UCF "Practice" Local Contest — Aug 25, 2012

Pete's Pantry

filename: `pantry`

Pete isn't feeling too well today. In an effort to try and get better, he wants to heat up a can of chicken soup. However, his pantry is a mess, and as soon as he grabs the can he wants, the stacks of cans all collapse and roll onto the kitchen floor. "If I only had a way to stack these cans to keep this from happening," Pete thought.

The Problem:

Your job is to write a program to help Pete stack cans in his pantry. Cans come in all shapes and sizes, but for simplicity, we'll assume that all cans are the same height, and only their width (diameter) will vary. The stacking must follow one simple rule:

No can shall be stacked atop a narrower can, or a can of equal width.

Pete doesn't follow written directions very well, so he wants your program to show him how to stack his cans. For example, a can with the label "BEANS" should look like this:

```
#######
#     #
#     #
#BEANS#
#     #
#     #
#     #
#######
```

The width of each can is determined by the can's label. The label can have multiple lines, so the width of the can is simply the width of the longest line of the can's label, plus two (for # delimeters) . All cans are 8 units tall, counting the edges, so no can label will have more than six lines. Pete also wants a particular order to his cans, so your program should process the input cans in the order given. Try fitting a given can in the leftmost stack first, and work to the right if needed. If there is no existing stack that will hold a given can, start a new stack just to the right of the last existing stack (the first can should be placed in the leftmost spot on the shelf).

The Input:

Pete will have multiple sets of cans for you to stack. The first line of input will contain a positive integer, n, indicating the number of sets of cans (i.e., the number of data sets to be processed). Each set of cans will begin with a positive integer c ($1 \leq c \leq 100$), on a line by itself. On each of the next c lines will be a can label. Can labels consist of upper- and lower-case letters, digits, and spaces. The pound sign ('#') may also appear in a can label and designates the start of a new line (i.e., the can label must be put on multiple lines in output). There will be no leading or

trailing spaces on any line of the can label (in input) and the input lines will not exceed column 60. Assume that there is at least one letter (or digit) before and after each '#'.

The Output:

For each set of cans print "`Can Stack #`*d*:", where *d* is the number of the set (starting with 1). Then, print two header lines to show column numbers. Then, print the stacks of cans using the format described above. Can labels must be centered both vertically and horizontally on the can (if an exact centering isn't possible, favor the left half and/or top half of the can). Each stacked can should be centered above the can below it (again, favor the left side if an exact centering is impossible). There should be exactly one space between the bottommost cans in each stack. Assume that no final arrangement will be more than 60 characters wide, and no stack will be taller than 10 cans.

Leave a blank line after the output for each data set. Folow the format illustrated in Sample Output.

Sample Input:

```
2
2
Coaches8#Ali88
Geeks of#UCFdorms#Soup
6
Pork and#Beans
Cream of#Mushroom#Soup
Baby#Carrots
Snow#Peas
Grape#Jelly
Aunt Helens#Down Home#Country#Goodness
```

(Sample Output on the next page)

12

Sample Output:

```
Can Stack #1:
          1         2         3         4         5         6
12345678901234567890123456789012345678901234567890123456789012345678901234567890
########## ##########
#         # #         #
#         # #Geeks of#
#Coaches8# #UCFdorms#
# Ali88   # #  Soup   #
#         # #         #
#         # #         #
########## ##########

Can Stack #2:
          1         2         3         4         5         6
12345678901234567890123456789012345678901234567890123456789012345678901234567890
 ######
 #    #
 #    #
 #Snow#
 #Peas#
 #    #
 #    #
 ######
##########   #######
#        #   #     #
#        #   #     #
# Baby   #   #Grape#
#Carrots#   #Jelly#
#        #   #     #
#        #   #     #
##########   #######
########## ########## ##############
#         # #         # #            #
#         # #Cream of# #Aunt Helens#
#Pork and# #Mushroom# # Down Home  #
# Beans   # #  Soup   # #  Country   #
#         # #         # # Goodness   #
#         # #         # #            #
########## ########## ##############
```