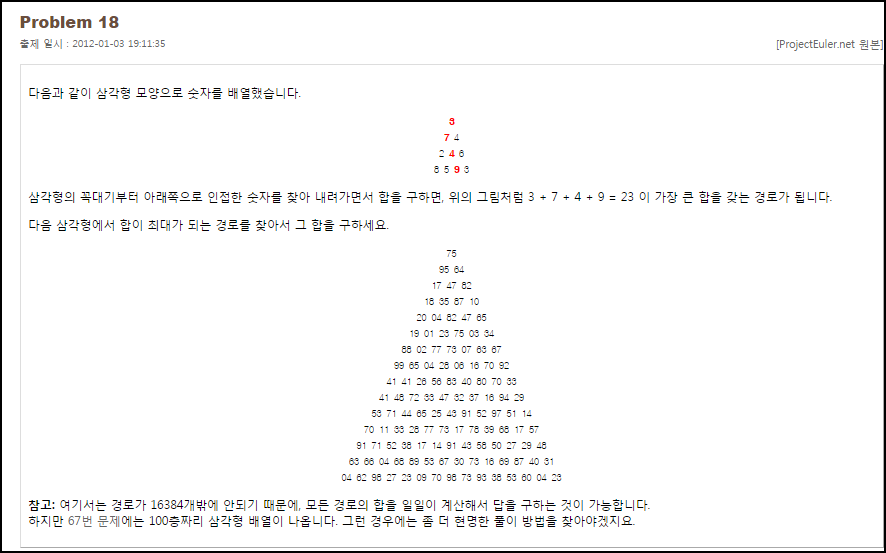
**Exercise**

삼각형 모양의 숫자 배열에서, 인접하는 숫자를 더해 내려가는 경우 합이 최대가 되는 경우를 구하는 문제입니다.



**Solution\_code**

해당 삼각형 배열은, 아래와 같이 2차원 리스트로 구축을 하였습니다. 처음엔 노드를 이용해, 트리 탐색에 대해서 알아보고 구현하려 했으나, 아래쪽에서 부터 거꾸로 올라오는 경우로 계산을 하면 간단히 해결할 수 있었기 때문에 해당 방법을 이용해 보았습니다.

가장 아래 이층 삼각형부터, 합이 최대가 되는 수를 선택하여, 합을 구해 올라오는 방법입니다.

|  |  |  |
| --- | --- | --- |
| **1**  **2**  **3**  **4**  **5**  **6**  **7**  **8**  **9**  **10**  **11**  **12**  **13**  **14** | **l = [[75],[95,64],[17,47,82],[18,35,87,10],[20,4,82,47,65],[19,1,23,75,3,34],[88,2,77,73,7,63,67],[99,65,4,28,6,16,70,92],[41,41,26,56,83,40,80,70,33],[41,48,72,33,47,32,37,16,94,29],[53,71,44,65,25,43,91,52,97,51,14],[70,11,33,28,77,73,17,78,39,68,17,57],[91,71,52,38,17,14,91,43,58,50,27,29,48],[63,66,04,68,89,53,67,30,73,16,69,87,40,31],[04,62,98,27,23,9,70,98,73,93,38,53,60,04,23]]**  **Sum = 0**  **result = []**  **L = len(l)**  **l = l[::-1]**  **cnt = 0**    **for i in range(L):**  **for j in range(len(l[i])-1):**  **l[i+1][j] += max(l[i][j],l[i][j+1])**  **# print j, l[i+1][j]**  **print l[i]**    **print l[i]**  ***[Colored by Color Scripter](http://colorscripter.com/info#e)*** | [cs](http://colorscripter.com/info#e) |

**Result**

