

CS1020 Lab #5

Exercise #2: Candy Crush

http://www.comp.nus.edu.sg/~cs1020/3_ca/labs.html

Learning Objectives:

- (1) Working with two-dimensional array.
- (2) Writing recursive function.

Task Statement:

In the popular game Candy Crush, swapping two candies causes candies to be crushed if they are connected in a row of three or more (see examples below). Once the candies are crushed, they disappear from the board and the candies in the rows above drop down. In this exercise, you are to implement a new level called the bonus 1020 level. In this level, no candies will be dropped from the rows above after the connected candies are crushed. Instead, the level is cleared if you can crush a target number of candies within a specified number of moves.

You are to write a program **CandyCrush.java** that will compute whether the target number of destroyed candies is met given the board of candies, the number of moves as well as the moves that are made.

Input

The first line gives you the number of rows and columns, N and M, of the candies board. The next N lines contain M characters each, with each character representing a candy.

The next input is T, the target number of candies that needs to be crushed in order to clear the level. This is followed by K, the number of moves which will be made.

The next K lines consist of four integers each, identifying the two candies that will be swapped. For example:

- **2 3 2 4** means swapping the candy in row 2 column 3 with the candy in row 2 column 4.
- **4 5 3 5** means swapping the candy in row 4 column 5 with the candy in row 3 column 5.

Output

The program outputs the final board after the moves have been made, followed by a line indicating whether the level has been cleared.

Sample Input 1:

3 3 ← Number of rows and number of columns of the candies board
ryb
bry
rgb
6 ← Target number of candies to destroy
2 ← Number of moves to be made
1 0 1 1 ← Swap candy at row 1 column 0 (b) with the candy at row 1 column 1 (r)
1 1 1 2 ← Swap candy at row 1 column 1 (now it is b) with the candy at row 1 column 2 (y)

Sample Output 1:

-y-
-y-
-g- ← '-' represents a space that is unoccupied by any candy
Level 1020 Cleared!

Sample Input 2:

5 4
haha
hehe
eh eh
hahe
hihi
6
1
2 0 2 1

Sample Output 2:

-aha
-ehe
-eeh
-ahe
-ihi
Level 1020 Not Cleared!

Sample Input 3:

```
5 6
abcdef
abcdef
eefegg
abcdef
abcdef
15
1
2 2 2 3
```

Sample Output 3:

```
---def
---def
---fgg
---def
---def
Level 1020 Cleared!
```

Note: Part of a suggested solution has been provided in the skeleton program. However, you may edit it as you wish or come up with your own code.