Problem 266: Just One Thing

Difficulty: Hard

Author: Javier Jimenez, Marietta, Georgia, United States

Originally Published: Code Quest 2025

Problem Background

Lockheed Martin has been contracted by the Transportation Security Administration to develop some new airport security equipment, to be trialed at Atlanta International Airport. They'd like to implement a process in which passengers will place only a single item in each bin to be passed through the x-ray scanner; this will help TSA agents better identify what's going through the scanner. To help enforce this, they're planning to set up a camera that will watch as bins approach the x-ray scanner. If a bin contains more than one item, it will flash a warning light so agents know to rearrange things.

Problem Description

You'll be provided with an image of a security bin in X PixMap, or XPM, format. XPM files are text-based image files that use ASCII characters to represent individual pixels in an image. Your task is to identify if there are multiple objects within the bin. Since the bin itself is a uniform gray color (specifically #C0C0C0), you should be able to identify objects by finding areas of the image that are *not* that color. Objects may be any shape and size, and may contain any combination of colors other than #C0C0C0.

XPM files are easily viewed within a text editor, and follow a consistent format:

- The first line contains a comment identifying the file as an XPM file (this is irrelevant to this problem):
  `/* XPM */`
- The second line declares a string array in the C programming language (also irrelevant for our purposes):
  `static char * file_name_here_xpm[] = {`
- The third line is the first one to contain any useful information; it contains a string with four integers, separated by spaces:
  - The width of the image, in pixels
  - The height of the image, in pixels
  - The number of colors defined in the image
  - The number of characters used to define each pixel (for this problem, this value will always be 1)
- The next several lines – the exact number determined by the third value in the previous string - will contain strings that define the colors used in the image. Each color is defined as follows:

    - An ASCII character (including spaces!) used to represent a pixel in the image
    - A tab character
    - A lowercase letter indicating the type of the color; for this problem, this will always be 'c', for "color"
    - A space
    - A hexadecimal RGB color string, such as #C0C0C0 for the bin's background.

The remaining lines contain strings representing the image itself. The number of strings will be equal to the image's height, and the length of each string equal to its width. Each character represents a single pixel; the color of that pixel is determined using the color mapping provided above.

Sample Input

The first line of your program's input, **received from the standard input channel**, will contain a positive integer representing the number of test cases. Each test case will include a single image in XPM format; while this format is generally outlined above, to review:

- The first two lines of each test case contain no relevant data and may be ignored
- The third line of each test case will contain a string (wrapped with double quotes, and followed by a comma) containing the following positive integer values separated by spaces:
    - W, the width of the image in pixels
    - H, the height of the image in pixels
    - C, the number of colors defined for potential use in the image
    - P, the number of characters representing each pixel. This value will always be 1.
- C lines containing color definitions. Not all colors defined here will necessarily be used within an image. These are also defined as strings (wrapped with double quotes, and followed by a comma), using the following format:
    - A single printable ASCII character, which is the character that will be used to represent pixels in this color. This character will not be a tab, double quote, or backslash, but may be any other ASCII character appearing in the table in the Reference Materials.
    - A tab (\t), a lowercase c, and a space
    - A hexadecimal color code, prefaced with a # character
- H lines, each containing a string of length W wrapped in double quotes and followed by a comma (except for the last line, which is followed by a closing curly bracket and semicolon). These strings represent the pixels in the image, and will contain characters listed in the color definition section above.

Due to the length of the sample input, it is not replicated here; please download the sample input from the contest website's "Problemset" tab.

Sample Output

For each test case, your program must print a single line containing the word “OK” if the image contains at most one object; or containing the word “WARNING” if the image depicts more than one object.

```
OK
WARNING
```