1. Write a DCG that accepts strings of the form $u2v$ where $u$ and $v$ are strings over the alphabet $\{0,1\}$ such that the number of 0’s in $u$ is the same as the number of 1’s in $v$. For example,

```prolog
?- s([0,1,1,2,0,0,1,0],L).
L = [];
L = [0];
no
```

2. Exercise 6.6 in Learn Prolog Now describes a street with

\((\ast)\) three neighbouring houses that all have a different colour, namely red, blue, and green. People of different nationalities live in the different houses and they all have a different pet.

Leaving out all the other constraints mentioned in that exercise, write a DCG that outputs strings

\([\text{Col1},\text{Nat1},\text{Pet1}, \text{Col2},\text{Nat2},\text{Pet2}, \text{Col3},\text{Nat3},\text{Pet3}]\)

satisfying \((\ast)\), where the nationalities are english, spanish, japanese and the pets are jaguar, snail, zebra. For example,

```prolog
?- s([red,english,snail, blue,japanese,jaguar, green,spanish,Z], []).
Z = zebra;
no
```

3. Write a DCG that given a non-negative integer \texttt{Sum}, accepts lists of integers $\geq 1$ that add up to \texttt{Sum}. For example,

```prolog
?- s(3,L,[]).
L = [2,1] ? ;
L = [1,2] ? ;
L = [1,1,1] ? ;
no
```

It may be useful to write a predicate \texttt{mkList(+Num,?List)} that returns a list \texttt{List} of integers from \texttt{Num} down to 1. For example,

```prolog
?- mkList(3,L).
L = [3,2,1] ? ;
```

Be sure you understand how the DCG clauses translate to ordinary Prolog clauses with difference lists.

---

\(^1\)Submit to Blackboard by Tuesday, Dec 5. For any extensions beyond that date, email your demonstrator, David Woods, dwoods@tcd.ie.