-3))  // cd softe\t (성공: 1, 현재 경로: /selc/softech/)  // rm sec\* (결과: 0)  // mkdir sec (성공: 1)  // ls \* (결과: 1[[4]](#endnote-4))  // rm sec\* (결과: 1[[5]](#endnote-5))  // ls \* (결과: 0)  // mkdir sec (성공: 1)  // ls \* (결과: 1[[6]](#endnote-6))  // cd .. (성공: 1, 현재 경로: /selc/)  // ls softe\* (결과: 3[[7]](#endnote-7))  // rm softe\* (결과: 3[[8]](#endnote-8))  // cd softe\t (실패: 0)  // ls s\* (결과: 9[[9]](#endnote-9))  // cd / (성공: 1, 현재 경로: /)  // cd .. (실패: 0)  // rm se\* (결과: 11[[10]](#endnote-10))  // 두 번째 테스트 케이스의 S N  // 이하 첫 번째 테스트 케이스와 동일  // 세 번째 테스트 케이스의 S N; 랜덤 데이터 사용 |

1. soft, softicecream, software, softball, softech, sofrano, sofa, sofia, softener [↑](#endnote-ref-1)
2. soft, soft/sec, soft/samsung, softicecream, software, softball, softech, sofrano, sofa, sofia, softener [↑](#endnote-ref-2)
3. softech, softener [↑](#endnote-ref-3)
4. sec [↑](#endnote-ref-4)
5. sec [↑](#endnote-ref-5)
6. sec [↑](#endnote-ref-6)
7. softech, softech/sec, softener [↑](#endnote-ref-7)
8. softech, softech/sec, softener [↑](#endnote-ref-8)
9. soft, soft/sec, soft/samsung, softicecream, software, softball, sofrano, sofa, sofia [↑](#endnote-ref-9)
10. selc, selc/kenny, selc/soft, selc/soft/sec, selc/soft/samsung, selc/softicecream, selc/software, selc/softball, selc/sofrano, selc/sofa, selc/sofia [↑](#endnote-ref-10)