BDA Test

Name: Ritu Pal

Enrollment NO.: 230297

Test 1 Task

In [1]:
```python
from collections import defaultdict

# Mapper: splits lines into words and emits (word, 1)
def mapper(file_path):
    mapped = []
    with open(file_path, 'r') as f:
        for line in f:
            for word in line.strip().split():
                mapped.append((word.lower(), 1))
    return mapped

# Reducer: sums counts for each unique word
def reducer(mapped_data):
    reduced = defaultdict(int)
    for word, count in mapped_data:
        reduced[word] += count
    return reduced

if __name__ == "__main__":
    file_path = "data1.txt"

    mapped_data = mapper(file_path)
    reduced_data = reducer(mapped_data)

    # Sort the reduced data by frequency
    sorted_words = sorted(reduced_data.items(), key=lambda x: x[1], reverse=True)

    # Output Top 10
    print("Top 10 Most Frequent Words:")
    for word, freq in sorted_words[:10]:
        print(f"{word}: {freq}")
```

```
Top 10 Most Frequent Words:
the: 6
fox: 3
dog: 3
quick: 1
brown: 1
jumps: 1
over: 1
lazy: 1
barked: 1
at: 1
```

Test 3 Task

In [2]:
```python
from collections import Counter

STOPWORDS = {"the", "and", "of", "to", "in", "is", "it", "that", "for", "a", "at",

# Mapper function
def mapper(file_paths):
    if isinstance(file_paths, str):
        file_paths = [file_paths]
    words = []
    for path in file_paths:
        with open(path, 'r') as f:
            for word in f.read().lower().split():
                if word in STOPWORDS or len(word) <= 5:
                    continue
                words.append(word)
    return words

# Reducer function
def reducer(words):
    return Counter(words)

if __name__ == "__main__":
    # Multiple files
    files = ["data1.txt", "sample_data.txt"]

    # Map + Reduce
    filtered_words = mapper(files)
    counts = reducer(filtered_words)

    # Top 5 least frequent
    least_frequent = sorted(counts.items(), key=lambda x: x[1])[:5]

    print("Top 5 least common words with > 5 characters:")
    for word, freq in least_frequent:
        print(f"{word}: {freq}")
```

```
Top 5 least common words with > 5 characters:
barked: 1
chased: 1
hadoop_0: 1000
hadoop_1: 1000
hadoop_2: 1000
```