ECM2418 Computer Languages and Representations
Continuous Assessment
Functional and Logic Programming

Dr David Wakeling

Handed out	Handed in
Thursday 26th October 2023 (T1:05)	Thursday 14th December 2023 (T1:12)

This Continuous Assessment is worth 40% of the module mark.

Question 1: Light Show

Every week, *The Sunday Times* newspaper publishes a Teaser. Teaser 3172, of Sunday 9th July 2023, was as follows.

> My bedside clock displays the time and date using eight digits; for example, at 9.43am on 28th June, the display would be
>
> 
>
>
> Each digit in the electronic display lights up some (or all) of seven light segments, the above display lighting up a total of 45 segments.
>
> On one occasion recently, the displayed digits were all different and the total number of lit segments was prime. The same was true exactly one day later. Then, just one minute after the second occasion, the number of lit segments was the average of the numbers of lit segments on those two previous occasions.
>
> **What was that third display?**

Question 1.1

Show a Haskell function `generator1` that returns a list of tuples (HR, MN, DY, MT) that may be solutions to the Teaser. That is, for which, HR, MN, DY and MT are valid hours, minutes, days (assume a non-leap year) and months.

This function will be assessed by the number of tests that it passes, as counted by the function `x_generator1` below. The expected answer is 10.

```
x_generator1 :: Int
x_generator1 =
  length [t | t <- ts, t `elem` g]
  where
  g  = generator1
  ts =
     [ ( 2,15,14,11)
     , ( 4,31,27, 9)
     , ( 6,47,10, 8)
     , ( 9, 3,23, 6)
     , (11,19, 6, 5)
     , (13,35,19, 3)
     , (15,51, 2, 2)
     , (18, 6,16,12)
     , (20,22,29,10)
     , (22,38,11, 9)
     ]
```

Question 1.2

Show a Haskell function `tester1` that returns true for tuples (HR, MN, DY, MT) that are solutions to the Teaser. That is, for which the tuple is "magic", a second tuple exactly one day later is also "magic", and just one minute on from this second tuple the number of lit segments on the display is the average of the number of lit segments of these two tuples. A tuple (HR, MN, DY, MT) is "magic" if the displayed digits of HR, MN, DY and MT are all be different, and the total number of lit segments in the display is prime.

This function will be assessed by the number of tests that it passes, as counted by the function `x_tester1` below. Note that these test cases were *NOT* produced by `generator1`. The expected answer is 10.

```
x_tester1 :: Int
x_tester1 =
  length [t | t <- ts, tester1 t]
  where
  ts =
     [ ( 6,59,17,24)
```

```
    , ( 6,59,17,34)
    , ( 6,59,27,14)
    , ( 6,59,27,41)
    , ( 8,59,12,46)
    , (16,59, 7,24)
    , (16,59, 7,42)
    , (16,59, 7,43)
    , (16,59,27,40)
    , (18,59, 2,46)
    ]
```

Question 1.3

On `blue18.ex.ac.uk`, my program

```haskell
main :: IO ()
main =
  print (filter tester1 generator1)
```

computes `[(16,59,27,4)]`, from which one can deduce the answer to Teaser 3172 is

in 0.004 seconds. Tune your program so that on the same machine, it computes this answer within 1.000 seconds.

(5 marks)

Question 2: Digital Trio

Teaser 3158, of Sunday 5th May 2023, was as follows.

> "I have a couple of subtraction problems for you", George told Martha. Look: N1 - N2 = N3 and N3 - N4 = N5. Can you solve them if I tell you that N1, N3 and N5 are all three-digit whole numbers whose sum is less than 2000, the same three non-zero digits appearing in all three numbers but no digit being repeated within any of those numbers? N2 and N4 are both two-digit whole numbers using two of the three digits mentioned above, and the first digit of N1 is not equal to the first digit of N2.
>
> **What is N1?**

Question 2.1

Show a Haskell function `generator2` that returns a list of tuples (*N1*, *N2*, *N3*, *N4*, *N5*) that may be solutions to the Teaser. That is, for which, *N1*, *N3* and *N5* are three-digit numbers, and *N2* and *N4* are two-digit numbers. The same three digits appear in *N1*, *N3* and *N5*, two of these digits appear in *N2*, and two of them appear in *N4*. In each number, no digit is zero and none is repeated. The first digit of *N1* is not equal to the first digit on *N2*.

