For this programming assignment, you have to write Python functions. Your function should return the value specified in the description of the problems. Do not print any messages or diagnostic information. Your code wll be evaluated automatically by comparing your program's output with the expected output, so any spurious output from your program will cause your answer to be reported as wrong.

There are some "public" test cases where you can see how your program does when you use "Compile and Run". Finally, you should "Submit" your code for evaluation. Your solution will be checked against "private" test cases, which you cannot see. You will get a score on 100 based on how many **private** test cases you solve correctly.

You can submit as many times as you like for programming assignments. The final submission will be counted for evaluation.

Write three Python functions as specified below. Paste the text for all three functions together into the submission window.

- You may define additional auxiiary functions as needed.
- In all cases you may assume that the value passed to the function is of the expected type, so your function does not have to check for malformed inputs.
- Ignore warnings about "Presentation errors".

1. Define a Python function "descending(l)" that returns True if each element in its input list is at most as big as the one before it

   For instance:

   ```
   >>> descending([])
   True

   >>> descending([4,4,3])
   True

   >>> descending([19,17,18,7])
   False
   ```

2. Define a Python function "alternating(l)" that returns True if the values in the input list alternately go up and down (in a strict manner).

   For instance:

   ```
   >>> alternating([])
   True

   >>> alternating([1,3,2,3,1,5])
   True

   >>> alternating([3,2,3,1,5])
   True

   >>> alternating([3,2,2,1,5])
   False

   >>> alternating([3,2,1,3,5])
   False
   ```

3. A two dimensional matrix can be represented in Python row-wise, as a list of lists: each inner list represents one row of the matrix. For instance, the matrix

   1 2 3
   4 5 6

   would be represented as [[1,2,3],[4,5,6]].

   Write a Python function "matmult(m1,m2)" that takes as input two matrices using this row-wise representation and returns the matrix product m1*m2 using the same representation.

   You may assume that the input matrices are well-formed and have compatible dimensions.

   For instance:

   ```
   >>> matmult([[1,2],[3,4]],[[1,0],[0,1]])
   [[1,2],[3,4]]

   >>> matmult([[1,2,3],[4,5,6]],[[1,4],[2,5],[3,6]])
   [[14, 32], [32, 77]]
   ```