# List Processing – Teamwork Project Assignment

This document describes the teamwork assignment for the “**List Processing**” console-based application. The goal of this teamwork exercise is to gain **team collaboration skills** using tools like **GitHub** and **Trello**.

# List Processing

Design and implement a **console-based application for list processing**. It should **enter a list of strings** from the console and **continuously execute commands** (like add / remove / replace, invert) over the list.

## General Requirements

* You will be assigned a **team of 3-4 students**.
* Use **GitHub** as source code repository. The code is shared between all team members.
* Use **Trello** for task management. Create tasks for each piece of work in your team.

## Commands Description

The **first input line** holds the **initial list of strings**:

* space separated list of items

**Print the list** immediately after you read it (space separated).

The next lines hold one of the following **commands** (until the commands “end” is found):

* append <string> – adds the specified string at the end of the list and prints the list (space separated).
* prepend <string> – inserts the specified string at the start of the list and prints the list.
* reverse – reorders the list items in reversed order (from its end to its start) and prints the list.
* insert <index> <string> – inserts the specified string at the specified position in the list and prints the list after that.
  + Positions are indexed from 0 to list\_size - 1.
  + In case of invalid index, print “*Error: invalid index <index>*”.
* delete <index> – deletes the string at the specified position in the list and prints the list.
  + Positions are indexed from 0 to list\_size - 1.
  + In case of invalid index, print “*Error: invalid index <index>*”.
* roll left – rolls the list on the left and prints the list. The first list item comes last.
* roll right – rolls the list on the right and prints the list. The last list item comes first.
* sort – sorts the list in alphabetical order and prints the list.
* count <string> – prints how many times the specified string occurs in the list (case sensitive). The command prints a single integer number as output.
* end – stops the execution of commands. Print “*Finished*” as command output.

Each command produces one of the following outputs:

* In case of success the command prints its expected output as described above.
* In case of error (e.g. invalid index or command parameter) the commands prints “*Error: <error message>*”.
  + In case of invalid command name print “*Error: invalid command*”. Command names are case-sensitive.
  + In case of invalid number of parameters or incorrect format of the parameters print “*Error: invalid command parameters*”.

## Examples

The examples below show the commands and their output:

|  |  |
| --- | --- |
| **Input** | **Output** |
| alpha beta gamma  append delta  prepend 00  invalid command  insert 1 ins  insert -1 ins  insert 1  delete 3  delete invalid  delete 5  roll left  roll  roll right  just roll  reverse  sort  sort 3  count alpha  append alpha  count alpha  count ALPHA  end 1  END  end | alpha beta gamma  alpha beta gamma delta  00 alpha beta gamma delta  Error: invalid command  00 ins alpha beta gamma delta  Error: invalid index -1  Error: invalid command parameters  00 ins alpha gamma delta  Error: invalid command parameters  Error: invalid index 5  ins alpha gamma delta 00  Error: invalid command parameters  00 ins alpha gamma delta  Error: invalid command  delta gamma alpha ins 00  00 alpha gamma delta ins  Error: invalid command parameters  1  00 alpha gamma delta ins alpha  2  0  Error: invalid command parameters  Error: invalid command  Finished |

## Tasks in Trello

Split the work into **tasks in a Trello project board**:

* Create 4 lists: **Backlog** | **In Progress** | **For Review** | **Done**.
  + All tasks initially start from the “**Backlog**” list. Assign project members for the first few tasks.
  + When the work on certain task starts, move it to “**In Progress**”. The task stays in this list during the time of working on it.
  + After a task is **completed**, put your changes in GitHub (with merge if required) and move the task to “**For Review**”. The task is still not done, because it requires someone to review it.
  + Assign team member to **review each completed task**. This is an important part of the process. The **concept of reviewing** for this project is as follows:
    - A team member **implements some functionality** and moves its task to “**For Review**”.
    - Other team member **verifies that the functionality** is correctly implemented and moves it to “**Done**” or fixes the bugs / problems and then **assigns another team member to review it**.
  + All tasks should flow during all the states: **Backlog** 🡪 **In Progress** 🡪 **For Review** 🡪 **Done**.
* Initially **split the project into tasks** and **create the tasks in Trello**. Typical tasks to consider:
  + Create the **GitHub project** and invite all team members.
  + Design the **project structure** (files, classes, interfaces, command parsing logic, others).
    - The project structure should allow **adding new commands** easily.
    - The project structure should allow **several team members to work in parallel** on the shared source code. Think about **how to merge conflicting changes**!
    - Design and implement the **command parsing and processing** infrastructure.
  + Implement the processing of the **first command line** (entering the input list)
  + Implement the “append” command.
  + Implement the “prepend” command.
  + Implement the “reverse” command.
  + Implement the “insert” command.
  + Implement the “delete” command.
  + Implement the “roll left” command.
  + Implement the “roll right” command.
  + Implement the “sort” command.
  + Implement the “count” command.
  + Implement the “end” command.
* **Process all Trello tasks** for the project until the project is finished.
  + Assign project members (one or several) for each task.
  + Use the task flow: **Backlog** 🡪 **In Progress** 🡪 **For Review** 🡪 **Done**.

## Submitting Your Work

You should submit until **21-Nov-2017** a **link to your GitHub** **repository** at the course site under the topic “Task Management Tools”: <https://softuni.bg/trainings/1799/practical-teamwork-sept-2017#lesson-7104>.

* In the README.md file put a link to your Trello project.
* Make sure your Trello project is public (for view only).

**Each team member submits the same link!**