**Dell@Dell-PC MINGW64 ~/Desktop/WhatsApp**

**$ git init**

Initialized empty Git repository in C:/Users/Dell/Desktop/WhatsApp/.git/

**Dell@Dell-PC MINGW64 ~/Desktop/WhatsApp (master)**

**$ touch f1.c**

**$ git add .**

**$ git commit -m " First File f1.c Added"**

[master (root-commit) 0ad8862] First File f1.c Added

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 f1.c

**$ git status**

On branch master

nothing to commit, working tree clean

**$ git log -2**

commit 0ad8862c1e5ee7a7403b75a441a3081ed4650663 (HEAD -> master)

Author: patil <madupatil1606@gmail.com>

Date: Mon Jun 27 08:16:46 2022 +0530

First File f1.c Added

**BRANCH 1**

**Dell@Dell-PC MINGW64 ~/Desktop/WhatsApp (master)**

**$ git branch Branch1**

**$ git checkout Branch1**

Switched to branch 'Branch1'

**Dell@Dell-PC MINGW64 ~/Desktop/WhatsApp (Branch1)**

**$ ls**

f1.c

**$ git tag RELEASE\_1**

**$ git tag**

RELEASE\_1

**$ touch a1.c**

**$ git add \***

**$ git commit -m "Added Second File a1.c in Branch1"**

[Branch1 e04f88a] Added Second File a1.c in Branch1

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 a1.c

**$ git status**

On branch Branch1

nothing to commit, working tree clean

**$ git log -3**

commit e04f88a996a03dac751b633e3e2d1b6f24f9ec55 (HEAD -> Branch1)

Author: patil <madupatil1606@gmail.com>

Date: Mon Jun 27 08:18:56 2022 +0530

Added Second File a1.c in Branch1

commit 0ad8862c1e5ee7a7403b75a441a3081ed4650663 (tag: RELEASE\_1, master)

Author: patil <madupatil1606@gmail.com>

Date: Mon Jun 27 08:16:46 2022 +0530

First File f1.c Added

**$ ls**

a1.c f1.c

**Master Branch**

**Dell@Dell-PC MINGW64 ~/Desktop/WhatsApp (Branch1)**

**$ git checkout master**

Switched to branch 'master'

**Dell@Dell-PC MINGW64 ~/Desktop/WhatsApp (master)**

**$ git branch**

Branch1

\* master

**$ touch f2.c**

**$ git add .**

**$ git commit -m " Added f2.c file in Master Branch"**

[master abc2e6d] Added f2.c file in Master Branch

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 f2.c

**$ git status**

On branch master

nothing to commit, working tree clean

**$ ls**

f1.c f2.c

**$ git log -3**

commit abc2e6db5fffbd2eb000fb4b6fe43a72317ed5c5 (HEAD -> master)

Author: patil <madupatil1606@gmail.com>

Date: Mon Jun 27 08:21:20 2022 +0530

Added f2.c file in Master Branch

commit 0ad8862c1e5ee7a7403b75a441a3081ed4650663 (tag: RELEASE\_1)

Author: patil <madupatil1606@gmail.com>

Date: Mon Jun 27 08:16:46 2022 +0530

First File f1.c Added

**$ git tag RELEASE\_MASTER**

**$ git tag**

RELEASE\_1

RELEASE\_MASTER

**Created Branch2**

**Dell@Dell-PC MINGW64 ~/Desktop/WhatsApp (master)**

**$ git checkout -b Branch2**

Switched to a new branch 'Branch2'

**Dell@Dell-PC MINGW64 ~/Desktop/WhatsApp (Branch2)**

**$ git branch**

Branch1

\* Branch2

master

**$ touch a2.c**

**$ git add .**

**$ git commit -m "Added a2.c file in Branch2"**

[Branch2 f93bcc3] Added a2.c file in Branch2

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 a2.c

**$ git status**

On branch Branch2

nothing to commit, working tree clean

**$ git log -3**

commit f93bcc3a35e4724d85c976379c5c4f8713bf5c59 (HEAD -> Branch2)

Author: patil <madupatil1606@gmail.com>

Date: Mon Jun 27 08:24:45 2022 +0530

Added a2.c file in Branch2

commit abc2e6db5fffbd2eb000fb4b6fe43a72317ed5c5 (tag: RELEASE\_MASTER, master)

