① First, we have extracted information from an input file and created a graph using its edges and weights. We have used Dijkstra's algorithm to solve this problem. We maintained a distance array where the source node is 0 and others are initially set to infinity. We also created a visited array and a priority queue. If the sum of the distance of a node and the cost to go to a vertex is less than the distance of that vertex in the distance array, then we update it in the distance array. Finally, we have written the distance of each node in an output file.

② We solved this problem in the same manner as problem 1. However, in this problem, we used two source points, one for Alice and another for Bob. This is why we have two distance arrays. Then, we compare the maximum distance of the same location visited by the ~~visited~~ two sources and store it along with the corresponding node. Finally, we write this information to an output file.

③ In this problem, we have used the same Dijkstra function as in Task 1, but with a slight modification. Instead of calculating the sum of the node's distance and the cast to go to a vertex, we find the maximum value between the distance of the node and the cost to go to a vertex, and update it if it is less than the existing distance, in order to determine the safest path.