## Gr 9 : Furkan Salman, Kenan Mukendi Kayembe

# Lab 7 Report

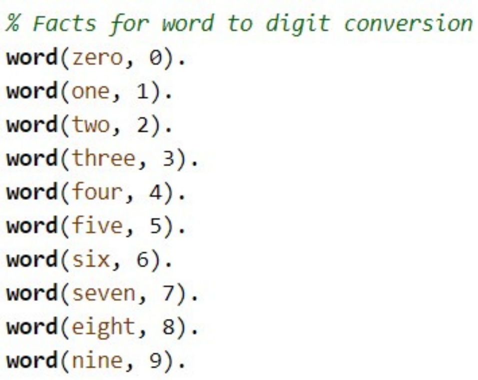
# Problem:

Convert English words of a written number (up to a thousand) into numerical digits N, where N <= 1000.

# Solution:

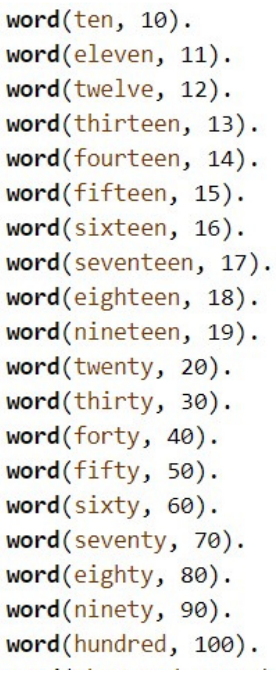
## Facts:

We need to define digits as facts since they are the fundamental components of other numbers.

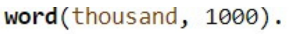


```
% Facts for word to digit conversion
word(zero, 0).
word(one, 1).
word(two, 2).
word(three, 3).
word(four, 4).
word(five, 5).
word(six, 6).
word(seven, 7).
word(eight, 8).
word(nine, 9).
```

There are numbers that cannot be formed by combining atomic digits. These numbers should also be defined as facts.



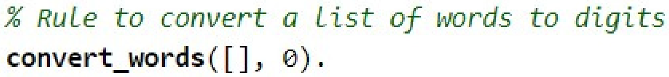
```
word(ten, 10).
word(eleven, 11).
word(twelve, 12).
word(thirteen, 13).
word(fourteen, 14).
word(fifteen, 15).
word(sixteen, 16).
word(seventeen, 17).
word(eighteen, 18).
word(nineteen, 19).
word(twenty, 20).
word(thirty, 30).
word(forty, 40).
word(fifty, 50).
word(sixty, 60).
word(seventy, 70).
word(eighty, 80).
word(ninety, 90).
word(hundred, 100).
```



```
word(thousand, 1000).
```

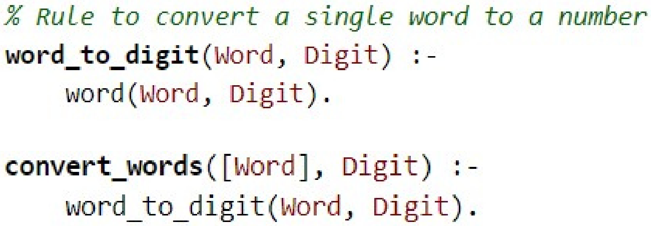
## Rules:

- If no words are provided as input, the conversion result is 0.



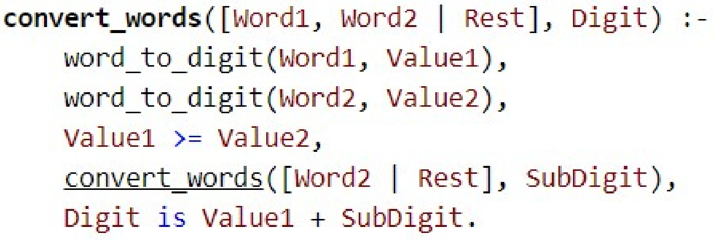
```
% Rule to convert a list of words to digits
convert_words([], 0).
```

- If only one word is provided as input, we can directly obtain its value from the predefined facts. The word can represent either a single digit or one of the special numbers defined in the facts.