Author: patil <madupatil1606@gmail.com>

Date: Mon Jun 27 08:21:20 2022 +0530

Added f2.c file in Master Branch

commit 0ad8862c1e5ee7a7403b75a441a3081ed4650663 (tag: RELEASE\_1)

Author: patil <madupatil1606@gmail.com>

Date: Mon Jun 27 08:16:46 2022 +0530

First File f1.c Added

**Merge Branch1 with Branch2**

**Dell@Dell-PC MINGW64 ~/Desktop/WhatsApp (Branch2)**

**$ git merge Branch1**

Merge made by the 'recursive' strategy.

a1.c | 0

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 a1.c

**$ ls**

a1.c a2.c f1.c f2.c

**$ touch a3.c**

**$ git add .**

**$ git commit -m "Added a3.c file in Branch2"**

[Branch2 2c32a72] Added a3.c file in Branch2

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 a3.c

**$ git status**

On branch Branch2

nothing to commit, working tree clean

**$ git log -3**

commit 2c32a72652972e99a1961b36108f39517a02e72f (HEAD -> Branch2)

Author: patil <madupatil1606@gmail.com>

Date: Mon Jun 27 08:27:14 2022 +0530

Added a3.c file in Branch2

commit d0bed4a7e621fad27fd0b03cb3f51a183f6a68cb

Merge: f93bcc3 e04f88a

Author: patil <madupatil1606@gmail.com>

Date: Mon Jun 27 08:26:09 2022 +0530

Merge branch 'Branch1' into Branch2

commit f93bcc3a35e4724d85c976379c5c4f8713bf5c59

Author: patil <madupatil1606@gmail.com>

Date: Mon Jun 27 08:24:45 2022 +0530

Added a2.c file in Branch2

**$ ls**

a1.c a2.c a3.c f1.c f2.c

**Switch To Master Branch**

**Create f3, f4, f5 Files**

**Merge with Branch2**

**Dell@Dell-PC MINGW64 ~/Desktop/WhatsApp (master)**

**$ ls**

f1.c f2.c

**$ touch f3.c**

**$ git add .**

**$ git commit -m "Added f3.c File in Master"**

[master d47e999] Added f3.c File in Master

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 f3.c

**$ ls**

f1.c f2.c f3.c

**$ touch f4.c**

**$ git add \***

**$ git commit -m "Added f4.c in Master Branch"**

[master 90825de] Added f4.c in Master Branch

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 f4.c

**$ ls**

f1.c f2.c f3.c f4.c

**$ touch f5.c**

**$ git add .**

**$ git commit -m "Added f5.c in Master Branch"**

[master a26e867] Added f5.c in Master Branch

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 f5.c

**$ ls**

f1.c f2.c f3.c f4.c f5.c

**$ git status**

On branch master

nothing to commit, working tree clean

**$ git log --oneline**

a26e867 (HEAD -> master) Added f5.c in Master Branch

90825de Added f4.c in Master Branch

d47e999 Added f3.c File in Master

abc2e6d (tag: RELEASE\_MASTER) Added f2.c file in Master Branch

0ad8862 (tag: RELEASE\_1) First File f1.c Added

**Merge with Branch2**

**Dell@Dell-PC MINGW64 ~/Desktop/WhatsApp (master)**

**$ git merge Branch2**

Merge made by the 'recursive' strategy.

a1.c | 0

a2.c | 0

a3.c | 0

3 files changed, 0 insertions(+), 0 deletions(-)

create mode 100644 a1.c

create mode 100644 a2.c

create mode 100644 a3.c

**$ ls**

a1.c a2.c a3.c f1.c f2.c f3.c f4.c f5.c

**$ git tag RELEASE\_MASTER2**

**$ git tag**

RELEASE\_1

RELEASE\_MASTER

RELEASE\_MASTER2

**GIT SQUASH**

Dell@Dell-PC MINGW64 ~/Desktop/Squash

**$ pwd**

/c/Users/Dell/Desktop/Squash

Dell@Dell-PC MINGW64 ~/Desktop/Squash

Dell@Dell-PC MINGW64 ~/Desktop/Squash (master)