This function will be assessed by the number of tests that it passes, as counted by the function `x_generator2` below. The expected answer is 10.

```
x_generator2 :: Int
x_generator2 =
  length [t | t <- ts, t `elem` g]
  where
  g  = generator2
  ts =
     [ ("123","21","123","12","123")
     , ("162","26","261","12","621")
     , ("219","19","912","21","291")
     , ("329","92","932","32","239")
     , ("439","94","394","43","394")
     , ("549","95","945","95","945")
     , ("568","68","586","56","586")
     , ("769","67","679","97","796")
     , ("879","79","897","98","789")
     , ("987","79","789","79","789")
     ]
```

(10 marks)

Question 2.2

Show a Haskell function `tester2` that returns true for tuples (*N1*, *N2*, *N3*, *N4*, *N5*) that are solutions to the Teaser. That is, for which $N1 - N2 = N3$, $N3 - N4 = N5$ and $N1 + N3 + N5 < 2000$.

This function will be assessed by the number of tests that it passes, as counted by the function `x_tester2` below. The expected answer is 10.

```
x_tester2 :: Int
x_tester2 =
  length [t | t <- ts, tester2 t]
  where
  ts =
     [ ("138","01","137","50","87")
```

```
    , ("143","01","142","52","90")
    , ("171","02","169","79","90")
    , ("152","03","149","54","95")
    , ("159","04","155","61","94")
    , ("161","05","156","63","93")
    , ("182","06","176","80","96")
    , ("151","07","144","57","87")
    , ("165","08","157","64","93")
    , ("174","09","165","71","94")
    ]
```

(10 marks)

Question 2.3

On `blue18.ex.ac.uk`, my program

```
main :: IO ()
main =
  print (filter tester2 generator2)
```

computes `[("594","45","549","54","495")]` in ~~0.003 seconds~~ 1.832 seconds. Tune your program so that on the same machine, it computes this answer within ~~1.000 seconds~~ 5.000 seconds.

(5 marks)

Question 3: Easier to Ask the Audience

Teaser 3145, of Sunday 1st January 2023, was as follows.

> "I have forgotten the phone number!" complained Martha, about to phone a friend. "So have I!" replied George, "but I have some vague memories of it. It is a perfect square with all the digits different, and the last digit is equal to the number of digits to be dialled. The last-but-one digit is odd and one of the digits is zero. Also the second and third and last-but-one digits are all exact multiples of the first digit. Maybe you can work it out."
>
> Martha proceeded to dial the number correctly.
>
> **What number did she dial?**

Question 3.1

Show a Prolog predicate `generator3` that yields successive numbers N between 1,000 to 1,000,000 (inclusive) that may be solutions to the Teaser. That is, integers N that are perfect squares.

This predicate will be assessed by the number of tests that it passes, as counted by the predicate `x_generator3` below. The expected answer is 10.

```
x_generator3( N ) :-
  x_generator3_loop(
      [   1024,    9409,  23716,  51529
      , 123904,  185761, 868624, 962361
      , 982081, 1000000 ], 0, N ).

x_generator3_loop( [], C, C ).
x_generator3_loop( [T|TS], C, N ) :-
  generator3( T ),
  C1 is C + 1,
  x_generator3_loop( TS, C1, N ).
x_generator3_loop( [_|TS], C, N ) :-
  x_generator3_loop( TS, C, N ).
```

(10 marks)

Question 3.2

Show a Prolog predicate `tester3` that is true for phone numbers N that are solutions to the Teaser. That is, for integers N where all of the digits are different, the last digit is equal to the number of digits, the last-but-one digit is odd and one of the digits is zero. In addition, the second and third and last-but-one digits are all exact multiples of the first digit.

This predicate will be assessed by the number of tests that it passes, as counted by a predicate `x_tester3` below. The expected answer is 10.

```
x_tester3( N ) :-
  x_tester3_loop(
      [ 123056, 128036, 139076, 142076
      , 148056, 159076, 173096, 189036
      , 193056, 198076 ], 0, N ).

x_tester3_loop( [], C, C ).
x_tester3_loop( [T|TS], C, N ) :-
  tester3( T ),
  C1 is C + 1,
  x_tester3_loop( TS, C1, N ).
```

```
x_tester3_loop( [_|TS], C, N ) :-
  x_tester3_loop( TS, C, N ).
```

(10 marks)

Question 3.3

On Swish Prolog, my program

```
main :-
  generator3( N ), tester3( N ), write( N ).
```

computes `173056` in 0.40 seconds. Tune your program so that on the same system, it computes this answer within 2.00 seconds.

(5 marks)

Question 4: Cube Route

Teaser 3149, of Sunday 29th January 2023, was as follows.

