GIT BRANCHING AND MERGING

To know the current active branch
#git branch

To create a branch
#git branch branch_name

Why do we need to create a branch?

Suppose we need to create a new feature. What we usually do is:

1. Create a new branch
2. Code for new features.
3. Make all commits to that branch.
4. Then merge it with main branch

This is the workflow that usually happens.

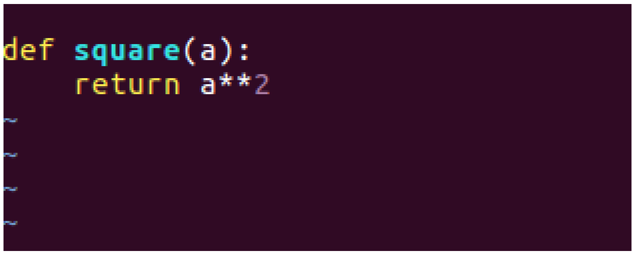
--

Scenario: (Fast forward merge)

Suppose We need to implement a new feature for finding square of a number. As it is considered as a new feature, it is to be implemented from new_feature branch.

Create a new branch
# git branch new_feature

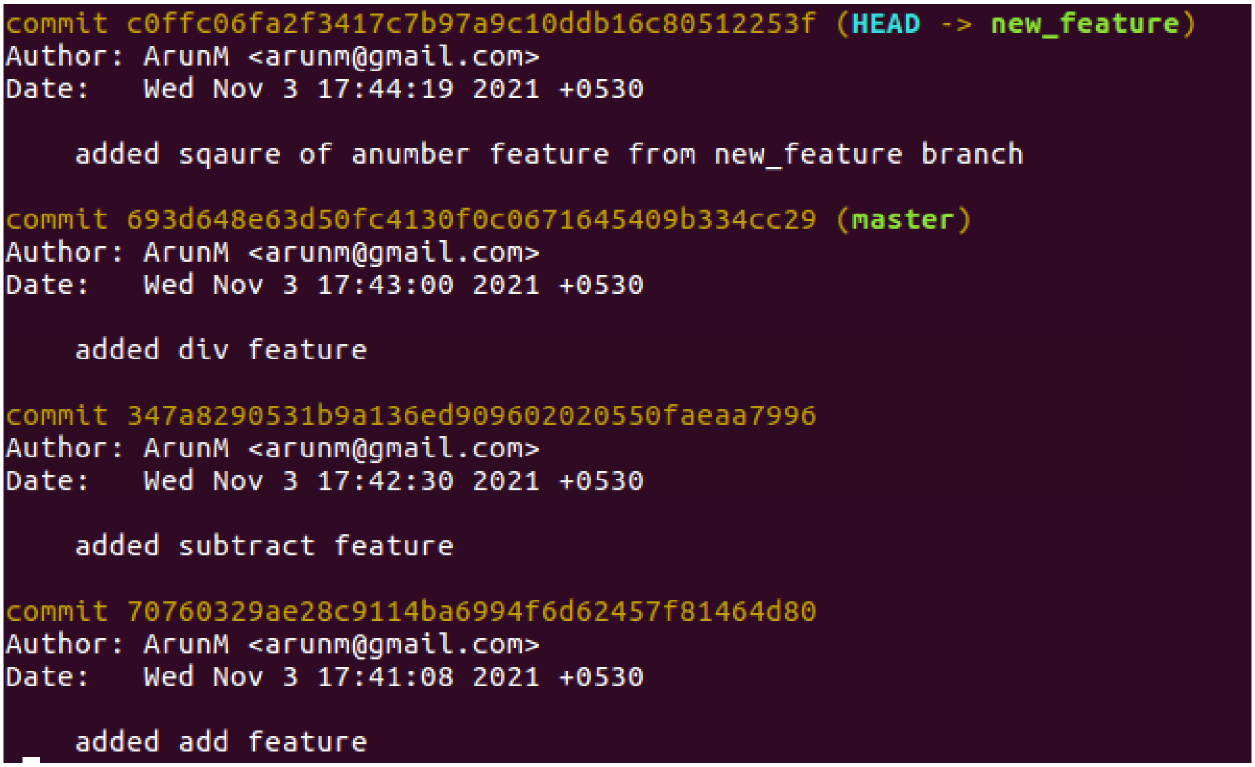
Switch to that branch
# git checkout new_feature
# vi power.py



```
def square(a):
    return a**2
~
~
~
~
```