**$ touch f1.c --> f1.c File Created**

**$ git add .**

**$ git commit -m "Commit 1"**

Dell@Dell-PC MINGW64 ~/Desktop/Squash (master)

**$ touch f2.c --> f2.c File Created**

**$ git add .**

**$ git commit -m "Commit 2"**

**$ git log --oneline**

c8b0e59 (HEAD -> master) Commit 2

95c3155 Commit 1

**Create & Switch another Branch Called Testing**

Dell@Dell-PC MINGW64 ~/Desktop/Squash (master)

**$ git branch**

Development

\* master

**$ git checkout -b Testing**

Switched to a new branch 'Testing'

**$ git branch**

Development

\* Testing

master

Dell@Dell-PC MINGW64 ~/Desktop/Squash (Testing)

**$ touch x1.c**

**$ git add .**

**$ git commit -m "Committed x1 File"**

[Testing 94c264e] Committed x1 File

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 x1.c

Dell@Dell-PC MINGW64 ~/Desktop/Squash (Testing)

**$ touch y1.c**

**$ git add .**

**$ git commit -m "Committed y1 File"**

[Testing 42c9230] Committed y1 File

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 y1.c

Dell@Dell-PC MINGW64 ~/Desktop/Squash (Testing)

**$ touch z1.c**

**$ git add .**

**$ git commit -m "Committed z1 File"**

[Testing 7a24ed9] Committed z1 File

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 z1.c

**$ git log --oneline**

7a24ed9 (HEAD -> **Testing**) Committed z1 File

42c9230 Committed y1 File

94c264e Committed x1 File

c8b0e59 (**master**) Commit 2

95c3155 Commit 1

**Now Squash the Commits in Testing Branch**

**REWRITING GIT HISTORY**

**1. amending Commits:**

**Add untracked Files with previous commit**

Suppose you have committed 3 commits but some files are in Working Area. Now you want to add those files in Last Commit then we are using **Amending** Option.

* Assume you are in **my\_bracnh.** You have 3 commits.
* Now you have 2 untracked files OR Files are in working Area.
* You dont want to create New Commit but you want to Add these 3 files with Latest Commit.

**Dell@Dell-PC MINGW64 ~/Desktop/rewrite (my\_branch)**

**$ git log --oneline**

f99e8bf (HEAD -> my\_branch) **Added A3 File**

2c6af43 **Added A2 File**

22df98a **Added A1 File**

5f0f13e (master) F1.c Added

**$ git status**

On branch my\_branch

Untracked files:

(use "git add <file>..." to include in what will be committed)

B1.java

B2.java

Now B1 and B2 files you want to Combine with latest Commit

First you need to Add Files into Staging Area

**$ git add .**

Dell@Dell-PC MINGW64 ~/Desktop/rewrite (my\_branch)

$ git status

On branch my\_branch

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

new file: B1.java

new file: B2.java

Now Added into Staging Area

**$ git commit --amend --no-edit**

[my\_branch 4af3557] Added A3 File

Date: Mon Jul 4 14:19:37 2022 +0530

3 files changed, 0 insertions(+), 0 deletions(-)

create mode 100644 A3.sh

create mode 100644 B1.java

create mode 100644 B2.java

**$ git log --oneline**

4af3557 (HEAD -> my\_branch) Added A3 File

2c6af43 Added A2 File

22df98a Added A1 File

5f0f13e (master) F1.c Added

**$ git show 4af3557**

Author: patil <madupatil1606@gmail.com>

Date: Mon Jul 4 14:19:37 2022 +0530

**Added A3 File**

**diff --git a/A3.sh b/A3.sh**

**new file mode 100644**

**index 0000000..e69de29**

**diff --git a/B1.java b/B1.java**

**new file mode 100644**

**index 0000000..e69de29**

**diff --git a/B2.java b/B2.java**

**new file mode 100644**

**index 0000000..e69de29**

**BEFORE Amending Commits ID Numbers**

**f99e8bf** (HEAD -> my\_branch) Added A3 File

2c6af43 Added A2 File

22df98a Added A1 File

5f0f13e (master) F1.c Added

**AFTER Amending Commits ID Numbers**

**4af3557** (HEAD -> my\_branch) Added A3 File

2c6af43 Added A2 File

22df98a Added A1 File

5f0f13e (master) F1.c Added

So After Amending the Commit id will be changed.

