

Thanos

Thanos is a fictional supervillain character that has a long list of enemies to eliminate. In order to help streamline and manage this long list, he has hired Rar the Cat to code him a **queue**. Thanos has a special ability that can eliminate half of the living things in the universe at the snap of his fingers. Hence, he has a special request for the queue to support removing the **first half** of the queue from the front. Rar the Cat knows that he might also disappear if he fails to implement this queue, so he has sought your help to do so. Can you help him?

The queue is **initially empty** and will need to support a few operations:

Operation	Description
ADD [name]	Adds a person with name [name] to the back of the queue.
POP	Removes the person at the front of the queue and output the name of the removed person.
SNAP	<p>If there are N persons in the queue, the first N/2 persons (rounded up) at the front of the queue should be removed one by one starting from the front of the queue. Output the names of people removed by this operation, in the order of their removal.</p> <p>For example, if there are N = 5 persons in the queue, the first 3 persons from the front of the queue should be removed. If there are N = 2 persons in the queue, the person at the front of the queue should be removed.</p>

Input

The first line of input will be a single integer **Q**, the number of operations to be performed on the queue.

Q lines will follow, representing an operation each. The operations should be executed in order and the format would be as described in the table above. (See sample)

Output

For every **POP** operation, output the name of the removed person on a single line.

For every **SNAP** operation, output the list of people removed by the operation on a single line from the first person removed to the last person removed. Add a single space between two consecutive names. **Do not print a space after the last name.**

Limits

- $1 \leq Q \leq 500,000$
- The names of persons will only contain lowercase and uppercase English alphabets. It will also not be longer than 20 characters.
- It is guaranteed that there will always be at least one person in the queue when any **POP** or **SNAP** operation is given.

Sample Input (thanos1.in)	Sample Output (thanos1.out)
10 ADD Rar ADD Panda ADD Sharkie ADD BellCurveGod ADD PoorStudent POP ADD SoCCat SNAP POP SNAP	Rar Panda Sharkie BellCurveGod PoorStudent SoCCat

Explanation of Sample Testcase 1

Operation	Queue (Front to Back)
ADD Rar	Rar
ADD Panda	Rar, Panda
ADD Sharkie	Rar, Panda, Sharkie
ADD BellCurveGod	Rar, Panda, Sharkie, BellCurveGod
ADD PoorStudent	Rar, Panda, Sharkie, BellCurveGod, PoorStudent
POP	Rar , Panda, Sharkie, BellCurveGod, PoorStudent
ADD SoCCat	Panda, Sharkie, BellCurveGod, PoorStudent, SoCCat
SNAP	Panda, Sharkie, BellCurveGod , PoorStudent, SoCCat
POP	PoorStudent , SoCCat
SNAP	SoCCat

Sample Input (thanos2.in)	Sample Output (thanos2.out)
12 ADD Alice ADD Bob ADD Charlie ADD Darwin ADD Eve ADD Fred ADD Grey ADD Harry SNAP SNAP SNAP SNAP	Alice Bob Charlie Darwin Eve Fred Grey Harry

Notes:

1. You should develop your program in the subdirectory **ex3** and use the skeleton java file provided. You should not create a new file or rename the file provided.
2. You are free to define your own helper methods and classes (or remove existing ones).
3. Please be reminded that the marking scheme is:
 - a. Public Test Cases (1%) - 1% for passing **all** test cases, 0% otherwise
 - b. Hidden Test Cases (1%) - Partial scoring depending on test cases passed
 - c. Manual Grading (1%)
 - i. Overall Correctness (correctness of algorithm, severity of bugs)
 - ii. Coding Style (meaningful comments, modularity, proper indentation, meaningful method and variable names)
4. Your program will be tested with a time limit of not less than **2 sec** on Codecrunch.

Skeleton File – Thanos.java

You are given the below skeleton file `Thanos.java`. You should see a non-empty file when you open the skeleton file. Otherwise, you might be in the wrong working directory.

```
/**
 * Name      :
 * Matric. No :
 * PLab Acct. :
 */

import java.util.*;

public class Thanos {
    private void run() {
        //implement your "main" method here
    }

    public static void main(String[] args) {
        Thanos newThanos = new Thanos();
        newThanos.run();
    }
}
```