> I have a set of ten cards, each of which has a different digit written on it. All the cards have been used to make a set of prime numbers. After discarding the smallest prime, and without changing the order of any cards, I have placed the remaining primes in order of decreasing size to give a large number. It is possible, without changing the order of any cards, to break this number into a set composed entirely of cubes. Neither set contains a number with more than four digits.
>
> **List, in order of decreasing size, my set of prime numbers.**

Question 4.1

Show a Prolog predicate `generator4` that yields arrangements of the digits 0 to 9 divided into runs of one, two, three or four digits that form prime numbers. Importantly (and somewhat surprisingly) leading zero digits do not count, so "`251`" is considered to be prime, but "`0251`" is not.

This predicate will be assessed by the number of tests that it passes, as counted by a predicate `x_generator4` below. The expected answer is 10.

```
x_generator4( N ) :-
  x_generator4_loop(
      [ [[9,6,7],[4,0,1],[2,8,3],[5]]
      , [[9,8,3],[6,0,1],[5],[4,7],[2]]
      , [[9,8,3],[6,7],[4,2,0,1],[5]]
      , [[9,8,5,1],[2],[4,3],[6,0,7]]
      , [[9,8,5,1],[2],[3],[6,0,4,7]]
      , [[9,8,5,1],[2],[7],[4,6,0,3]]
      , [[8,9],[7],[6,0,1],[2,5,4,3]]
      , [[8,9],[7],[5,6,3],[4,0,2,1]]
      , [[8,9],[5],[4,7],[6,0,1],[3],[2]]
      , [[3],[5],[6,0,7],[2],[4,1],[8,9]] ], 0, N ).

x_generator4_loop( [], C, C ).
x_generator4_loop( [T|TS], C, N ) :-
  generator4( T ),
  C1 is C + 1,
  x_generator4_loop( TS, C1, N ).
x_generator4_loop( [_|TS], C, N ) :-
  x_generator4_loop( TS, C, N ).
```

(10 marks)

Question 4.2

Show a Prolog predicate `tester4` that is true for lists of lists of digits that form prime numbers and may be solutions to the Teaser. That is, for collections of prime numbers that after discarding the smallest prime, may be arranged in order of decreasing size to give a large number that may be divided into runs of one, two three or four digits that form cubes.

This predicate will be assessed by the number of tests that it passes, as counted by the predicate `x_tester4` below. The expected answer is 10.

```
x_tester4( N ) :-
  x_tester4_loop(
      [ [[8,2,7],[6,1],[5,3],[4,0,9]]
      , [[8,2,7],[6,1],[4,0,9],[5,3]]
      , [[8,2,7],[5,3],[6,1],[4,0,9]]
      , [[8,2,7],[4,0,9],[6,1],[5,3]]
      , [[6,1],[8,2,7],[4,0,9],[5,3]]
      , [[6,1],[4,0,9],[5,3],[8,2,7]]
      , [[5,3],[6,1],[4,0,9],[8,2,7]]
      , [[5,3],[4,0,9],[6,1],[8,2,7]]
      , [[4,0,9],[5,3],[8,2,7],[6,1]]
      , [[4,0,9],[8,2,7],[6,1],[5,3]] ], 0, N ).
```

```
x_tester4_loop( [], C, C ).
x_tester4_loop( [T|TS], C, N ) :-
  tester4( T ),
  C1 is C + 1,
  x_tester4_loop( TS, C1, N ).
x_tester4_loop( [_|TS], C, N ) :-
  x_tester4_loop( TS, C, N ).
```

(10 marks)

Question 4.3

On Swish Prolog, my program

```
main :-
    generator4( XS ), tester4( XS ), write( XS ).
```

computes a first result in in 33.0 seconds, from which one can deduce the answer to the Teaser is 827, 409, 61, 53. Tune your program so that on the same system, it computes this answer within 120.0 seconds.

(5 marks)

Submission

You should submit a single ".zip" file to the ELE system. Other compression formats, such as ".rar", ".7z", ".gz" and ".bz2" are unacceptable, and will receive a mark of zero. The ".zip" file should contain four completed text files "`Light.hs`" (containing the answer to Question 1), "`Trio.hs`" (containing the answer to Question 2), "`Audience.pl`" (containing the answer to Question 3) and "`Cube.pl`" (containing the answer to Question 4).

If there is any question as to whether your functional programs compute the correct result, these questions will be answered on the implementation at

`https://www.tutorialspoint.com/compile_haskell_online.php`

If there is any question as to whether your logic programs compute the correct result, these questions will be answered on the implementation at

`https://swish.swi-prolog`

All students are reminded of the University regulations on academic honesty and plagiarism.

In particular, functions an predicates clearly intended ONLY to pass the given tests will be treated as malpractice ("an attempt to deceive the examiners").