Sortierte Punkteliste erweitern mit Erhalt der Ordnung

1. Schreiben Sie eine Funktion `insertScore(score, scoreList)`, die einer absteigend sortierten Liste `scoreList` derart eine weitere Zahl `score` hinzufügt, sodass die Liste weiterhin absteigend sortiert bleibt. Die Liste soll keine Duplikate enthalten. Die Funktion soll *keinen* Rückgabewert haben.

2. Schreiben Sie eine weitere Python-Funktion `testInsertScore()`, um Ihr Programm zu testen.

Solution:

```python
#!/usr/bin/env python
# coding: utf-8
# <h1>Table of Contents<span class="tocSkip"></span></h1>
# <div class="toc"><ul class="toc-item"><li><span><a href="#High-Scores" data-toc-modified-id="High-Scores-1"><span class="toc-item-num">1&nbsp;&nbsp;</span>High Scores</a></span></li></ul></div>
# ### High Scores
def insertScore(score, scoreList):
    """
    This function expects a descending list of scores, and inserts
    the a new score into it. Duplicates are ignored.

    :param score: float
    :param scoreList: list
    """
    n = len(scoreList)
    k = 0
    while (k < n):
        entry = scoreList[k]
        if (score > entry):
            break
        k += 1
    # If scoreList is empty entry will be undefined! However, python will not evaluate the second
    # statement of and if the first one is False. Therefore we just check if n == 0.
    if n and (score != entry):
        scoreList.insert(k, score)
def testInsertScore():
    scores = [0,1,2,3,4,5] # multiple examples for scores
    for sc in scores:
        scoreList = [4,3,1] # one example for scoreList
        print("\nScore", sc)
        print("Input scoreList", scoreList)
        insertScore(sc, scoreList)
        print("New scoreList", scoreList)

    insertScore(0, [])
testInsertScore()
```