**f99e8bf --> 4af3557**

**2. Rewording Commits Message**

**Correct the Spelling Mistake in Commits**

Rewording Commit Option is user to Correct in your Commit Message by using **Rebase & Reword**

Ex: Frotpag bug fiz

This commit have Spell mistake now you can correct these mistake using

**$ git log --oneline**

6f3515a (HEAD -> my\_branch) **Frotpag bug fiz**

4af3557 Added A3 File

2c6af43 Added A2 File

22df98a Added A1 File

5f0f13e (master) F1.c Added

**$ git rebase -i HEAD~2**

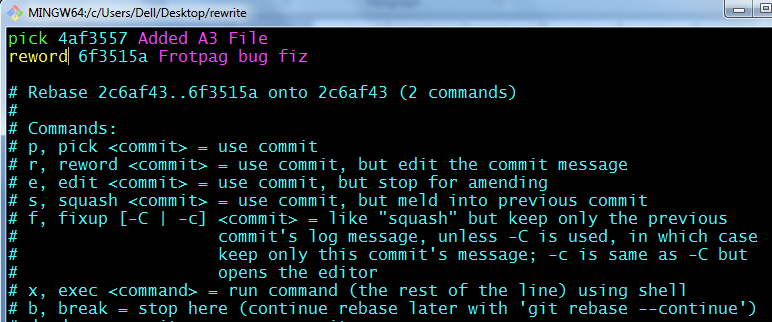
It will Open 1 Vi Windows



Now Press i

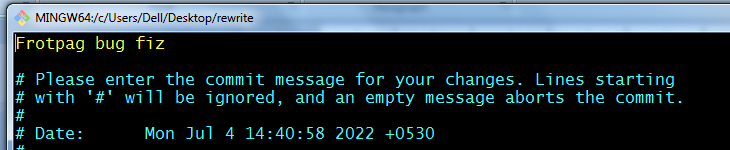
Then go to which commit ID Message you want edit.

Replace **pick** with **reword**

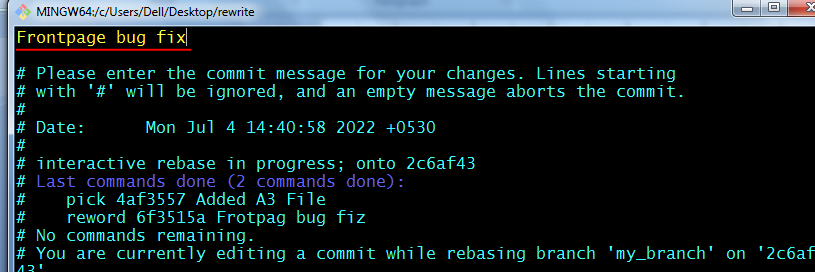
****

Save: ESC:wq

Now 1 More Window will Opens Here you Correct your Commit Message



Now Press i it will come Insert Mode

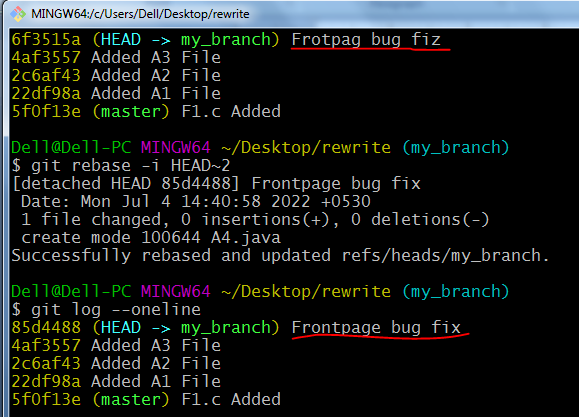


Once Completed

Save : ESC:wq

Now you can check your Commit message has been corrected

**$ git log --oneline**



**3. Deleting Commits**

**Delete unwanted Commits**

Suppose if you want to delete the Commit you can use **Drop** Option with **Rebase** Command

