Item 1

Questions Data structures

You're tasked with developing a simple video rental store management system. The system uses lists to track which movies have been rented and returned. The lists `movies_rented` and `movies_returned` store the titles of the movies rented out and returned throughout the day, respectively, in ascending order of rental/return (i.e. the first movie rented/returned is the first element of the list). For now, **suppose the store has one** (**and only one**) **copy of all movies in existence** (it's a big video club). For example:

```
movies_rented = ["The Matrix", "Inception", "Toy Story", "The Lion King"]
movies_returned = ["Toy Story", "The Lion King"]
```

Given these lists, answer the following questions.

Which of the following answers the question "What was the first movie to be rented?"

A `movies_returned[0]`

B `movies_returned[1]`

[✓ C] `movies_rented[0]`

D `movies_returned[1]`

Which of the following answers the question "What was the last movie returned?"

A `movies_rented[-1]`

[✓ B] `movies_returned[-1]`

C `movies_returned[0]`

D `movies_returned[2]`

Which of the following answers the question "How many movies have been rented today?"

[✓ A]

```
len(movies_rented)
```

B

```
len(movies_rented) - 1
```

C

```
len(movies_rented) - len(movies_returned)
```

D

```
len(movies_returned)
```

Which of the following answers the question "Is there any movie that was not yet returned"?

A

```
len(movies_rented) == len(movies_returned)
```

B

```
len(movies_rented) - len(movies_returned) == 1
```

C

```
len(movies_rented) - len(movies_returned) > 1
```

[✓ D]

```
len(movies_rented) - len(movies_returned) > 0
```

Which of the following answers the question "Is The Matrix available for renting?":

[✓ A]

```
("The Matrix" not in movies_rented) or ("The Matrix" in movies_returned)
```

B

```
"The Matrix" in movies_rented
```

C

```
("The Matrix" in movies_rented) and ("The Matrix" not in movies_returned)
```

D

```
("The Matrix" in movies_rented) or ("The Matrix" in movies_returned)
```

Which of the following correctly answers the question "Is The Matrix currently rented?"

A

```
"The Matrix" not in movies_returned
```

B

```
"The Matrix" in movies_rented
```

C

```
("The Matrix" in movies_rented) or ("The Matrix" not in movies_returned)
```

[✓ D]

```
("The Matrix" in movies_rented) and ("The Matrix" not in movies_returned)
```

Which of the following correctly records the return of Inception?:

A

```
movies_rented.append("Inception")
```

B

```
movies_returned.add("Inception")
```

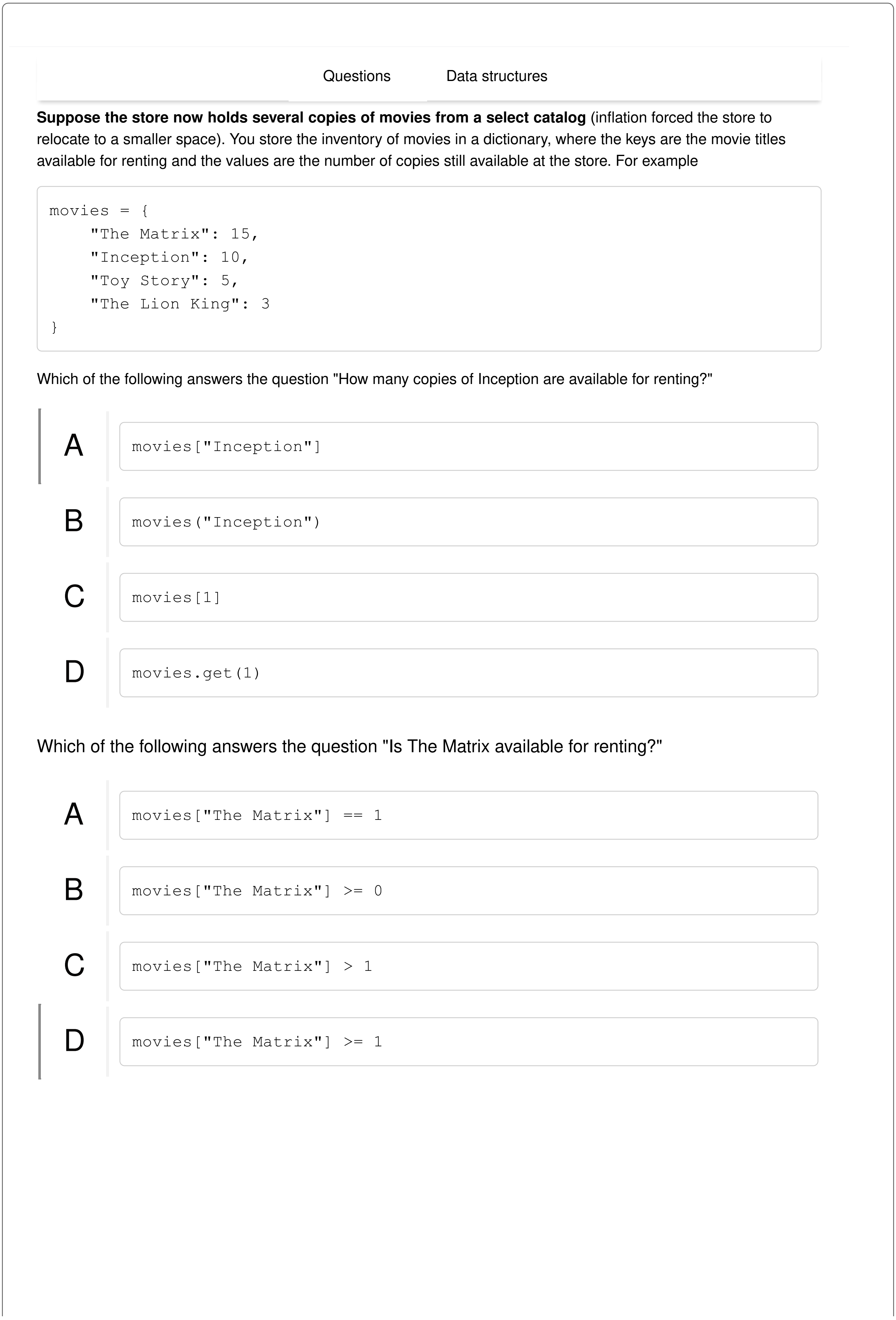
[✓ C]

```
movies_returned.append("Inception")
```