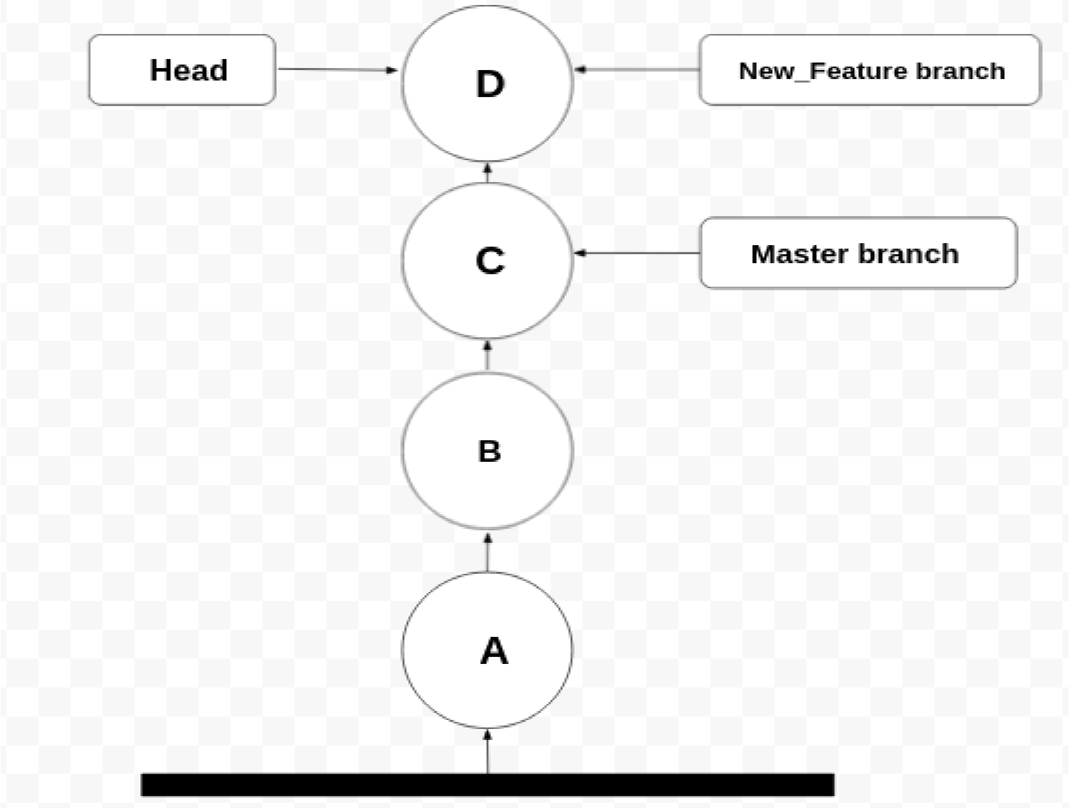
# git add power.py
# git commit -m “added square of a number feature from new_feature branch”
Just check for logs

#git log --all --decorate



```
commit c0ffc06fa2f3417c7b97a9c10ddb16c80512253f (HEAD -> new_feature)
Author: ArunM <arunm@gmail.com>
Date:   Wed Nov 3 17:44:19 2021 +0530

    added sqaure of anumber feature from new_feature branch

commit 693d648e63d50fc4130f0c0671645409b334cc29 (master)
Author: ArunM <arunm@gmail.com>
Date:   Wed Nov 3 17:43:00 2021 +0530

    added div feature

commit 347a8290531b9a136ed909602020550faeaa7996
Author: ArunM <arunm@gmail.com>
Date:   Wed Nov 3 17:42:30 2021 +0530

    added subtract feature

commit 70760329ae28c9114ba6994f6d62457f81464d80
Author: ArunM <arunm@gmail.com>
Date:   Wed Nov 3 17:41:08 2021 +0530

    added add feature
```

