Practice S06P06: Pyramid

http://www.comp.nus.edu.sg/~cs1010/4_misc/practice.html

Week of release: Week 6

Objectives: 2D array

Task statement:

Given a pyramid of integers, you can trace a path from the top to bottom, moving from a number to either the number on its left or right in the next row below. For example, Figure 1(a) shows a 4-row pyramid. Figure 1(b) shows a path with a sume of 13. Figure 1(c) shows another path with a sum of 18, which is the largest sum possible.

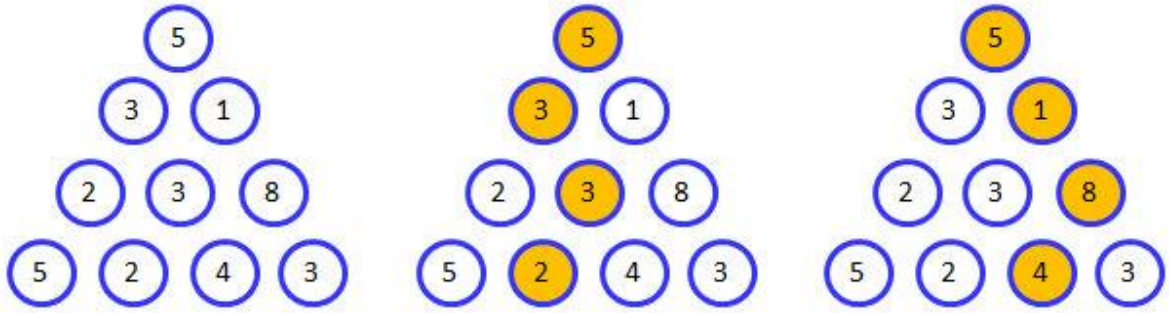

Figure 1. (a) A pyramid of integers. **(b)** A path with sum of 13. **(c)** A path with sum of 18.

Write a program **pyramid.c** to perform the following:

- Read a positive integer value for *size* This *size* indicates the number of rows in the pyramid. You may assume that *size* is at most 10.
- Create a 2-dimensional triangular array whereby the first row has 1 element, second row 2 elements, ..., last row *size* elements.

  Note that in C, a 2-dimensional aray has rows with the same number of elements. Hence, you have to leave some of these elements unused if the input data do not occupy all the elements.
- Read integers to fill in the array created above.
- Find the sum of the best path from the top-most row to the bottom-most row in the array that has the maximum sum.

Your program should have a function called **maxPathValue()** to compute the return the answer, and a function **scanTriangularArray()** to read data into the array and return the number of rows. A function **printTriangularArray()** to print the array is provided.

In the above task description, we describe the path to be from top to bottom. However, since all we want is the sum of the path, it is immaterial whether the path is from top to bottom, or from bottom to top. In fact, if we find the path from bottom to top, the code would be simpler, and more efficient. Try both approaches as a practice.

Sample run #1:

```
Enter number of rows: 4
Enter values for array:
5
3 1
2 3 8
5 2 4 3
Maximum path value = 18
```

Sample run #2:

```
Enter number of rows: 4
Enter values for array:
2
7 3
-1 4 9
3 0 6 2
Maximum path value = 20
```