```
# Library Book Borrowing Management

# Sample dictionary storing borrow records
# Key = member name, Value = list of books borrowed
borrow_records = {
    "Alice": ["Book A", "Book B"],
    "Bob": ["Book A"],
    "Charlie": [],
    "David": ["Book A", "Book C", "Book B"],
    "Eve": [],
    "Frank": ["Book C", "Book C"],
}


# Function to calculate average number of books borrowed per member
def average_borrowed(records):
    total = 0  # Variable to hold the total number of borrowed books
    for books in records.values():  # Loop through all borrowed book lists
        total += len(books)  # Add number of books borrowed by each member
    return total / len(records)  # Return average = total books / total members


# Function to find book with highest and lowest borrow counts
def max_min_borrowed(records):
    book_count = {}  # Dictionary to store how many times each book is borrowed
    for books in records.values():  # Loop through each member's borrow list
        for book in books:  # Loop through books borrowed by that member
            if book in book_count:  # If book already counted
                book_count[book] += 1  # Increase its count
            else:
                book_count[book] = 1  # Start counting the book
```

```
    # Find book with highest borrow count
    max_book = max(book_count, key=book_count.get)
    # Find book with lowest borrow count
    min_book = min(book_count, key=book_count.get)
    return max_book, min_book  # Return both books


# Function to count how many members have not borrowed any books
def count_zero_borrowers(records):
    count = 0  # Initialize counter
    for books in records.values():  # Loop through borrow lists
        if len(books) == 0:  # If list is empty (no books borrowed)
            count += 1  # Increase the counter
    return count  # Return total count


# Function to find the most frequently borrowed book (mode)
def most_frequent_book(records):
    book_count = {}  # Dictionary to count each book
    for books in records.values():  # Loop through each borrow list
        for book in books:  # Loop through books in that list
            if book in book_count:  # If book is already being counted
                book_count[book] += 1  # Add 1 to count
            else:
                book_count[book] = 1  # Start counting this book
    return max(book_count, key=book_count.get)  # Return book with
max count


# --------- OUTPUT SECTION ---------
# Display the average number of books borrowed
print("Average books borrowed:", average_borrowed(borrow_records))

# Get the most and least borrowed books
max_b, min_b = max_min_borrowed(borrow_records)
```

```
print("Most borrowed book:", max_b)  # Print the most borrowed book
print("Least borrowed book:", min_b)  # Print the least borrowed book

# Display how many members borrowed 0 books
print("Members who borrowed 0 books:",
count_zero_borrowers(borrow_records))

# Display the book that was borrowed most frequently
print("Most frequently borrowed book:",
most_frequent_book(borrow_records))
```

*Output:

Average books borrowed: 1.3333333333333333

Most borrowed book: Book A

Least borrowed book: Book B

Members who borrowed 0 books: 2

Most frequently borrowed book: Book A *\