GIT Manager

We are about to create a GIT Manager on console using Python 3. We can’t use any database or UI framework. For Managing GIT we need to follow our custom command. Our Commands is listed below -

1. git init

--- This command will be the start of GIT Manager. Without this command, we can’t execute other command.

2. git commit “commit message”

--- By this command, we will store this commit message with a auto generated sequence number. The sequence number must be integer and start with 1.

3. git show commit

--- By this command, we can show the current commit message.

4. git show all commit

--- By this command, we can view all commit commit message with sequence number. Need to show star on current commit. Like -

*1 message-one
2 message-two
3 message-three

5. git delete commit_number

--- By this command, we can delete specified commit with mentioned commit_number. If commit number is current commit then current commit pointer will move backward.

6. git jump commit_number

--- By this command, we can jump to an specific **commit_number**. Then Commit sequence will change as the specified commit-number will be current commit. For example – we have three commit message as -

**1 message-one*
2 message-two
3 message-three

(here * means current commit). Now we execute “**git jump 3**”. then current commit will be 3 and sequence will change. Output will be -

**3 message-three*
1 message-one
2 message-two

7. git move back

--- By this command, Our current commit pointer will be one step backward. And Sequence will remain same, only pointer will move. For example – we have three commit message as -

**1 message-one*
2 message-two
3 message-three

(here * means current commit). Now we execute **git move back**. then current commit will be 2 . Output will be -

1 message-one
**2 message-two*
3 message-three

(special note: on any point when when we commit new message using second command, then new message will be current commit pointer and anything that was forward will be deleted. For example – we have three commit message as -
1 message-one
**2 message-two*
3 message-three
(here * means current commit). Now we execute **git commit "new message"**. then current commit will be 4 and sequence will change. Output will be -
**4 new message*
2 message-two
3 message-three)

8. git update "New Message"
--- By this command, we can update current commit message

9. exit
--- this the the termination command.

INPUT – is a line of string as command
OUTPUT – Only with command 3 & 4 we can see the output.

Sample Input	Sample Output
git init **git commit "commit message one"** **git commit "commit message two"** **git commit "commit message three"** **git show commit** **git show all commit** **git jump 2** **git show all commit** **git move back** **git update "New Message"** **git commit "commit message four"** **git show all commit** **exit**	**commit message three** ***3 commit message three** **2 commit message two** **1 commit message one** ***2 commit message two** **3 commit message three** **1 commit message one** ***4 commit message four** **3 New message** **1 commit message one**