Programming Refresher Workshop

Session 4 Exercises

Learning objective:

- Problem-solving on 2-dimensional arrays

Exercise 10 (ex10): Maximum Sum of Path in a Pyramid

Given a pyramid of integers, you can trace a path from the top to the 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 of integers. Figure 1(b) shows a path with a sum of 13. Figure 1(c) shows another path with a sum of 18, which is the largest sum possible.

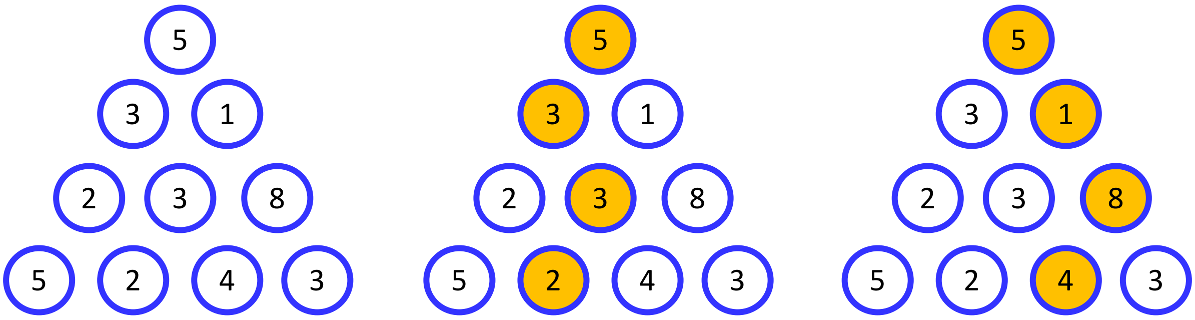

Figure 1. (a) A pyramid of integers. **(b)** A path with sum of 13. **(c)** A path with sum of 18.

Write a program 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 for certain language such as C, a 2-dimensional array 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. You may also assume that a pyramid has at most 10 rows.
- 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 contain a function/method called **maxPathValue**, with the array as its parameter, to compute and return the answer.

Sample runs

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

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