Introduction to Algorithms (CELEN086)

Problem Sheet 2

Topics: IF and Nested IF; Luhn algorithm; Sub-algorithm; Tracing your algorithm

1. Let $f(x)$ be a piecewise function defined by

$$f(x) = \begin{cases} x + 1; \ x > 4 \\ \sqrt{x}; \ 0 \leq x \leq 4 \\ 3 - 2x^2; \ -1 < x < 0 \\ 0; otherwise \end{cases}$$

   Write an algorithm using if-elseif-elseif-else structure for computing f(x).

2. For the same function in Q1, write an algorithm with nested if for computing f(x).

3. Write an algorithm called **isUnique()** that takes three numbers a, b, c, and returns 3 if all three numbers are equal; returns 2 if only two numbers are equal; otherwise returns 1.

4. How many test cases are needed for testing your algorithm isUnique() in Q3? Write each of your test cases (and trace it) to validate your algorithm.

5. Write an algorithm **div34()** that checks if a positive integer n is divisible by 3 or 4. It should return four kinds of messages depending on the value of n:
   - 'only divisible by 3'
   - 'only divisible by 4'
   - 'divisible by both'
   - 'divisible by neither'

6. Use Luhn algorithm to determine if the following mini card numbers are valid or not:
   i. 2389
   ii. 7120
   iii. 1958
   iv. 8949

7. Verify the Luhn algorithm using your own debit/credit card number (16-digit/19-digit). You should obtain a final sum, which is divisible by 10 (if not, contact your bank!).

   Note: skip this question if you do not have a bankcard. Always keep your data safe and do not share your personal information with others!

Introduction to Algorithms (CELEN086)

Problem Sheet 2

8. Consider the following algorithms (main and sub-algorithm)

Algorithm: **alg1 (n)** [main]

Requires: 3-digit positive integer n

Returns: True/False

```
1. let a = n%10, d = n/10
2. let b = d%10, c = d/10
3. let x = a+b+c
4. y = alg2(x)
5. if y==3 || y==6 || y==9
6.    return True
7. else
8.    return False
9. Endif
```

Algorithm: **alg2 (n)** [sub-algorithm]

Requires: an integer n

Returns: an integer

```
1. if n>=10
2.   return n%10 + n/10
3. else
4.   return n
5. endif
```

i. What are the results by calling alg1 (365) and alg1 (798)? Trace them carefully.

ii. What task does this algorithm solve? (You can trace more cases for conclusion.)

9. Write an algorithm called **minimum3( )** that finds the smallest value of three real numbers a, b and c.

10. Check your algorithm in Q9 thoroughly by designing different test cases and trace them.