Suppose we want to write a Prolog program `double` that, given a list of items $I_i$, produces a list of elements $A_i$, such that $A_i$ is a list of two items, $[I_i, I_i]$. Fill in the blanks in its definition:

```
double([], __________).

double([I|L], __________) :-
__________________________.
```

2 The Prolog query `?- append(X, X, L).` binds $X$ to a list that, when appended to itself, yields $L$. For instance, if $L = [a, b, a, b]$, the query would bind $X$ to $[a, b]$. (We’ll assume that $L$ is a list of terms containing no variables.) How would you express a query to find two unequal lists $Y$ and $Z$ that when appended yield $L$? For example, if $L$ were again $[a, b, a, b]$, the answers should be $Y=[]$, $Z=[a, b, a, b]$; $Y=[a]$, $Z=[b, a, b]$; $Y=[a, b, a]$, $Z=[b]$; and $Y=[a, b, a, b]$, $Z=[b]$. Write the query in this box:
What language does the following DCG accept?

\[
\text{mystery} \rightarrow [\_]. \\
\text{mystery} \rightarrow [A, A]. \\
\text{mystery} \rightarrow [A], \text{mystery}, [A].
\]

Write your answer in the following box: