Assignment #20

Spring Examination Programs

1. An *encoding* of a string is a list of tuples, with each consisting of a character from the string and the number of consecutive occurrences of that character.

   (a) Write a Python function `Encode(s)` to return the encoding of string `s`. Thus:

   ```
   Encode("aaabaaccc")  ⇒  [("a",3),("b",1),("a",2),("c",3)]
   ```

   (b) Write a Python function `Decode(e)` to return the string with encoding `e`. Thus:

   ```
   Decode([("a",3),("b",1),("a",2),("c",3)])  ⇒  "aaabaaccc"
   ```

2. Suppose records of the top ten players of a game are stored in a file, where each line contains a score, a space, and a name, in descending order of score. For example:

   ```
   1358 Ann
    972 Tim
    775 Bob
   ... ...
   ```

   Each score is a positive integer below 10,000, and each name is a single word without spaces. The file will contain less than ten lines, if less than ten players have so far played the game.

   Write a Python function `Update(oldfile,newfile,score,name)` which takes a file `oldfile` of the above form, an integer `score` and a name `name`, representing the result of the latest game, and writes to the file `newfile` the updated records of top players.

   If `score` equals an existing top score, then it should appear *after* it, if at all, in `newfile`.

   If either file cannot be accessed, then issue an appropriate error message and return directly from the function.

3. Let `L` be a list of distinct integers, sorted in ascending order.

   A *fixed point* of `L` is any index `i` such that `L[i] = i`. For example, the list

   ```
             0   1  2  3  4  5  6
   L = [ -3, -1, 0, 2, 4, 5, 7 ]
   ```

   has two fixed points, 4 and 5.

   Now consider a Python function `FixedPoint(L)` which takes a list `L` of the above form, and returns either *some* fixed point of `L`, if one exists, or else `None`.

   (a) Write an *efficient iterative* version of `FixedPoint`.

   (b) Write an *efficient recursive* version of `FixedPoint`.

Program Submission:

Store the function definitions in a file named 'a20.py', and turn it in for grading by typing:

```
submit-cs1117 a20.py
```

Due Date: Fri Apr 8, 5:00pm