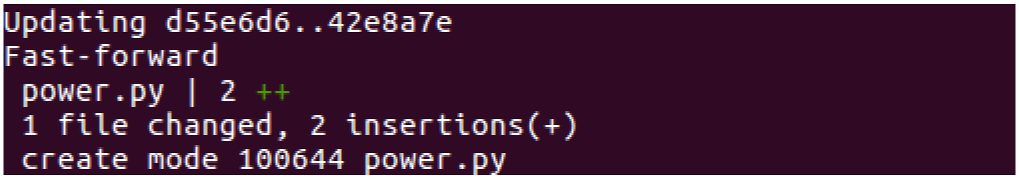
Git flow looks as follows:



Now we can go to master and merge this content from there

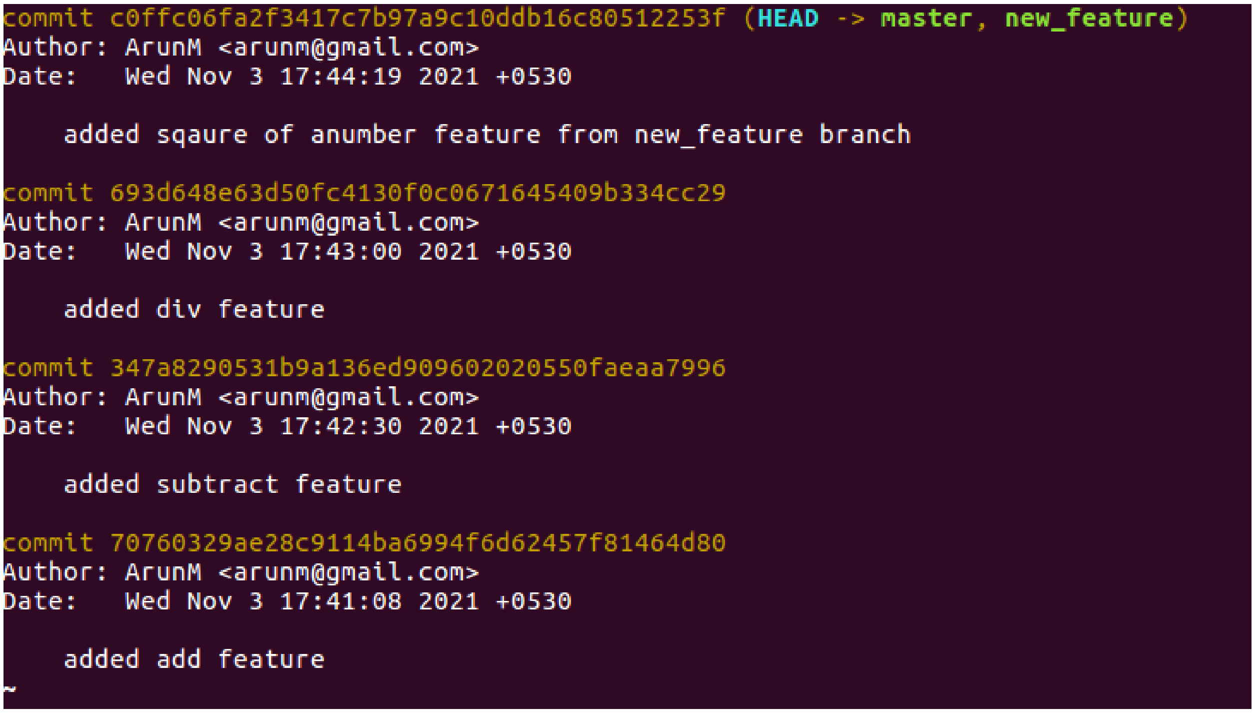
# git checkout master

# git merge new_feature



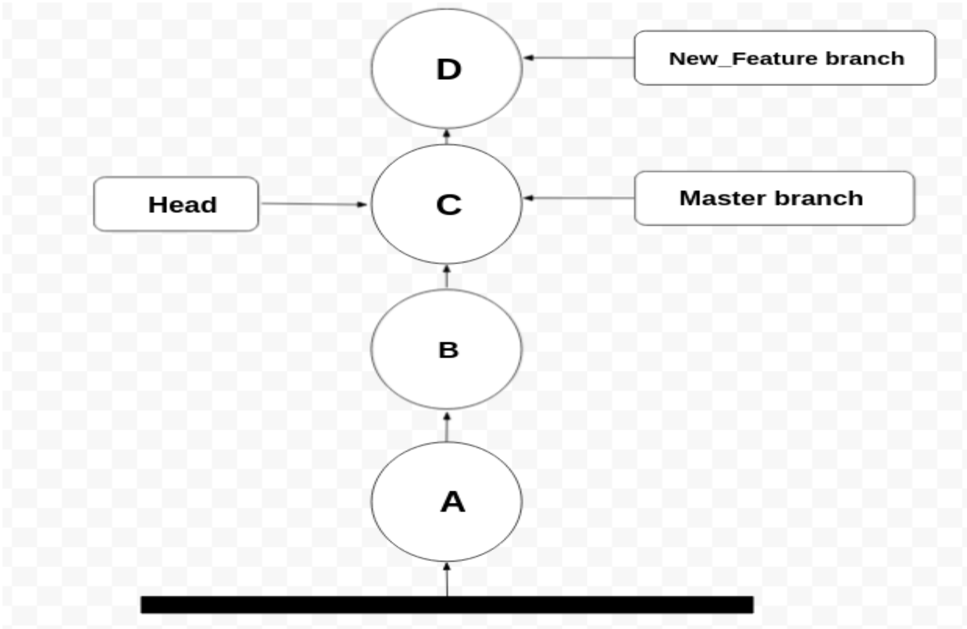
```
Updating d55e6d6..42e8a7e
Fast-forward
 power.py | 2 ++
 1 file changed, 2 insertions(+)
 create mode 100644 power.py
```

# git log --all --decorate



```
commit c0ffc06fa2f3417c7b97a9c10ddb16c80512253f (HEAD -> master, new_feature)
Author: ArunM <arunm@gmail.com>
Date:   Wed Nov 3 17:44:19 2021 +0530

    added sqaure of anumber feature from new_feature branch

commit 693d648e63d50fc4130f0c0671645409b334cc29
Author: ArunM <arunm@gmail.com>
Date:   Wed Nov 3 17:43:00 2021 +0530

    added div feature

commit 347a8290531b9a136ed909602020550faeaa7996
Author: ArunM <arunm@gmail.com>
Date:   Wed Nov 3 17:42:30 2021 +0530

    added subtract feature

commit 70760329ae28c9114ba6994f6d62457f81464d80
Author: ArunM <arunm@gmail.com>
Date:   Wed Nov 3 17:41:08 2021 +0530

    added add feature
~
```

Now we can see that HEAD is moved to the master branch and it is up to date with new feature branch.



Scenario: (Recursive Merge)

For our calculator, We need to create 2 new features .

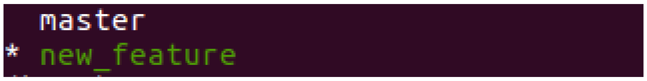
1. ***Cube of a number -- to be implemented in new_feature branch on separate callable file (since it is considered as a new feature)***
2. ***Multiplication --- to be implemented on calc.py on main branch after implementing cube of a number***

Note: Recursive merge is used in cases where Suppose two branches are moving in different directions. For example: A new feature is added in the new_feature branch. Then some code is changed from the main branch.

