1. Suppose that the list \( L \) equals \([100, ["hello", "bye"], [[1, 2], [2, 3], [3, 4]], 900]\).

Write down the values that the following expressions evaluate to.

(a) \( \text{len}(L) \)
(b) \( \text{len}(L[1][0]) \)
(c) \( L[2][2] \)
(d) \( \text{type}(\text{len}(L)-1) \)
(e) \( \text{type}(L[\text{len}(L)-2]) \)
(f) \( \text{type}(L[\text{len}(L)-1]) \)
(g) \( L[1][1][1] \)
(h) \( L[2][2][1] \)
(i) "bye" in \( L \)
(j) "bye" in \( L[1] \)
(k) 900 in \( L \)
(l) 100.0 in \( L \)
(m) \((3 \text{ in } L[2][0]) \text{ or } (3 \text{ in } L[2][1])\)

2. What is the output the following code fragment produces.

```python
L = [10, [1, 2, 3], "hello", [23]]
L = L + L[1]
print(L)
L = L + 2*[20, 10]
print(L)
L = 3*L[\text{len}(L)-1]
print(L)
```

3. Write a function called `equalLengthStrings` that takes as its single parameter a list of strings and returns `True` if all the strings in the given list have the same length. The function returns `False` otherwise.

4. Write a function called `addSubsequence` that takes a list \( L \) of numbers, and two indices \( i \) and \( j \) and returns the sum \( L[i] + L[i+1] + \ldots + L[j] \). You can assume that \( L \) is non-empty and \( 0 \leq i \leq j < \text{len}(L) \). For example, if \( L = [2, 4, 6, 3, 7, 2, 8, 9, 1] \) then the call to `addSubsequence(L, 2, 5)` should return 18.

5. Write a function called `deleteSubsequence` that takes a list \( L \) of numbers, and two indices \( i \) and \( j \) and returns the list with the sublist \( L[i], L[i+1], \ldots, L[j] \) removed. For example, if \( L = [2, 4, 6, 3, 7, 2, 8, 9, 1] \) then the call to `deleteSubsequence(L, 2, 5)` should return \([2, 4, 8, 9, 1]\).
6. Write a function called `maxPairSum` that takes a list of numbers as a parameter and returns the pair of numbers in consecutive positions that add up to the largest value. For example, if the given list is `[3, -1, 4, 2, 5, -1, 11, -8]` then the function would return the list `[-1, 11]`.

7. Write a function called `minIndex` that takes a list of numbers as a parameter and returns the index of a smallest element in the list. If there are several smallest numbers in the list, it does not matter which index is returned. For example, if the parameter is `[3, -1, 2, 3, -1, 11]` then the function could return 1 or it could return 4.