CS-1117 Introduction to Programming 2015-2016

Summer Examination 2015

Examination Number : ☐☐☐☐☐☐

INSTRUCTIONS

- Time : 3 hours.
- Answer all questions.
- *All final answers must be written on this Examination Paper*; use the backs of the pages if extra space is needed.
- The Answer Book is *only* for Rough Work.
- Complete the Identification Section of the Answer Book, and place this Examination Paper inside the Answer Book at the end of the examination.

1. Consider the following Python functions:

```
def F1( x, s ) :
    return F2( x, s, len( s ) - 1 )

def F2( x, s, i ) :
    if i < 0 :
        return None
    elif s[ i ] == x :
        return i
    else :
        return F2( x, s, i - 1 )
```

10% (a) Write a clear and concise *comment*, describing the purpose of F1 :

10% (b) Rewrite F1 so that it no longer calls F2, and now uses *iteration* rather than *recursion*:

7% 2. (a) Write a Python function `Distinct( s )` to return the string which is formed by the first occurrences of each item in string `s`. For example:

```
Distinct( "cacdabb" )   ⇒   "cadb"
```

6% (b) Fill in clear and concise comments for both of the following functions; the comments should explain the overall *purpose* of the functions, and not merely describe the code:

```
def G1( s ) :
    #
    return len( Distinct( s ) ) <= 1

def G2( s ) :
    #
    return len( Distinct( s ) ) == len( s )
```

7% (c) Write an alternative version of *either* `G1` *or* `G2`, which implements the very same test as its original version, but does so *without* calling the function `Distinct`:

---

3. A *date of birth* may be stored as a tuple of three numbers: `( day, month, year )`.

4% (a) Write a Python function `SameBirthday( dob1, dob2 )` which takes two dates of birth and tests if both yield the same *birthday* (the respective days and months must be equal, but not not necessarily years). Avoid using `if` statements, if possible.

4% (b) Write a Python function `IsOlderThan( dob1, dob2 )` which takes two dates of birth and tests if `dob1` occurs strictly earlier in time than `dob2`. Avoid using `if` statements, if possible.

2% (c) Assuming that all names are distinct, a *dictionary* may be used to link people's names with their dates of birth. Write down an example of such a dictionary, with three entries.

10% (d) Write a Python function `CommonPair( d )` which takes a dictionary of this form and writes out the names of *one* pair of people who have the same birthday, along with their common birthday, or else an appropriate message if no such pair exists.

20% 4. Suppose there is a file containing data on countries and their populations, with each line of the file consisting of the name of a country, a space, and the corresponding population. The file is non-empty, there are no spaces within country names (e.g., New Zealand is given as `NewZealand`), and all populations are distinct positive integers.

Write a Python function `Summary( datafile )` to read in such a file and print out the number of countries, the total and average populations, and the names of the countries with the largest and smallest populations, along with their respective populations.

Issue an appropriate error message if `datafile` cannot be read, and return directly.

20% 5. *Fixed Points of a List of Integers* : Identical to Question #5 on Spring Examination 2016.
