Homework 6 – Looping

Write a program using `for` loops to produce the following output. Please read carefully the entire specification! Your program must work for any odd number entered by the user, not just 13. (NOTE that there is no space after the colon on the Hill, Triangle, and Diamond labels.)

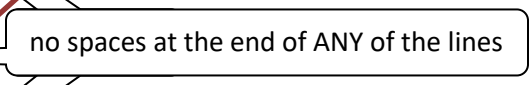
You may not use any if statements, no while loops, no special formatting code.

You must print 1 * at a time, i.e. you must use `cout << '*';` in your code.

You must use for loops inside for loops to solve the problem. See Program Development below for a sketch of the form of the solution.

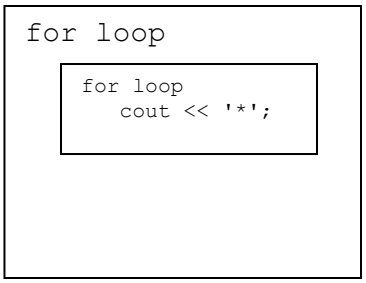
Sample Input & Output (Your output must look exactly like this!)

```
Enter an odd number width: 13

Hill:
*
***
*****
*******
*********
***********
*************

Triangle:
      *
     ***
    *****
   *******
  *********
 ***********
*************

Diamond:
      *
     ***
    *****
   *******
  *********
 ***********
*************
 ***********
  *********
   *******
    *****
     ***
      *
```



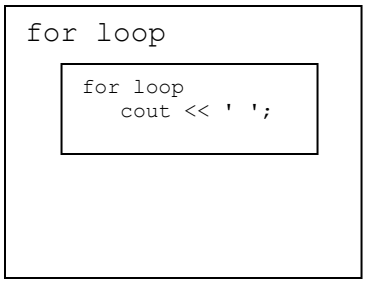
Program Development:

Start with the Hill. Each line must be printed with 1 `for` loop. Figure out how to print a line of stars, printing 1 star at a time. The *number of lines* is controlled by a `for` loop around that loop and prints each line.

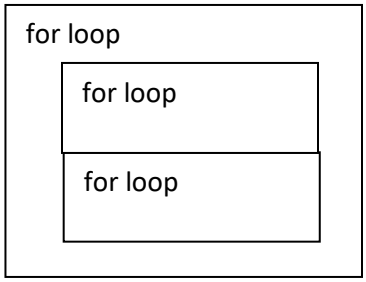


To print the Triangle, figure out how to print an upside down Hill, like this

```
----
---
--
-
```



Then, change the star - to a single space! **Each line** should be printed with **more than** 1 `for` loop, which is embedded in another `for` loop that prints multiple lines (i.e. the top 6 lines). This diagram might help:



Finally, to print the Diamond, print the Triangle for the top, and then use similar, sort of backward `for` loops to produce the bottom of the Diamond.

You may **not** use setw() or library functions to replicate a string multiple times; this assignment is about learning to use loops.

Submit your program to Web-CAT

You must match the output format exactly. Please note that there are not spaces before the stars of the Hill nor in front of the longest lines of the Triangle or Diamond.

Your output will be compared to string like **`"\nHill:\n*\n***\n*****\n\nTriangle:".`**

Here is another sample run:

```
Enter an odd number width: 5

Hill:
*
***
*****

Triangle:
  *
 ***
*****

Diamond:
  *
 ***
*****
 ***
  *
```