const readline = require("readline");

const rl = readline.createInterface({

  input: process.stdin,

  output: process.stdout,

});

let list = [];

function addTask(string) {

  list.push(string);

  console.log(`"${string}" has been added!`);

}

function removeTask(num) {

  if (num >= 0 && num < list.length) {

    console.log(`"${list[num]}" was removed`);

    list.splice(num, 1);

  } else {

    console.log("Invalid task number!");

  }

}

function displayTasks() {

  if (list.length === 0) {

    console.log("Your to-do list is empty.");

  } else {

    console.log("Your Tasks:");

    list.forEach((task, index) => {

      console.log(`${index + 1}. ${task}`);

    });

  }

}

function clearTasks() {

  list = [];

  console.log("All tasks have been cleared!");

}

function exit() {

  console.log("Goodbye!");

  rl.close();

}

function showMenu() {

  console.log("\nTO-DO-LIST MANAGER");

  console.log("Commands:");

  console.log("1. Add a Task");

  console.log("2. Remove a Task");

  console.log("3. Display List");

  console.log("4. Clear all tasks");

  console.log("5. Exit");

}

function run() {

  showMenu();

  rl.question("\nEnter Command: ", (inp) => {

    switch (inp.trim()) {

      case "1":

        rl.question("Enter Task: ", (task) => {

          addTask(task);

          run();

        });

        break;

      case "2":

        rl.question("Enter Task Index to Remove (1-based): ", (num) => {

          const index = parseInt(num) - 1;

          removeTask(index);

          run();

        });

        break;

      case "3":

        displayTasks();

        run();

        break;

      case "4":

        clearTasks();

        run();

        break;

      case "5":

        exit();

        break;

      default:

        console.log("Invalid command! Please try again.");

        run();

    }

  });

}

run();

