Problem Statement Fox Ciel is going to walk through an unpaved road to meet her friend. The road is one-dimensional. It is separated into N parts numbered from 0 to N-1, where part 0 is Ciel's current part and part N-1 is her destination part.Ciel will perform her trip tomorrow. Unfortunately today it is raining, so tomorrow some parts of the road will be muddy. You are given a vector <int> road containing N elements. The probability that the i-th part of the road will be muddy tomorrow is road[i]/100.Ciel can walk along the road using any combination of strides with lengths 1 and 2. If she is at part i, a stride of length 1 will move her to part i+1 and a stride of length 2 will move her to part i+2 (skipping part i+1). If there are many ways to reach part N-1 from part 0, Ciel will choose the one among them where the number of visited muddy parts is minimal.Return the expected number of muddy parts that she will visit tomorrow.Definition Class: MuddyRoadMethod: getExpectedValueParameters: vector <int>Returns: doubleMethod signature: double getExpectedValue(vector <int> road)(be sure your method is public) Notes- Assume that events "i-th part of the road will be muddy tomorrow" are totally independent.- Ciel has very good sight, so when starting her trip at part 0, she is already able to see for each part whether it is muddy or not.Constraints- road will contain between 2 and 50 elements, inclusive.- Each element of road will be between 0 and 100, inclusive.- The first element and the last element of road will be 0.Examples0) {0, 60, 60, 0}Returns: 0.36There can be four different states of the road tomorrow:.... with probability = 0.16, 0 steps to muddy parts.M.. with probability = 0.24, 0 steps to muddy parts..M. with probability = 0.24, 0 steps to muddy parts.MM. with probability = 0.36, 1 step to muddy parts(Here, '.' represents a non-muddy part and 'M' represents a muddy part.)Thus, the expected number of steps is 0\*0.16+0\*0.24+0\*0.24+1\*0.36=0.36.1) {0, 50, 50, 50, 50, 0}Returns: 0.56252) {0, 0, 100, 100, 100, 100, 100, 100, 0, 0, 100, 0}Returns: 3.03) {0, 12, 34, 56, 78, 91, 23, 45, 67, 89, 0}Returns: 1.73525394200319234) {0, 50, 50, 100, 50, 100, 50, 50, 100, 66, 0}Returns: 2.288125This problem statement is the exclusive and proprietary property of TopCoder, Inc. Any unauthorized use or reproduction of this information without the prior written consent of TopCoder, Inc. is strictly prohibited. (c)2003, TopCoder, Inc. All rights reserved.