The Elevator Problem

Imagine a high-rise building that has 12 floors and exactly one elevator. The elevator can travel to all floors in the building and it receives commands in the following format:

<start floor> - <end floor>

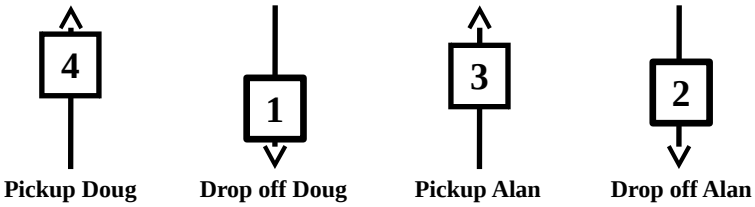
For example, if the elevator receives the commands 4-1, 3-2, 1-5, and 6-8 it would mean that someone will get on at floor 4 and off at floor 1, someone else will get on at floor 3 and off at floor 2, and so on. In addition, the elevator can operate in two unique modes (described below), which specify the priority and optimizations that the elevator should use when completing a set of commands.

Your task is to write a program that requires an argument for the name of a text file containing multiple sets of commands as an input, and for each set of commands writes to standard output both the path that the elevator will take and the total distance in floors that the elevator must travel. In addition, your program must accept an argument to specify which mode the elevator will operate in throughout the application lifecycle. The mode argument should *follow* the filename argument.

Your program must support both of the following modes.

Mode A

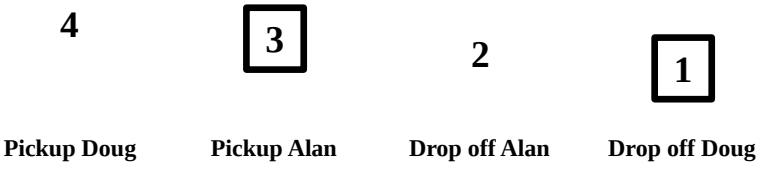
Mode A is very inefficient and only allows for transporting one person at a time. If Doug calls the elevator on floor 4 and wants to go to floor 1, and Alan subsequently calls the elevator on floor 3 and wants to go to floor 2, the elevator will go pick up Doug and take



him to floor 1 before picking up Alan and taking him to floor 2.
In this example the system receives the commands 4-1, 3-2 and tells the elevator to travel from its current floor to floors 4, 1, 3, and 2. If the elevator starts at floor 1, its total distance traveled will be 9 floors.

Mode B

Mode B allows for operating the elevator more efficiently. Similar to Mode A, the elevator in Mode B will acknowledge each command in the order it is received. However Mode B supports transporting more people at once and if any consecutive requests to travel in the same direction are received, the elevator can handle them all in one trip. If in the example above Doug and Alan had requested the elevator in Mode B, the elevator would have picked up Doug on 4, picked up Alan on 3, dropped off Alan on 2, and finally dropped off Doug on 1.



In other words, the commands 4-1, 3-2 can be merged because they are consecutive requests to travel from a higher floor to a lower floor. **Test Input**

10:8-1
9:1-5,1-6,1-5
2:4-1,4-2,6-8
3:7-9,3-7,5-8,7-11,11-1
7:11-6,10-5,6-8,7-4,12-7,8-9
6:1-8,6-8

Each line of your text file should consist of the initial floor location of the elevator, a colon ":", and one or more commands that are delimited with a comma ",". Each line represents a set of commands that should be processed exclusively and sequentially and the result of processing one line should not influence the result of the next.

Expected Output

Mode A	Mode B
10 8 1 (9) 9 1 5 1 6 1 5 (30) 2 4 1 4 2 6 8 (16) 3 7 9 3 7 5 8 7 11 1 (36) 7 11 6 10 5 6 8 7 4 12 7 8 9 (40) 6 1 8 6 8 (16)	10 8 1 (9) 9 1 5 6 (13) 2 4 2 1 6 8 (12) 3 5 7 8 9 11 1 (18) 7 11 10 6 5 6 8 12 7 4 8 9 (30) 6 1 6 8 (12)

For each set of commands the system should print to standard output a single line consisting of a space-delimited list of floors followed by the total distance in floors that the elevator travels, in parenthesis "(" and ")". The list of floors should begin with the initial floor location followed by the visited floors in the order that the elevator visits them.