**$ git log --oneline**

85d4488 (HEAD -> my\_branch) Frontpage bug fix

4af3557 Added A3 File

**2c6af43 Added A2 File**

22df98a Added A1 File

5f0f13e (master) F1.c Added

You want to Delete This Commit **2c6af43 Added A2 File**

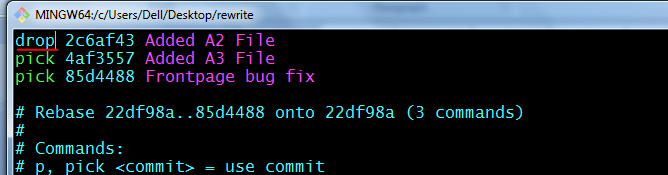
$ git rebase -i HEAD~3

It will open 1 Window

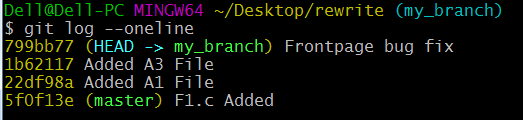
Press i it will go Insert Mode

Replace **pick** with **Drop** Option

Save- ESC:wq



You can see **2c6af43 Added A2 File** Commit has been deleted



**4. Reordering Commits**

**Reorder OR Rearrange the Commits**

If you want Reorder OR Swap the Commit with One Another we are using Rebase with order option

$ git log --oneline

**799bb77 (HEAD -> my\_branch) Frontpage bug fix**

1b62117 Added A3 File

**22df98a Added A1 File**

5f0f13e (master) F1.c Added

Suppose you want swap these 2 commits like

**Previous**

**799bb77 (HEAD -> my\_branch) Frontpage bug fix**

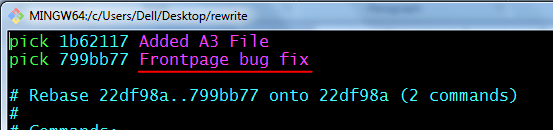
**22df98a Added A1 File**

**After Swap**

**22df98a Added A1 File**

**799bb77 (HEAD -> my\_branch) Frontpage bug fix**

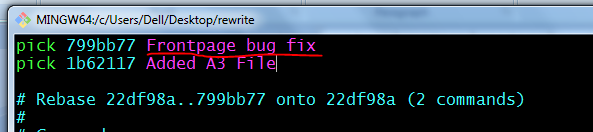
$ git rebase -i HEAD~2



Now Press i it will go Insert Mode

Then Copy **799bb77 Frontpage bug fix** this commit on Top.

Then Remove the below **799bb77 Frontpage bug fix** Commit



Now Your Commits has been Reorder Commit ID has been changed

**$ git log --oneline**

3247eb3 (HEAD -> my\_branch) Added A3 File

**3686077 Frontpage bug fix**

22df98a Added A1 File

5f0f13e (master) F1.c Added

**Previous**

**799bb77 (HEAD -> my\_branch) Frontpage bug fix**

1b62117 Added A3 File

22df98a Added A1 File

5f0f13e (master) F1.c Added

**After Reorder**

3247eb3 (HEAD -> my\_branch) Added A3 File

**3686077 Frontpage bug fix**

22df98a Added A1 File

5f0f13e (master) F1.c Added

**5. Fixup Commits**

**Merge Bottom Commits with Top level Commits**

If you want to Merge Or Fix Up Below Commits with Above Commit will use Rebase with Fixup Option

**$ git log --oneline**

3247eb3 (HEAD -> my\_branch) Added A3 File

**3686077 Frontpage bug fix**

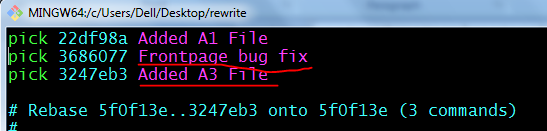
**22df98a Added A1 File**

5f0f13e (master) F1.c Added

You want to fix up these Commits **3686077 Frontpage bug fix**

and **22df98a Added A1 File** with 3247eb3 Added A3 File

$ git rebase -i HEAD~3



**$ git log --oneline**

**3f3df2a (HEAD -> my\_branch) Added A1 File**

5f0f13e (master) F1.c Added

You can See Both Commits are Merged Or Fixed up with Above Commit

**6. Splitting Commits**

**Split OR Divide Single Commit into Multiple Commits**

Used to Split 1 Commit into 2 Commits.