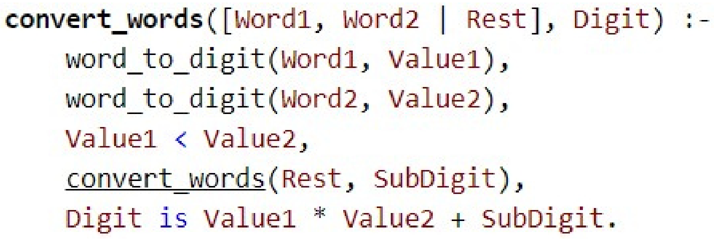
```
% Rule to convert a single word to a number
word_to_digit(Word, Digit) :-
    word(Word, Digit).

convert_words([Word], Digit) :-
    word_to_digit(Word, Digit).
```

- If more than one word is provided, we compare the values of the first and second words. If the value of the first word is greater, it indicates that we need to add the value obtained from the rest of the words to the value of the first word. We exclude the first word and recursively call the convert_words function for the remaining words. Finally, we add the value obtained from the rest to the value of the first word. Example: For the input "hundred seventy five," since 100 > 70 and 70 > 5, we calculate rest = 70 + 5 and firstWord = 100. The final value is obtained by adding 100 to 75.



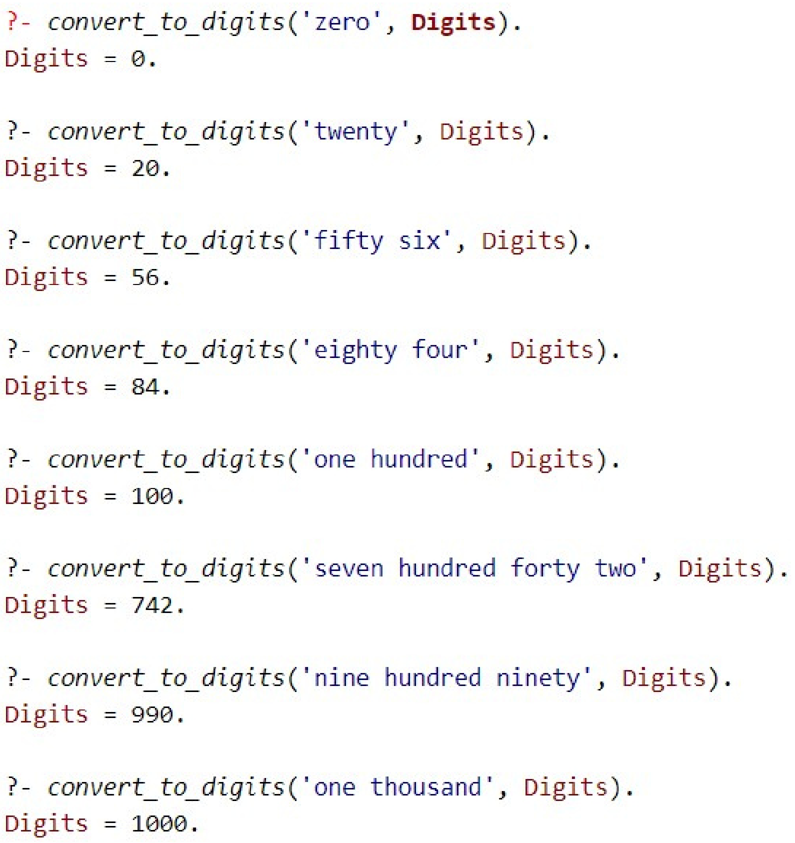
```
convert_words([Word1, Word2 | Rest], Digit) :-
    word_to_digit(Word1, Value1),
    word_to_digit(Word2, Value2),
    Value1 >= Value2,
    convert_words([Word2 | Rest], SubDigit),
    Digit is Value1 + SubDigit.
```

- If the value of the first word is not greater than the value of the second word, it indicates that the first word is used to indicate how many times the value of the second word is repeated. Therefore, we multiply the value of the first word by the value of the second word and recursively call the function for the rest of the words. We then add the value obtained from the rest to get the final value for the input. Example: For the input "three hundred five" since 3 < 100, we calculate 3 * 100 + 5 = 305.



```
convert_words([Word1, Word2 | Rest], Digit) :-
    word_to_digit(Word1, Value1),
    word_to_digit(Word2, Value2),
    Value1 < Value2,
    convert_words(Rest, SubDigit),
    Digit is Value1 * Value2 + SubDigit.
```

# Test Cases and Results:



```
?- convert_to_digits('zero', Digits).
Digits = 0.

?- convert_to_digits('twenty', Digits).
Digits = 20.

?- convert_to_digits('fifty six', Digits).
Digits = 56.

?- convert_to_digits('eighty four', Digits).
Digits = 84.

?- convert_to_digits('one hundred', Digits).
Digits = 100.

?- convert_to_digits('seven hundred forty two', Digits).
Digits = 742.

?- convert_to_digits('nine hundred ninety', Digits).
Digits = 990.

?- convert_to_digits('one thousand', Digits).
Digits = 1000.
```