Extra Explanation on first project:

1) In this project you must reach a goal state from an initial state. For example:

**Input to your program:**
Initial state: 2 linear chains with length 3, 4 and a circular chain of length 8
Goal state: circular chain of length 15

**Output of your program:**
The way is (one possible way):
1) unjoin circular chain of length 8 and make a linear chain
2) join chains of length 3, 4
3) join chains of length 7, 8
4) circulate linear chain of length 15

You can see here that I have chosen to have these operators:
join, unjoin, circulate, … (upon your own idea)

Please note that in this problem you can use an arraylist, to extend each state, and use it as a queue. In each step you can use one of the operators, and check that is that the reached state equal to goal state or not.

2) Implementing a divide and conquer algorithm is not mandatory. You may use another algorithm and it is still ok, remember that there might be no divide and conquer algorithm for the problem and you should simply explain it in your documentation, based on your reasonings.

**HINT:**
3) The answer to this problem is not unique, and different algorithms can find different answers, but a simple search (bfs one) can find exactly the optimum answer. Implementing an algorithm with the optimum answer will surely get extra point.

4) Do not waste your time on the GUI. You may use any in/output preferred type.

5) Feel free to contact me and ask your questions by: k_ariafar@ce.sharif.edu

Regards
Aryafar