In [1]:
```python
# Define a class representing a to-do list
class TodoList:
    # Constructor to initialize an empty list for tasks
    def __init__(self):
        self.tasks = []

    # Method to display the tasks in the to-do list
    def display_tasks(self):
        # Check if the to-do list is empty
        if not self.tasks:
            print("Your to-do list is empty.")
        else:
            print("To-Do List:")
            # Iterate over each task and display its index, description, and completion status
            for i, task in enumerate(self.tasks, start=1):
                status = "Done" if task["completed"] else "Not Done"
                print(f"{i}. {task['task']} ({status})")

    # Method to add a task to the to-do list
    def add_task(self, task_name):
        # Create a dictionary representing the task with the provided name and mark it as not completed
        task = {"task": task_name, "completed": False}
        # Add the task to the list
        self.tasks.append(task)
        # Print a message confirming the addition of the task
        print(f"Task '{task_name}' added to your to-do list.")

    # Method to mark a task as completed
    def mark_completed(self, task_number):
        # Check if the provided task number is within the valid range
        if 1 <= task_number <= len(self.tasks):
            # Mark the task at the specified index as completed
            self.tasks[task_number - 1]["completed"] = True
            # Print a message confirming the task completion
            print(f"Task {task_number} marked as completed.")
        else:
            # Print an error message for an invalid task number
            print("Invalid task number. Please enter a valid task number.")

    # Method to remove a task from the to-do list
    def remove_task(self, task_number):
        # Check if the provided task number is within the valid range
        if 1 <= task_number <= len(self.tasks):
            # Remove the task at the specified index and store it in a variable
            removed_task = self.tasks.pop(task_number - 1)
            # Print a message confirming the removal of the task
            print(f"Task '{removed_task['task']}' removed from your to-do list.")
        else:
            # Print an error message for an invalid task number
            print("Invalid task number. Please enter a valid task number.")

# Main function to run the program
def main():
    # Create an instance of the TodoList class
    todo_list = TodoList()

    # Main program loop
    while True:
        # Display the menu options
        print("\nOptions:")
        print("1. Display to-do list")
        print("2. Add a task")
        print("3. Mark a task as completed")
        print("4. Remove a task")
        print("5. Quit")
        # Prompt the user for their choice
        choice = input("Enter your choice: ")

        # Check the user's choice and perform the corresponding action
        if choice == '1':
            todo_list.display_tasks()
        elif choice == '2':
            task_name = input("Enter the task: ")
            todo_list.add_task(task_name)
        elif choice == '3':
            todo_list.display_tasks()
            task_number = int(input("Enter the task number to mark as completed: "))
            todo_list.mark_completed(task_number)
        elif choice == '4':
            todo_list.display_tasks()
            task_number = int(input("Enter the task number to remove: "))
            todo_list.remove_task(task_number)
        elif choice == '5':
            break
        else:
            print("Invalid choice. Please enter a valid option.")

# Check if the script is being run directly
if __name__ == "__main__":
    # Call the main function to start the program
    main()
```

```
Options:
1. Display to-do list
2. Add a task
3. Mark a task as completed
4. Remove a task
5. Quit
Your to-do list is empty.

Options:
1. Display to-do list
2. Add a task
3. Mark a task as completed
4. Remove a task
5. Quit
Task 'eating' added to your to-do list.

Options:
1. Display to-do list
2. Add a task
3. Mark a task as completed
4. Remove a task
5. Quit
To-Do List:
1. eating (Not Done)

Options:
1. Display to-do list
2. Add a task
3. Mark a task as completed
4. Remove a task
5. Quit
To-Do List:
1. eating (Not Done)
Task 1 marked as completed.

Options:
1. Display to-do list
2. Add a task
3. Mark a task as completed
4. Remove a task
5. Quit
To-Do List:
1. eating (Done)

Options:
1. Display to-do list
2. Add a task
3. Mark a task as completed
4. Remove a task
5. Quit
To-Do List:
1. eating (Done)
Task 'eating' removed from your to-do list.

Options:
1. Display to-do list
2. Add a task
3. Mark a task as completed
4. Remove a task
5. Quit
Your to-do list is empty.

Options:
1. Display to-do list
2. Add a task
3. Mark a task as completed
```

4. Remove a task
5. Quit

In [ ]: