General Instructions

1. This activity consists of multiple short problems. Create one Python script file per problem.
2. Save each Python script according to the filename indicated in the problem statement.
3. **Each** Python script file should contain header information in comments, indicating your full name, your ID number, and the date you created your program.
4. **Each** Python script file should also contain **a comment block for the certification of authorship**, directly following the header information. See code_certification_template.py.
5. Create a folder named according to the convention: **HOA2-Surname-GivenName-IDNumber**. Place all of your Python files for this lab activity in this folder.
6. Archive your folder. On Windows, right-click on the folder and choose **Send To** > **Compressed (zipped) folder**. On a Mac, right-click on the folder and choose **Compress "folder name"**. Ensure that it is named **HOA2-Surname-GivenName-IDNumber.zip** and submit it through the appropriate Moodle submission module.

Note: For this lab, there is **no need** to include a separate Certificate of Authorship document.

IMPORTANT:

- Do not produce any excess output (e.g. *The answer is...*)
- Do not print cues for input (e.g. *Please input a number:*)
- The format of your output must match the output specifications exactly (see **Sample Output** column for examples for each problem)
- Unless explicitly stated otherwise, assume that the user will always follow the input restrictions (e.g. if input `n` is described as `0 < n < 100`, then the user will always input a value within that range), so there is no need for you to check for those.

Problem A : Do I Win?
Filename : hoa2a.py
Description : A program that determines if one score is higher than the other
Input : The program accepts two positive numbers, a and b, one on each line.
Note: The numbers may be floating point numbers.
Output : Print **`True`** if score a is higher than score b. Otherwise, print **`False`**.

Sample Input #1
```
5
4
```

Sample Output #1
```
True
```

Sample Input #2
```
99.9
99.999
```

Sample Output #2
```
False
```

Sample Input #3
```
1
1
```

Sample Output #3
```
False
```

Problem B : Letter Grades
Filename : hoa2b.py
Description : A program that determines the equivalent letter grade of a given numerical score
Input : The program accepts **any number** of integer scores ranging from `0` to `100`, with one score on each line. Input is terminated by an input score of `-1`.
Output : Print the corresponding letter grade of each numerical score, according to the grading system shown below.

A	92 - 100
B+	87 - 91
B	83 - 86
C+	79 - 82
C	75 - 78
D	70 - 74
F	< 70

NOTE: For this problem, you may produce the output immediately after entering each line of input (e.g. input `98`, then immediately output `A`). You are **not** allowed to use `lists` and `for` loops.

Sample Input
```
98
90
76
81
50
-1
```

Sample Output
```
A
B+
C
C+
F
```

Problem C : Dim Sum
Filename : hoa2c.py
Description : A program that adds all the numbers from the input
Input : The program accepts **any number** of integers, one on each line, ranging from `-100` to `100` inclusive. Input is terminated by any value that is outside the acceptable range.
Output : On a single line, print the sum of all the numbers from the input.

NOTE: For this problem, produce the output only after the input is terminated. You are **not** allowed to use `lists`, `for` loops, and the `sum` function.

Sample Input #1
```
4
5
6
11
1
2
4
6
-101
```

Sample Output #1
```
39
```

Sample Input #2
```
-3
5
2
-5
8
9
107
```

Sample Output #2
```
16
```

Problem D : Pyramid Scheme
Filename : hoa2d.py
Description : A program that outputs a triangle of numbers, given a height
Input : The input consists of an arbitrary number of lines.
Each line will contain a positive integer N, which denotes the height of one triangle.
Input is terminated by a value of `-1`.
Output : For each test case, output a triangle with a height of N.
Output exactly one blank line after every test case (i.e. a blank line after each triangle).

NOTE: For this problem, you may produce the output immediately after entering each line of input (e.g. input `3`, then immediately output the size 3 triangle). You must use a `while` loop to solve this problem. `lists`, `for` loops, and the `range()` function are **not** allowed. Use of the `end` parameter of the `print` function is **not** allowed. Use string concatenation to build the string instead.

Sample Input

```
4
3
1
5
-1
```

Sample Output

```
1
1 2
1 2 3
1 2 3 4

1
1 2
1 2 3

1

1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

Problem E : Diamonds in the Sky
Filename : hoa2e.py
Description : A program that prints diamonds composed of asterisks
Input : The input consists of an arbitrary number of lines.
Each line will contain a positive integer ***N***, which denotes the row number with the most number of asterisks.
Input is terminated by a value of `-1`.
Output : Print the diamonds representing each of the values of ***N*** in the input.
Output exactly one blank line after every test case (i.e. a blank line after each diamond).

NOTE: For this problem, you are **not** allowed to use string multiplication, the `range()` function, `lists`, and `for` loops. Use while loops, instead. Each output may be produced immediately after each input (e.g. input `3`, then immediately output the size 3 diamond). Use of the `end` parameter of the `print` function is **not** allowed. Use string concatenation to build the string instead.

Sample Input

```
1
2
3
5
-1
```

Sample Output

```
*

 *
***
 *

  *
 ***
*****
 ***
  *

    *
   ***
  *****
 *******
*********
 *******
  *****
   ***
    *
```

Reminders:

- Follow file naming conventions.
- Follow submission procedures.
- After submitting, double-check to see if you have successfully submitted the correct file.