D

```
movies_returned + "Inception"
```

Item 2



Suppose the store now holds several copies of movies from a select catalog (inflation forced the store to relocate to a smaller space). You store the inventory of movies in a dictionary, where the keys are the movie titles available for renting and the values are the number of copies still available at the store. For example

```
movies = {
    "The Matrix": 15,
    "Inception": 10,
    "Toy Story": 5,
    "The Lion King": 3
}
```

Which of the following answers the question "How many copies of Inception are available for renting?"

[✓ A]

```
movies["Inception"]
```

B

```
movies("Inception")
```

C

```
movies[1]
```

D

```
movies.get(1)
```

Which of the following answers the question "Is The Matrix available for renting?"

A

```
movies["The Matrix"] == 1
```

B

```
movies["The Matrix"] >= 0
```

C

```
movies["The Matrix"] > 1
```

[✓ D]

```
movies["The Matrix"] >= 1
```

What is the output of the following code?

```
if "The Matrix" in movies:
  print("Found")
else:
  print("Not found")
if movies["Inception"] > 10:
  print("Found")
elif "The Lion King" in movies:
  print("Found")
if "Triangle of Sadness" in movies:
  print("Found")
else:
  print("Not found")
```

A

```
Found
Found
Not found
```

B

```
Not found
Found
```

C

```
Not found
Found
Found
Not found
```

D

```
Not found
Found
Found
Found
```

What is the output of the following code?

```
for title, qnt in movies.items():
  if 0 < qnt < 10 or qnt > 15:
    print(title)
    break
```

A

```
Toy Story
```

B

```
The Matrix
Inception
Toy Story
The Lion King
```

C

```
Toy Story
The Lion King
```

D

Nothing

What is the output of the following code?

```
i = 1
values = list(movies.values())
while i > len(values):
  print(values[i])
  i += 2
```

A

```
The Matrix
Toy Story
```

B

```
The Matrix
Inception
Toy Story
The Lion King
```

C

```
Inception
The Lion King
```

D

Nothing ✓

Which of the following answers the question "Which movie titles are currently available for renting?"

A

```
available = []
count = 0
  while count < len(movies):
    if movies[count] >= 0:
      available.append(count)
print(available)
```

B

```
available = []
for movie in movies:
    if movies[movie] > 0:
        available.append(movies[movie])
print(available)
```

C

```
available = []
for movie in movies:
    if movie > 0:
        available.append(movie)
print(available)
```

D

```
available = []
for movie in movies:
    if movies[movie] > 0:
        available.append(movie)
print(available)
```

Which of the following correctly answers the question "Which movie has the most copies available"? For this exercise, assume there are never the same number of copies for two different movies.

A

```
most_copies = 0
most_copies_movie = ""
for movie in movies:
    if movies[movie] > most_copies:
        most_copies_movie = movie
        most_copies = movies[movie]
        break
print(most_copies_movie)
```

B

```
most_copies = 0
most_copies_movie = ""
for movie in movies:
    if movies[movie] > most_copies:
        most_copies_movie = movie
        most_copies = movies[movie]
print(most_copies_movie)
```

C

```
most_copies = 0
most_copies_movie = ""
for movie in movies:
    if movies[movie] < most_copies:
        most_copies_movie = movie
        most_copies = movies[movie]
print(most_copies_movie)
```

D

```
most_copies = 0
most_copies_movie = ""
for movie in movies:
    if movie > most_copies:
        most_copies_movie = movie
        most_copies = movies[movie]
print(most_copies_movie)
```

Fill in the blanks to answer the question "How many different movie titles (ie not copies) are available for renting?"

```
count = 0
for movie in  1           :
  if  2           > 0:
     3          
print(count)
```

Correct answers:

1 movies   2 movies[movie]   3 count += 1

Item 3

Questions Data structures

Given the following function for counting how many copies of a given movie title are available:

```
def movies_by_title(title):
    ...
```

Which of the following options **throws an error** when calling the function?

