Happy Ladybugs

by nabila_ahmed

Problem Submissions Leaderboard Discussions

Happy Ladybugs is a board game having the following properties:

- The board is represented by a string, b, of length n. The i^{th} character of the string, b_i, denotes the i^{th} cell of the board.
  - If b_i is an underscore (i.e., `_`), it means the i^{th} cell of the board is empty.
  - If b_i is an uppercase English alphabetic letter (i.e., `A` through `Z`), it means the i^{th} cell contains a ladybug of color b_i.
  - String b will not contain any other characters.
- A ladybug is *happy* only when its left or right adjacent cell (i.e., $b_{i\pm1}$) is occupied by another ladybug having the same color.
- In a single move, you can move a ladybug from its current position to any empty cell.

Given the values of n and b for g games of Happy Ladybugs, determine if it's possible to make all the ladybugs happy. For each game, print `YES` on a new line if all the ladybugs can be made happy through some number of moves; otherwise, print `NO` to indicate that no number of moves will result in all the ladybugs being happy.

Input Format

The first line contains an integer, g, denoting the number of games. The $2 \cdot g$ subsequent lines describes a Happy Ladybugs game in the following format:

1. The first line contains an integer, n, denoting the number of cells on the board.
2. The second line contains a string, b, describing the n cells of the board.

Constraints

- $1 \leq g \leq 100$
- $1 \leq n \leq 100$
- It is guaranteed that string b consists of underscores and/or uppercase English alphabetic letters (i.e., `_` and `A` through `Z`).

Output Format

For each game, print `YES` on a new line if it is possible to make all the ladybugs *happy*; otherwise, print `NO`.

Sample Input

```
4
7
RBY_YBR
6
X_Y__X
2
__
6
B_RRBR
```

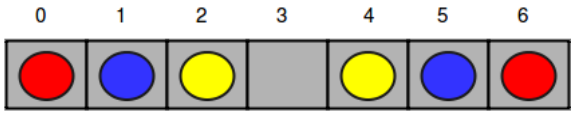
Sample Output

```
YES
NO
YES
YES
```

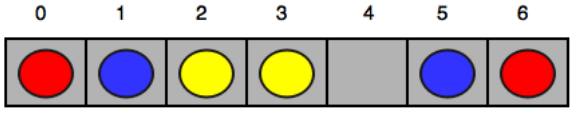
Explanation

The first three games of Happy Ladybugs are explained below:

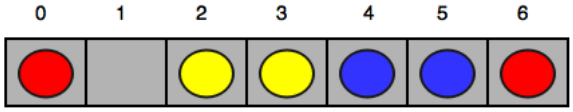
1. Initial board:



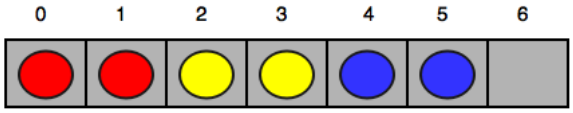
After the first move:



After the second move:



After the third move:



Now all the ladybugs are happy, so we print `YES` on a new line.

2. There is no way to make the ladybug having color `Y` happy, so we print `NO` on a new line.
3. There are no unhappy ladybugs, so we print `YES` on a new line.


```
#!/bin/python

import sys
from collections import Counter


Q = int(raw_input().strip())
for a0 in xrange(Q):
    n = int(raw_input().strip())
    b = raw_input().strip()

    dic = Counter(b)
    if (len(dic.keys()) > (n/2)) or (dic['_'] <1):
```

```
        print "NO"
    else:
        keys = dic.keys()
        flag = True
        if '_' in keys:
            keys.remove('_')
            for color in keys:
                if dic[color] < 2:
                    flag = False
        else:
            colors = iter(n)
            prev_color = next(colors)
            for cur_color in colors:
                if prev_color == cur_color:
                    if flag:
                        flag = True
                    else:
                        flag = False
        if flag:
            print "YES"
        else:
            print "NO"
```

Line: 23 Col: 14

Upload Code as File Test against custom input Run Code Submit Code

Testcase 0 ✖ Testcase 1 ✖

Nice try, but you did not pass this test case.

Input (stdin)

```
4
7
RBY_YBR
6
X_Y__X
2
__
6
B_RRBR
```

Your Output (stdout)

```
NO
NO
YES
YES
```

Expected Output

```
YES
NO
YES
YES
```

Compiler Message

```
Wrong Answer
```