In [10]:
```
from nltk.tokenize import word_tokenize
from transformers import AutoTokenizer

text = "unbelievable success"


word_tokens = word_tokenize(text)


tokenizer = AutoTokenizer.from_pretrained("bert-base-uncased")
hybrid_tokens = []
for word in word_tokens:
    subwords = tokenizer.tokenize(word)
    hybrid_tokens.extend(subwords)

print("Hybrid Tokenization:", hybrid_tokens)
```

```
Hybrid Tokenization: ['unbelievable', 'success']
```

In [13]:
```
import nltk
nltk.download('punkt')
from nltk.tokenize import word_tokenize
text = "This is an example sentence, showing off the tokenization process "
tokens = word_tokenize(text)
print(tokens)
```

```
['This', 'is', 'an', 'example', 'sentence', ',', 'showing', 'off', 'the', 'tokeniza
tion', 'process']

[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\faree\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

In [15]:
```
from collections import Counter

def calculate_word_frequency(text):

    words = text.lower().split()


    word_counts = Counter(words)


    for word, count in word_counts.items():
        print(f"{word}: {count}")


text = input("Enter a text: ")
calculate_word_frequency(text)
```

```
Enter a text: hello bittesh
hello: 1
bittesh: 1
```

In [16]:
```
import re
from collections import Counter

def calculate_word_frequency_advanced(text, sort_by="word"):
    """
    Calculate and print the frequency of each word in a given text.

    Args:
        text (str): The input text to analyze.
        sort_by (str): "word" to sort by alphabetical order of words,
                       "frequency" to sort by descending frequency.
    """

    words = re.findall(r'\b\w+\b', text.lower())


    word_counts = Counter(words)


    if sort_by == "frequency":
        sorted_word_counts = sorted(word_counts.items(), key=lambda x: x[1], reverse
    else:
        sorted_word_counts = sorted(word_counts.items())


    for word, count in sorted_word_counts:
        print(f"{word}: {count}")


if __name__ == "__main__":
    text = input("Enter a text: ")
    sort_option = input("Sort by 'word' or 'frequency': ").strip().lower()
    if sort_option not in {"word", "frequency"}:
        sort_option = "word"
    calculate_word_frequency_advanced(text, sort_by=sort_option)
```

```
Enter a text: hello mallareddy university
Sort by 'word' or 'frequency': word
hello: 1
mallareddy: 1
university: 1
```

In [17]:
```
import re
from collections import Counter

def calculate_word_frequency_advanced(text, sort_by="word"):
    """
    Calculate and print the frequency of each word in a given text.

    Args:
        text (str): The input text to analyze.
        sort_by (str): "word" to sort by alphabetical order of words,
                       "frequency" to sort by descending frequency.
    """

    words = re.findall(r'\b\w+\b', text.lower())


    word_counts = Counter(words)


    if sort_by == "frequency":
        sorted_word_counts = sorted(word_counts.items(), key=lambda x: x[1], reverse
    else:
        sorted_word_counts = sorted(word_counts.items())


    for word, count in sorted_word_counts:
        print(f"{word}: {count}")


if __name__ == "__main__":
    text = input("Enter a text: ")
    sort_option = input("Sort by 'word' or 'frequency': ").strip().lower()
    if sort_option not in {"word", "frequency"}:
        sort_option = "word"
    calculate_word_frequency_advanced(text, sort_by=sort_option)
```

```
Enter a text: very good morning nlp
Sort by 'word' or 'frequency': frequency
very: 1
good: 1
morning: 1
nlp: 1
```

In [ ]: