**Problems Creation or Modification**

**Problem 1:**

|  |  |
| --- | --- |
| Name | Yiwen and RPG |
| Description  (detailed)  #include image if necessary | This is a very simple problem. (I have actually created this problem for fun during the June holidays.) Anyways here’s the link:[https://vjudge.net/contest/379555#problem/A](https://vjudge.net/contest/379555" \l "problem/A)  Basically you are on a grid and the person will move in some directions. You are to find the number of times the person went to a place he already went to. |
| Solution | Just use a set. |
| Alternative Solution | Dict  List |
| Test Cases  With expected output | 1. [ “move down”, “move down”, “move up”, “move up”]   output is 2   1. [“move up”, “move up”]   output is 0   1. [“move down”, “move left”, “move right”, “move up”]   output is 2   1. Please generate more testcases |
| Concepts Needed / Prior Knowledge | List/Set/Map/Tuple |
| Reference | ***Self-Created*** / Modification  If Modification is chosen above. Please include the link for the reference below: |

**Problem 2:**

|  |  |
| --- | --- |
| Name | Swapping smurfs |
| Description  (detailed)  #include image if necessary | You have a permutation A of length N.(e.g. A=(0,1,2,3,4) or A=(4,1,3,2,0))  you have also N smurfs in 1 line, ordered 0 to N-1. Then, after each turn, each smurf will go move to the index pointed by A.  For example, If A = (4,1,3,2,0), then  Turn 1: (0,1,2,3,4)  Turn 2: (4,1,3,2,0)  Turn 3: (0,1,2,3,4)  Turn 4: (4,1,3,2,0)  You need to find where the xth smurf is at after yth turn.(y should be a huge number(like 10^ 20 or something), so they will be forced to use a clever method) |
| Solution | Basically, each smurf will be trapped in a “cycle”, until he reaches back to its original spot. So, first find the length of the cycle that the smurf is in. For example, in the example, smurf 1 is in the cycle [1,5] because he’ll reach home immediately after going to the 5th index. So, we can take y %= cycle\_length, and then we can just return the yth element inside the cycle.  (Code with sample tests: <https://ideone.com/GbXEEW>) |
| Alternative Solution | Just simulate until the end of the life as we know it. |
| Test Cases  With expected output | Test case format:  (see above link) [[permutation A],[x],[y]] |
| Concepts Needed / Prior Knowledge | Lists, mod, some creativity I guess. |
| Reference | Self-Created / **Modification**  I made the problem easier  If Modification is chosen above. Please include the link for the reference below:  [http://usaco.org/index.php?page=viewproblem2&cpid=1043](http://usaco.org/index.php?page=viewproblem2&cpid=1043(I) |

**Problem 3:**

|  |  |
| --- | --- |
| Name | -- --- .-. ... . / -.-. --- -.. . |
| Description  (detailed)  #include image if necessary | Morse code is amazing! Ch's amburr have been using morse code to communicate with his master. However, amburr has a small memory, so it cannot store a lot of things. Also, it can only write 2 letters, '.' and '-'. It can't even write space. Thus, ch has been getting a lot of trouble decoding the messages from amburr. Help him by telling him all possible combinations of english messages that could have been sent. |
| Solution | Recursion. See code:  <https://ideone.com/4ejF5t> |
| Alternative Solution | No idea. |
| Test Cases  With expected output | See the ideone link above  morse(".-.") returns ['R', 'EN', 'ETE', 'AE', 'ETE']  morse(".--..-") returns ['ETX', 'ETTU', 'ETTEA', 'ETTEET', 'ETTIT', 'ETTEET', 'ETNA', 'ETNET', 'ETTEA', 'ETTEET', 'ETDT', 'ETTIT', 'ETTEET', 'ETNET', 'ETTEET', 'ENU', 'ENEA', 'ENEET', 'ENIT', 'ENEET', 'ETEU', 'ETEEA', 'ETEEET', 'ETEIT', 'ETEEET', 'EGA', 'EGET', 'ETNA', 'ETNET', 'ETTEA', 'ETTEET', 'ENEA', 'ENEET', 'ETEEA', 'ETEEET', 'EZT', 'ETDT', 'ETTIT', 'ETTEET', 'ETNET', 'ETTEET', 'ENIT', 'ENEET', 'ETEIT', 'ETEEET', 'EGET', 'ETNET', 'ETTEET', 'ENEET', 'ETEEET', 'AX', 'ATU', 'ATEA', 'ATEET', 'ATIT', 'ATEET', 'ANA', 'ANET', 'ATEA', 'ATEET', 'ADT', 'ATIT', 'ATEET', 'ANET', 'ATEET', 'ETX', 'ETTU', 'ETTEA', 'ETTEET', 'ETTIT', 'ETTEET', 'ETNA', 'ETNET', 'ETTEA', 'ETTEET', 'ETDT', 'ETTIT', 'ETTEET', 'ETNET', 'ETTEET', 'RU', 'REA', 'REET', 'RIT', 'REET', 'ENU', 'ENEA', 'ENEET', 'ENIT', 'ENEET', 'ETEU', 'ETEEA', 'ETEEET', 'ETEIT', 'ETEEET', 'AEU', 'AEEA', 'AEEET', 'AEIT', 'AEEET', 'ETEU', 'ETEEA', 'ETEEET', 'ETEIT', 'ETEEET', 'PA', 'PET', 'EGA', 'EGET', 'ETNA', 'ETNET', 'ETTEA', 'ETTEET', 'ENEA', 'ENEET', 'ETEEA', 'ETEEET', 'ANA', 'ANET', 'ATEA', 'ATEET', 'ETNA', 'ETNET', 'ETTEA', 'ETTEET', 'REA', 'REET', 'ENEA', 'ENEET', 'ETEEA', 'ETEEET', 'AEEA', 'AEEET', 'ETEEA', 'ETEEET', 'EZT', 'ETDT', 'ETTIT', 'ETTEET', 'ETNET', 'ETTEET', 'ENIT', 'ENEET', 'ETEIT', 'ETEEET', 'EGET', 'ETNET', 'ETTEET', 'ENEET', 'ETEEET', 'ADT', 'ATIT', 'ATEET', 'ANET', 'ATEET', 'ETDT', 'ETTIT', 'ETTEET', 'ETNET', 'ETTEET', 'RIT', 'REET', 'ENIT', 'ENEET', 'ETEIT', 'ETEEET', 'AEIT', 'AEEET', 'ETEIT', 'ETEEET', 'PET', 'EGET', 'ETNET', 'ETTEET', 'ENEET', 'ETEEET', 'ANET', 'ATEET', 'ETNET', 'ETTEET', 'REET', 'ENEET', 'ETEEET', 'AEEET', 'ETEEET']  morse("...") returns ['R', 'EN', 'ETE', 'AE', 'ETE'] |
| Concepts Needed / Prior Knowledge | Recursion, dict, for loops |
| Reference | **Self-Created** / Modification  Came up with this idea while chatting |