

TASK 1 - Develop a Python task management app for adding, removing, listing, prioritizing, and receiving task recommendations based on task descriptions.

```
class Task:
```

```
    def __init__(self, description, priority=0):
```

```
        self.description = description
```

```
        self.priority = priority
```

```
    def __repr__(self):
```

```
        return f"Task(description='{self.description}', priority={self.priority})"
```

```
class TaskManager:
```

```
    def __init__(self):
```

```
        self.tasks = []
```

```
    def add_task(self, description, priority=0):
```

```
        new_task = Task(description, priority)
```

```
        self.tasks.append(new_task)
```

```
        self.tasks.sort(key=lambda x: x.priority, reverse=True)
```

```
    def remove_task(self, description):
```

```
        self.tasks = [task for task in self.tasks if task.description != description]
```

```
    def list_tasks(self):
```

```
        if not self.tasks:
```

```
            print("No tasks available.")
```

```
        else:
```

```
            for index, task in enumerate(self.tasks, start=1):
```

```
                print(f"{index}. {task}")
```

```
    def prioritize_task(self, description, new_priority):
```

```
for task in self.tasks:
    if task.description == description:
        task.priority = new_priority
        self.tasks.sort(key=lambda x: x.priority, reverse=True)
        break
else:
    print(f"Task '{description}' not found.")
```

```
def recommend_task(self, query):
    matched_tasks = []
    for task in self.tasks:
        if query.lower() in task.description.lower():
            matched_tasks.append(task)
    if matched_tasks:
        matched_tasks.sort(key=lambda x: x.priority, reverse=True)
        return matched_tasks[0]
    else:
        return None
```

# Example usage:

```
task_manager = TaskManager()
```

```
task_manager.add_task("Write report for meeting")
```

```
task_manager.add_task("Prepare presentation slides", priority=2)
```

```
task_manager.add_task("Review project proposal")
```

```
task_manager.list_tasks()
```

```
task_manager.prioritize_task("Review project proposal", 1)
```

```
print("\nAfter prioritization:")
```

```
task_manager.list_tasks()
```

```
recommended_task = task_manager.recommend_task("report")
```

```
if recommended_task:
```

```
    print(f"\nRecommended task based on description: {recommended_task}")
```

```
else:
```

```
    print("No matching task found.")
```