Ex: Added Files and Bug Fixed

**$ git log --oneline**

**913cad5 (HEAD -> my\_branch) Added Files and Bug Fixed**

3f3df2a Added A1 File

5f0f13e (master) F1.c Added

Here we want to Split commit **dbf7815 (HEAD -> my\_branch) Added Files and Bug Fixed** into 2 Commits. Like Below

**913cad5 (HEAD -> my\_branch) Added Files and Bug Fixed**

In this Commit we have 3 files F5.c, F6.c and F7.java

**Added Files --> 1** Commit

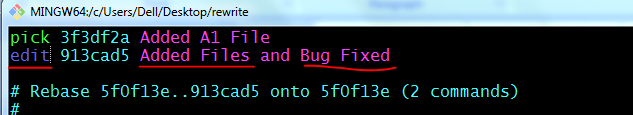
**Bug Fixed --> 2** Commit

$ git rebase -i HEAD~2

It will Open window

Press i it will go Insert Mode

Replace pick with edit



Save - ESC:wq

It will Open another branch

**$ git rebase -i HEAD~2**

Stopped at 913cad5... Added Files and Bug Fixed

You can amend the commit now, with

git commit --amend

Once you are satisfied with your changes, run

git rebase --continue

**Dell-PCMIN64~/Desktop/rewrite(my\_branch|REBASE 2/2)**

Now check the Status

**$ git status**

interactive rebase in progress; onto 5f0f13e

Last commands done (2 commands done):

pick 3f3df2a Added A1 File

edit 913cad5 Added Files and Bug Fixed

No commands remaining.

You are currently editing a commit while rebasing branch 'my\_branch' on '5f0f13e'.

(use "git commit --amend" to amend the current commit)

(use "git rebase --continue" once you are satisfied with your changes)

nothing to commit, working tree clean

**Dell-PCMIN64~/Desktop/rewrite(my\_branch|REBASE 2/2)**

Now you need to Unstage the Commit

**$ git reset HEAD^**

Now You check the Status and Now you can see Untracked Files which are present in Working Area.

**$ git status**

interactive rebase in progress; onto 5f0f13e

Last commands done (2 commands done):

pick 3f3df2a Added A1 File

edit 913cad5 Added Files and Bug Fixed

No commands remaining.

You are currently editing a commit while rebasing branch 'my\_branch' on '5f0f13e'.

(use "git commit --amend" to amend the current commit)

(use "git rebase --continue" once you are satisfied with your changes)

Untracked files:

(use "git add <file>..." to include in what will be committed)

F5.c

F6.c

F7.java

nothing added to commit but untracked files present (use "git add" to track)

Now you can Commit Files with Separate Commit

We are Moving these 2 files F5.c and F6.c with Commit ID 1

F7.java with Commit ID 2

**Dell@Dell-PC MINGW64 ~/Desktop/rewrite (my\_branch|REBASE 2/2)**

**$ git add F5.c F6.c**

**$ git commit -m "Added Files"**

[detached HEAD 4702caf] Added Files

2 files changed, 0 insertions(+), 0 deletions(-)

create mode 100644 F5.c

create mode 100644 F6.c

**Dell@Dell-PC MINGW64 ~/Desktop/rewrite (my\_branch|REBASE 2/2)**

**$ git add F7.java**

**$ git commit -m "Bug Fixed"**

[detached HEAD 1ef5a86] Bug Fixed

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 F7.java

**$ git log --oneline**

**1ef5a86 (HEAD) Bug Fixed**

**4702caf Added Files**

3f3df2a Added A1 File

5f0f13e (master) F1.c Added

Now you need to go My\_Branch from **My\_Branch|REBASE** and continue

**Dell@Dell-PC MINGW64 ~/Desktop/rewrite (my\_branch|REBASE 2/2)**

**$ git rebase --continue**

**$ git log --oneline**

**1ef5a86 (HEAD -> my\_branch) Bug Fixed**

**4702caf Added Files**

3f3df2a Added A1 File

5f0f13e (master) F1.c Added

**Now you can see 1 Commit is Splitted into 2 Commits.**