4. Dictionary

Dictionary

```python
def get_books_name_for_reader(books, readers, reader_name): # Exercise 2's Function
    leased_books = []
    for reader in readers:
        if reader['name'] == reader_name: # Found the dictionary of our reader
            for book_id in reader['borrowed']:
                for book in books:
                    if book_id == book['book_id']:
                        leased_books.append(book['title'])
    return leased_books


def most_read_book(books, readers): # Exercise 3's Function
    most_leased = set()
    highest_votes = 0
    for book in books:
        book['votes'] = 0
        for reader in readers:
            for book_id in reader['borrowed']:
                if book_id == book['book_id']:
                    book['votes'] += 1
        highest_votes = book['votes'] if book['votes'] > highest_votes else highest_votes
    for book in books:
        if book['votes'] == highest_votes:
            most_leased.add(book['title'])
    return most_leased


def readers_having_most_read_book(readers): # Exercise 6's Function
    books_by_id = []
    possessing_readers = set()
    highest_votes = 0
    for reader in readers: # Create a list with dicts containing book ID book's vote count
        for readers_book_id in reader['borrowed']:
            books_by_id.append(dict(book_id = readers_book_id, votes = 0))

    for book in books_by_id: # Count the votes and find the highest vote count
        for reader in readers:
            for readers_book_id in reader['borrowed']:
                if readers_book_id == book['book_id']:
                    book['votes'] += 1
        highest_votes = book['votes'] if book['votes'] > highest_votes else highest_votes

    for book in books_by_id: # Check which readers have the most-leased books
        if book['votes'] == highest_votes:
            for reader in readers:
                for readers_book_id in reader['borrowed']:
                    if readers_book_id == book['book_id']:
                        possessing_readers.add(reader['name'])

    return possessing_readers


if __name__ == '__main__':
    # Books List
    books = [dict(book_id=1001, title="Harry Potter", genre="fantasy", pages=500),
             dict(book_id=1002, title="A song of Ice and Fire", genre="fantasy", pages=700),
             dict(book_id=1003, title="1984", genre="classic", pages=800),
             dict(book_id=1004, title="Attack on Titan", genre="manga", pages=1400),
             dict(book_id=1005, title="One Piece", genre="manga", pages=12000)
    ]
    # Readers List
    readers = [{"name": "Ichi", "borrowed": [1001, 1003]},
               {"name": "Ni", "borrowed": [1002]},
               {"name": "San", "borrowed": [1005, 1002]},
               {"name": "Yon", "borrowed": [1005, 1002]},
               {"name": "Go", "borrowed": [1005]}
    ]
```