

Declension (Noun/Pronoun).

In Sanskrit, for a noun word, its inflected forms are generated in form of Declension table representing number (singular, dual and plural), case and gender (masculine, feminine, neuter) of the noun. Sanskrit has eight cases: nominative, accusative, instrumental, dative, ablative, genitive, locative and vocative, whereas for a pronoun word there exist seven cases excluding vocative.

For a noun/pronoun word , input to the declension function are the noun/pronoun stem in 'Devanagari' and its gender('masculine', 'feminine', 'neuter'), output is a declension table in 'Devanagari', no case and number names will be in written, by default above case and number sequence is followed. If any pronoun word is independent of gender like ('अस्मद्', 'युष्मद्', etc.), then there is no need of second argument to the above function.

```
In [1]: from Declension import Declension
```

```
In [2]: Declension('नदी','feminine')
```

```
Out [2]: नदी नद्यौ नद्यः
नदीम् नद्यौ नदीः
नद्या नदीभ्याम् नदीभिः
नद्यै नदीभ्याम् नदीभ्यः
नद्याः नदीभ्याम् नदीभ्यः
नद्याः नद्योः नदीनाम्
नद्याम् नद्योः नदीषु
हे नदि हे नद्यौ हे नद्यः
```

```
In [3]: Declension('तद्','neuter')
```

```
Out [3]: तत् ते तानि
तत् ते तानि
तेन ताभ्याम् तैः
तस्मै ताभ्याम् तेभ्यः
तस्मात् ताभ्याम् तेभ्यः
तस्य तयोः तेषाम्
तस्मिन् तयोः तेषु
```

```
In [4]: Declension('युष्मद्') # no gender required
```

```
Out [4]: त्वम् युवाम् यूयम्
त्वाम् युवाम् युष्मान्
त्वया युवाभ्याम् युष्माभिः
तुभ्यम् युवाभ्याम् युष्मभ्यः
त्वत् युवाभ्याम् युष्मत्
तव युवयोः युष्माकम्
त्वयि युवयोः युष्मासु
```

Stem Lookup.

Stemming is a core task for NLP which relates different forms of a given word to a main noun word. For highly inflected languages like Sanskrit, the task is even more ambiguous as words often have around two dozen possible forms which also varies depending on gender and their ending character.

Here, It reduces all possible inflected words to their word stem/main noun.

The function takes only inflected word in 'Devanagari' as input and return its word stem in same.

```
In [1]: from Stemming import stem
```

```
In [2]: stem('क्षुत्सु')
```

```
Out [2]: क्षुध्
```

```
In [3]: stem('अस्य')
```

```
Out [3]: इदम्
```

```
In [4]: stem('विद्वद्भ्यः')
```

```
Out [4]: विद्वस्
```

Case & Number.

For any noun, Sanskrit has eight cases: nominative, accusative, instrumental, dative, ablative, genitive, locative and vocative and three number forms (Singular, Dual and Plural), depending on them each inflected form has a meaning depicting its role (in the language structure) and quantity, whereas for pronoun word there exist only seven cases excluding vocative, as simply pronoun themselves are used to refer some noun word.

The function takes inflected word in 'Devanagari' and languages ('hi' for Hindi or 'eng' for English) as arguments and return a tuple containing its case and number at first occurrence in its declension table in defined language.

```
In [1]: from Stemming import case_number
```

```
In [2]: case_number('पोतृन्', 'eng')  
Out [2]: ('Accusative', 'Plural')
```

```
In [3]: case_number('अस्य', 'hi')  
Out [3]: ('षष्ठी', 'एकवचन')
```

```
In [4]: case_number('विद्वांसौ', 'hi')  
Out [4]: ('प्रथमा', 'द्विवचन')
```

```
In [5]: case_number('राज्ञे', 'eng')  
Out [5]: ('Dative', 'Singular')
```
