Given a set of keys, create a circular linked list.

A circular linked list is same as the singly linked list. However, in addition the next pointer of the tail node points to the head node.

You should support following functions on the circular linked list

1. Create a new empty circular linked list

2. Add a new value at the beginning of the list. The newly inserted value will become the new head of the list.

3. Add new value at the end of the list. The newly inserted value will become the new tail of the list.

4. Given a value to search, find out where the value is present in the linked list. Find its shortest distance from the head node. Note that there are two possible paths. You have to print the length of the shortest path. Assume that the list does not contain the duplicates. If the value is not present in the list then your answer should be -1.

How to interpret the input file

Line 1: Value 1 indicates that create a new empty circular linked list.

Lines 2 and 3: Value 2 on line 2 indicates insert at the beginning of the list. The value to be inserted is 10.

Lines 4 to 11: Insert values 20, 30, 40, and 50 at the beginning of the list.

Lines 12 and 13: Value 3 indicates insert at the end of the list. The value to be inserted is 60.

Lines 14 and 15: Insert 70 at the end of the list.

Lines 16 and 17: Insert 80 at the beginning of the list

At the ned of all these insertions your list should look like as follows

Head 80, 50, 40, 30, 20, 10, 60, 70 Tail and tail points back to head.

Lines 18 and 19: Value 4 indicates that we have to perform the search operation. The value to be searched is 71. It is not present in the list. Output should be -1.

Lines 20 and 21: Search for 80. It is present in the list. The head node is 80. It's distance from itself is 0. Output should be 0.

Lines 22 and 23: Search for 50. It is present in the list. There are two paths between 50 and head node 80. First, starting from 80 and ending at 50. Its length is 1. Second, starting from 50 and ending at 80. Its length is 7. Output should be 1 as we are trying to find out the shorter path.

Lines 24 and 25: Search for 10. It is present in the list. There are two paths between 10 and head node 80. First, starting from 80 and ending at 10. Its length is 5. Second, starting from 10 and ending at 80. Its length is 3. Output should be 3 as we are trying to find out the shorter path.