1. [6 marks] In this question you will write a function **`sum_squares`**, which consumes a list of numbers, and returns the sum of the squares of the *negative* numbers in the list, ignoring other values. For example, **`sum_squares( [-1, 456, -2.5, 2] )`** would return **`7.25`**.

a) **[3 marks]** Complete the body of the function **`sum_squares`** using *accumulative recursion*, without any loops or abstract list functions. You may add a helper function.

```
def sum_squares( lon ):
```

b) **[3 marks]** Complete the body of the function **`sum_squares`** using *abstract list functions* and *lambdas*, without any loops, recursion, or helper functions.

```
def sum_squares( lon ):
```

2. [4 marks] Consider a dictionary in which the keys are the names of cities, and the values are the day's high temperatures for those cities, for example:

```
d = { 'Kyiv' : 16.0, 'Amsterdam' : 13.0, 'Hangzhou' : 17.0,
      'Accra' : 32.2, 'Kuwait City' : 23.0  }
```

We would like to write a function **sort_temps** that consumes such a dictionary and returns a list of the city names it contains, sorted by increasing order of their temperatures. For example, **`sort_temps(d)`** would return **`['Amsterdam', 'Kyiv', 'Hangzhou', 'Kuwait City', 'Accra']`**.

a) **[1 marks]** Give a contract and requirements (if any) for the function **`sort_temps.`**

b) **[4 marks]** Complete the body of the function **`sort_temps.`** You may use any coding techniques you want from the course.

```
def sort_temps( cities ):
```

Multiple Choice: In the grid on the lower-right of this page, fill in the bubble corresponding to the correct answers to these questions.

1. True or false: in CS 116, 0 is considered a natural number.

   A. True   B. False

2. Which of the following algorithms runs in O(n log n) time in the worst case?

   A. Binary Search   B. Insertion Sort   C. Linear Search   D. Merge Sort   E. Selection Sort

3. What value results from evaluating the expression
   `'b'.join( 'banana'.split( 'n' )[::-2] )`?

   A. `'baba'`   B. `'abab'`   C. `'abba'`   D. `'baab'`   E. `'banana'`

4. Suppose that **bip** and **bop** are functions that consume a list and return another list of the same length. Both functions run in O(n) time, where n is the length of the consumed list. What is the worst-case running time of the following function?

   ```
   def fn1( L ):
       return bip( bop( L ) )
   ```

   A. O(1)   B. O(n)   C. O(n^2)   D. O(2^n)   E. Not enough information to determine

5. What is the value of **L** after the following code has executed?

   ```
   L = [1,2,3]
   L.append( L.append( 4 ) )
   ```

   A. **None**   B. **[1,2,3,None,4]**   C. **[1,2,3,4,None]**
   D. **[1,2,3,4,[1,2,3,4]]**   E. The code contains an error

6. Which answer is the most accurate worst-case running time of the following function, where n is the length of the list **L** and **m** is a positive natural number less than or equal to n?

   ```
   def fn2( L, m ):
       ret = 0
       for num in L[-m:]:
           ret *= num
       return ret
   ```

   A. O(1)   B. O(n)   C. O(m)   D. O(n+m)   E. O(n-m)

		Answers			
1	A○	B○	C○	D○	E○
2	A○	B○	C○	D○	E○
3	A○	B○	C○	D○	E○
4	A○	B○	C○	D○	E○
5	A○	B○	C○	D○	E○
6	A○	B○	C○	D○	E○