```
# 1. How does a Distributed File System protect against data loss due to a
computer crash? Answer in 2-3 sentences.
print("A Distributed File System replicates data across multiple nodes. If one
node crashes, the data can still be accessed from other replicas, protecting
against data loss.")
```

```
A Distributed File System replicates data across multiple nodes. If one node
crashes, the data can still be accessed from other replicas, protecting against
data loss.
```

```
# 2. Consider the problem of counting all length-3 sequences, including
overlapping sequences, i.e., the number of occurrences of AGA, GAT, ATA, … using
the following Map-Reduce program.
def map_fn(chunk):
    for i in range(len(chunk)-2):
        w = chunk[i:i+3]
        yield (w, 1)

def reduce_fn(key, values):
    yield (key, sum(values))


# Below is a string containing a DNA sequence:
dna = "AGATAGA"

# a) What are ALL the key, value pairs output by the Map step?
map_output = list(map_fn(dna))
print("a) What are ALL the key, value pairs output by the Map step?")
for kv in map_output:
    print(kv)

# b) What are ALL the key, value pairs input to the Reduce steps?
from collections import defaultdict
grouped = defaultdict(list)
for k, v in map_output:
    grouped[k].append(v)

print("\nb) What are ALL the key, value pairs input to the Reduce steps?")
for k, vs in grouped.items():
    print(k, vs)

# c) What is the output after the Reduce steps?
reduce_output = []
for k, vs in grouped.items():
    reduce_output.extend(list(reduce_fn(k, vs)))

print("\nc) What is the output after the Reduce steps?")
```

```
for kv in reduce_output:
    print(kv)
```

a) What are ALL the key, value pairs output by the Map step?
('AGA', 1)
('GAT', 1)
('ATA', 1)
('TAG', 1)
('AGA', 1)

b) What are ALL the key, value pairs input to the Reduce steps?
AGA [1, 1]
GAT [1]
ATA [1]
TAG [1]

c) What is the output after the Reduce steps?
('AGA', 2)
('GAT', 1)
('ATA', 1)
('TAG', 1)

```
import pandas as pd
from collections import defaultdict

# 3. Consider the following Map-Reduce program to process a data frame.
def map_fn(chunk):
    inputvalue = chunk.squeeze()
    if inputvalue.arr_delay > 30:
        yield (inputvalue.dest, 1)

def reduce_fn(key, values):
    yield (key, ())

# If the input data frame is:
data = pd.DataFrame({
    "flight": ["UA", "UA", "DL", "DL"],
    "carrier": [345, 452, 428, 567],
    "arr_delay": [20, 38, 4, 42],
    "dest": ["LAX", "MIA", "MIA", "ORD"]
})

# a. How many times will map() be executed?
map_executions = len(data)
print("a. How many times will map() be executed?\n", map_executions)

# b. What are ALL the (key, value) pairs output by the Map functions?
```

```
map_output = []
for i in range(len(data)):
    chunk = data.iloc[[i]]
    for kv in map_fn(chunk):
        map_output.append(kv)

print("\nb. What are ALL the (key, value) pairs output by the Map functions?")
for kv in map_output:
    print(kv)

# c. What are ALL the (key, value) pairs input to the Reduce functions?
grouped = defaultdict(list)
for k, v in map_output:
    grouped[k].append(v)

print("\nc. What are ALL the (key, value) pairs input to the Reduce functions?")
for k, vs in grouped.items():
    print(k, vs)

# d. What is the output (keys only) output by the Reduce functions?
reduce_output = []
for k, vs in grouped.items():
    for out in reduce_fn(k, vs):
        reduce_output.append(out)

print("\nd. What is the output (keys only) output by the Reduce functions?")
for kv in reduce_output:
    print(kv)

# e. What does the above map-reduce program do?
print("\ne. What does the above map-reduce program do?\nThis map-reduce program
filters rows by arr_delay > 30 and groups them by dest, effectively identifying
destinations with delayed flights.")
```

```
a. How many times will map() be executed?
 4

b. What are ALL the (key, value) pairs output by the Map functions?
('MIA', 1)
('ORD', 1)

c. What are ALL the (key, value) pairs input to the Reduce functions?
MIA [1]
ORD [1]

d. What is the output (keys only) output by the Reduce functions?
('MIA', ())
```

```
('ORD', ())
```

e. What does the above map-reduce program do?
This map-reduce program filters rows by arr_delay > 30 and groups them by dest, effectively identifying destinations with delayed flights.