This project involves building a Process Scheduler using a Linked List.

A Linked List is a data structure where each node contains a data field and a reference(link) to the next node in the list.

Each Node of this Linked List is a structure of type Procecss

```
struct Process {
  Process *previousProcess;
  Process *nextProcess;
  ProcessId id;
}
```

(ProcessId is of the type int)

1. `add_process(ProcessId)` : adds process (with given ProcessId) to the back of the Linked List.
2. `delete_process(ProcessId)` : deletes process with given ProcessId
3. `fork(ProcessId pid, ProcessId newId)` : Creates new process with id (newId) and adds it to the linked list right after the process with ProcessId pid (right after means process pid's nextProcess pointer must point to process newId)
4. `print_schedule()` : prints all the process id in order

Resources to better understand linked lists: 1) https://www.log2base2.com/data-structures/linked-list/linked-list-in-c.html 2) https://www.youtube.com/watch?v=VOpjAHCee7c

Input Format

The first line of the input is an integer n, the number of operations

The next lines represent the operation id and the corresponding process id(s) if necessary

0 represents the add_Process function followed by the processId to be added(int)

1 represents the delete_Process function followed by the processId to be deleted(int)

2 represents the fork function followed by 2 processIds to add the second porcessid after the first argument(int)

3 represents the print_schedule function to print the processes(no argument following it)

Constraints

Number Of Operations < 1000

Output Format

Whenever print_schedule is called, process ids of all nodes (in order, starting from head) should be printed with a “ ” in between.