A

```
movies_by_title("Inception")
```

B

```
movies_by_title()
```

C

```
movies_by_title(title="The Matrix")
```

Suppose the store charges more depending on the number of available copies. The base daily price is 5EUR/day per rental and the penalties are:

- 50% more if there are only two copies available
- 100% more if there is only one copy available

Fill in the blanks to complete the definition of a function named `rental_price` that takes one argument, `title`, the title of the movie, and computes the daily rental price for the movie according to the number of available copies.

```
def rental_price( 1 ):
  price = 15
  available_copies = 2
  if 3 :
    return price * 1.5
  elif 4 :
    return price * 2
  else
    return price
```

Correct answers:

1 title 2 movies[title] 3 available_copies == 2 4 available_copies == 1

The store also applies a **20% penalty for rentals of over 3 days.**

Fill in the blanks to complete the definition of a function named total_rental_price that takes two arguments, daily_price , the daily price the customer was charged when renting the movie, and days , the number of days the movie was rented, and returns the total price for the rental, based on the initial rental price and discount applied (if any):

```
def total_rental_price( 1 ):
  price = 2 * days
  if 3 :
    return price * 1.2
  else:
    return price
```

Correct answers:

1 daily_price, days   2 daily_price   3 days > 3

Given the following list of tuples of daily rental price and the days the movie was rented (e.g. the first customer was charged 5EUR/day and returned the movie after 8 days):

```
rentals = [(5, 8), (7.5, 2), (10, 4)]
```

Which of the following correctly defines the function profit that takes one argument, rentals , a list of tuples as described above, and computes the profit of the store. This value is obtained by adding the total rental price, for the length of their rental. Use the total_rental_price function to aid in the computation.

○
```
def profit(rentals):
  for price, days in rentals:
    return total_rental_price(price, days)
```

○ ✓
```
def profit(rentals):
  total = 0
  for data in rentals:
    total += total_rental_price(data[0], data[1])
  return total
```

○
```
def profit(rentals):
  total = 0
  for movie in movies:
    total += total_rental_price(movie)
  return total
```

○
```
def profit(rentals):
  total = 0
  for data in rentals:
    total += total_rental_price(data[1], data[0])
  return total
```

Considering the following function

```
def filter_movies(copies=3):
  for movie in movies:
    if movies[movie] == copies:
      print(movie)
      return movie
    elif copies == 0:
      print(movie)
```

What is the output of the following code?

```
print(filter_movies())
```

A

```
The Lion King
The Lion King
```

B

```
The Lion King
```

C

Nothing

D

```
Inception
The Lion King
```

Item 4

Questions Data structures

Consider the following class to represent a movie with its title, release year, and genre:

```
class Movie:
  def __init__(self, title, release_year, genre):
    self.title = title
    self.release_year = release_year
    self.genre = genre
```

Implement the `MovieStore` class to represent a movie rental store as described earlier.

The class attributes are:

- `movies` : A dictionary with the store inventory, as described earlier.

- `catalog` : A list of objects of class `Movie` as described earlier.

Fill in the blanks to implement the class constructor. The constructor takes one argument, `catalog` , a movie titles used to initialize the `catalog` attribute. The `movies` attribute should be initialized by adding the titles in `catalog` all having 15 copies.

```
class MovieStore:
  def __init__( 1          ):
    2
    3
    for movie in 4          :
      5          = 15
```

Correct answers:

1 self, catalog

2 self.catalog = catalog

3 self.movies = {}

4 self.catalog

5 self.movies[movie]

Which of the following answers the question "Create a new movie store with movie Shrek"?

A
```
store = MovieStore(self, catalog=["Shrek"])
```

B
```
store = MovieStore(self, catalog="Shrek")
```

[✓ C]
```
store = MovieStore(["Shrek"])
```

D
```
store = MovieStore("Shrek")
```

Fill in the blanks to complete the definition of method `return_movie` in the `MovieStore` class, to record the return of a movie. This method takes one argument, `title` and increments the inventory for the movie title by 1.

```
class MovieStore:
  ...
  def return_movie( 1 ):
    available = 2
    3 = available + 1
```

Correct answers:

1 self, title 2 self.movies[title] 3 self.movies[title]

Which of the following correctly defines the method `is_movie_available` that takes one argument, `movie` (an object of the `Movie` class), and checks if it is available for rental?

A
```
def is_movie_available(movie):
  return self.movies[movie.title] > 0
```

[✓ B]
```
def is_movie_available(self, movie):
  return self.movies[movie.title] > 0
```

C
```
def is_movie_available(self, movie):
  return self.movies[movie] > 0
```

D
```
def is_movie_available(self, title):
  return self.movies[title] > 0
```

What is the output of the following code?

```
store = MovieStore(["Shrek"])
store.return_movie("Shrek")
print(store.movies)
```

A Error

B
```
{}
```

[✓ C]
```
{"Shrek": 16}
```

D Nothing