

1. ArrayList in Java

A bus has a seating capacity of n . The fare depends on the capacity n and the number of passengers on the bus.

For example, if there is less than or equal to 25% of passengers currently on the bus, then they will be charged a traveling fare of $n+n*0.6$. If the passenger count is greater than 25% and less than or equal to 50% of the capacity of the bus, then they will be charged a fare of $n+n*0.3$. If the number of passengers on the bus is more than 50% of the seating capacity of the bus, then they will be charged a fare of n (If the value on dividing n by 2 or 4 gives a remainder other than 0, then **convert it into its ceiling value**).

You will be given the seating capacity, the number of stops during the trip, an arraylist of strings representing the passenger's ID prefixed with a "+" or a "-" sign denoting whether the passenger has got on or out of the bus, respectively, and a query q .

You have to write a program to get the output depending upon the query type. The following are the five type of queries:

- Query type 1 (1): You have to determine the number of passengers who have got on and out of the bus during that trip.
 - Example output: "(number of passengers who have got on the bus) passengers got on the bus and (number of passengers who have got out of the bus) passengers got out of the bus"

- Query type 2 (2): You have to determine the number of passengers who have traveled with a fare of $n+n*0.6$, $n+n*0.3$, and n .
 - Example output: "(number of passengers who have spent $n+n*0.6$ on their fare) passengers traveled with a fare of $(n+n*0.6)$, (number of passengers who have spent $n+n*0.3$ on their fare) passengers traveled with a fare of $(n+n*0.3)$ and (number of passengers who have spent n as their fare) passengers traveled with a fare of (n) "
- Query type 3 (3, Passenger id): You have to determine how much a particular passenger with the given passenger id has spent during the trip (Note: The same passenger can get on the bus more than once).
 - Example output: "Passenger (Passenger id) spent a total fare of (total fare spent by the passenger during the trip)"
- Query type 4 (4, Passenger id): You have to determine how many times a particular passenger with the given passenger id has got on the bus during the trip.
 - Example output: "Passenger (Passenger id) has got on the bus for (number of times the passenger has got on the bus) times"
- Query type 5 (5, Passenger id): You have to determine whether the given passenger is on the bus or not at the end of the trip
 - Example output: "Passenger (Passenger id) was (Inside/not inside depending upon whether the passenger is on the bus or not at the end of the trip) the bus at the end of the trip"

Note: In the example outputs, the values enclosed within the parentheses should be replaced with the respective calculated values.

Input Specifications:

- The first line of the input consists of an integer n describing the seating capacity of the bus.
- The second line of the input consists of an integer x describing the number of stops during the trip.
- The next x lines of input consist of the strings with the list of passengers who have got on and out of the bus.
- The last line of the input is an integer q describing the query (query types 3, 4, and 5 have a comma and a space separating the values).

Output Specifications:

You should print the output depending upon the query given.

Sample Input 1:

```
10
4
+2501 +2502 +2503 +2504
-2501 -2504 +2505 +2506 +2507 +2509
+2501 +2511 -2502 -2505
+2513 -2507 -2503 -2511 -2509
2
```

Sample Output 1:

1 passengers traveled with a fare of 16.0, 4 passengers traveled with a fare of 13.0 and 6 passengers traveled with a fare of 10.0

Explanation:

Stop no.	No. Of passengers got in the bus	No. Of passengers got out of the bus	Total number of passengers inside the bus	Passengers paying $(n + n \cdot 0.6)$ as fare	Passengers paying $(n + n \cdot 0.3)$ as fare	Passengers paying n as fare
1	4(2501, 2502, 2503, 2504)	0	4	-	4(2501, 2502, 2503, 2504)	-
2	4(2505, 2506, 2507, 2509)	2(2501, 2504)	$4+4-2 = 6$	-	-	4
3	2(2501, 2511)	2(2502, 2505)	$6+2-2 = 6$	-	-	2
4	1(2513)	4(2507, 2503, 2511, 2509)	$6+1-4 = 3$	1	-	-
Total				1	4	6

Therefore, in total, 1 passenger spent a fare of 16, 4 spent a fare of 13 and 6 passengers paid a fare of 10.

Sample Input 2:

15
6
+100 +101 +102 +103 +104 +105
+106 +107 +108 -101 -104 -105 -103 -102
+101 +109 +110 +111 -106 -100
+112 +113 -101 +114 -109 -107 -110 -111
+101 -112 -113
+115 +116 -101 +112
3, 101

Sample Output 2:

Passenger 101 spent a total fare of 63.0

Explanation:

Step no.	No. of passengers got in the bus	No. of passengers got out of the bus	Total no. of passengers inside the bus	Passengers paying $(n + n/4)$ as fare	Passengers paying $(n + n/2)$ as fare	Passengers paying n as fare
1	0,000, 000, 000, 000, 000	0	0	-	0	-
2	0,000, 000, 000	0,000, 000, 000, 000, 000	0, 0, 0, 0, 0	0	-	-
3	0,000, 000, 000, 000	0,000, 000	0, 0, 0, 0	0	0	-
4	0,000, 000, 000, 000	0,000, 000, 000, 000	0, 0, 0, 0	0	0	-
5	0,000	0,000, 000	0, 0, 0, 0	0	0	-
6	0,000, 000, 000	0,000	0, 0, 0, 0	0	0	-
Total				0	0	0

The passenger with id 101 has got on the bus thrice(highlighted in the table). The number of passengers on the bus when the passenger has got on the bus for the first and the second time is greater than 25% of the capacity but less than half of the capacity. Hence, the passenger has spent 19.5 each time. The last time when passenger 101 has got on the bus, there is less than 25% of the capacity of passengers on the bus, so the passenger has spent 24.0. In total, the passenger has spent 63.0 (19.5+19.5+24.0).

Sample Input 3:

```
15
6
+100 +101 +102 +103 +104 +105
+106 +107 +108 -101 -104 -105 -103 -102
+101 +109 +110 +111 -106 -100
+112 +113 -101 +114 -109 -107 -110 -111
+101 -112 -113
+115 +116 -101 +112
5, 101
```

Sample Output 3:

Passenger 101 was not inside the bus at the end of the trip

Explanation:

Passenger 101 was not inside the bus at the sixth stop as the last symbol associated with the id is -.

Note: You should use an **ArrayList** of the given class type **Passenger** to store the user Id and the fare.

Stop no.	No. of passengers got in the bus	No. of passengers got out of the bus	Total no. of passengers inside the bus	Passengers paying (n + n*0.6) as fare	Passengers paying (n + n*0.3) as fare	Passengers paying n as fare
1	6(100, 101, 102, 103, 104, 105)	0	6	-	6	-
2	3(106, 107, 108)	5(101, 102, 103, 104, 105)	$6+3-5 = 4$	3	-	-
3	4(101, 109, 110, 111)	2(100, 106)	$4+4-2 = 6$	-	4	-
4	3(112, 113, 114)	5(101, 107, 109, 110, 111)	$6+3-5 = 4$	3	-	-
5	1(101)	2(112, 113)	$4+1-2 = 3$	1	-	-
6	3(112, 115, 116)	1(101)	$3+3-1 = 5$	-	3	-
Total				7	13	0