First let us code for the cube of a number feature.

To switch to new branch
# git checkout new_feature

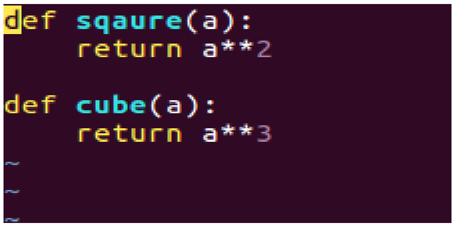
Now just check the branch we are working by
#git branch



```
  master
* new_feature
```

Now enter the code for cube of a number on power.py

# vi power.py

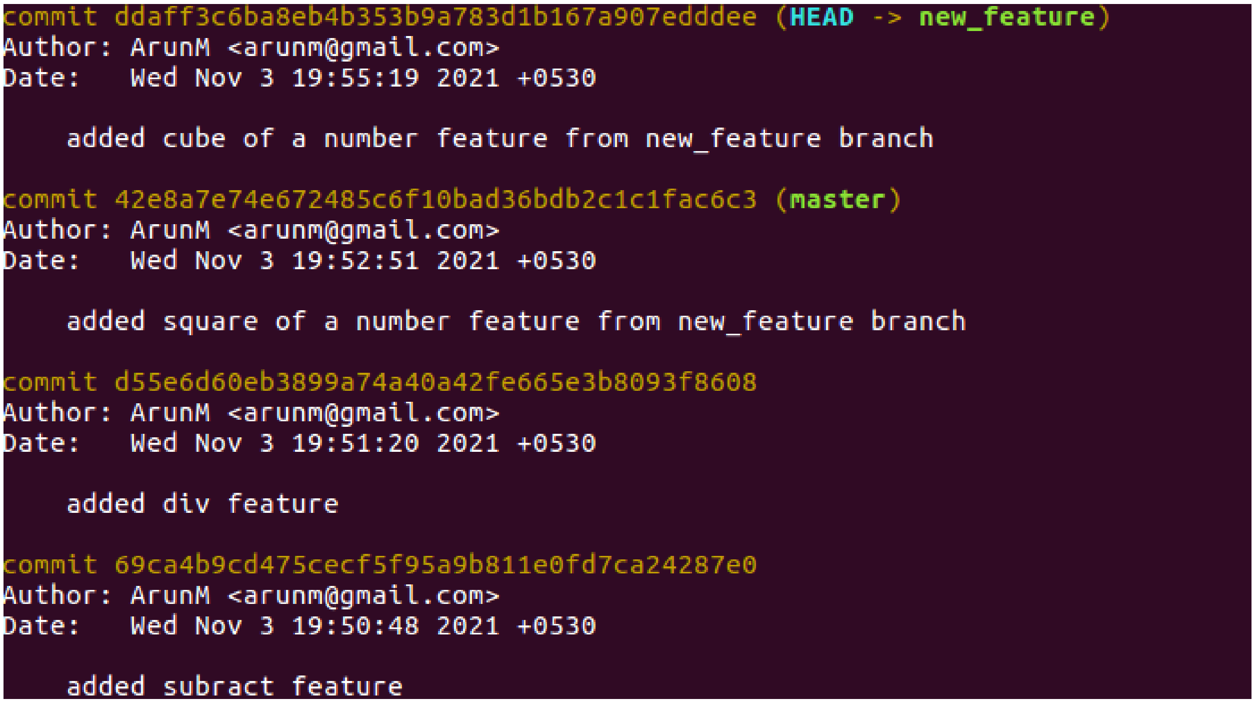


```
def sqaure(a):
    return a**2

def cube(a):
    return a**3
~
~
~
```

# git add .

# git commit -m “added cube of a number feature from new_feature branch”

# git log --all --decorate



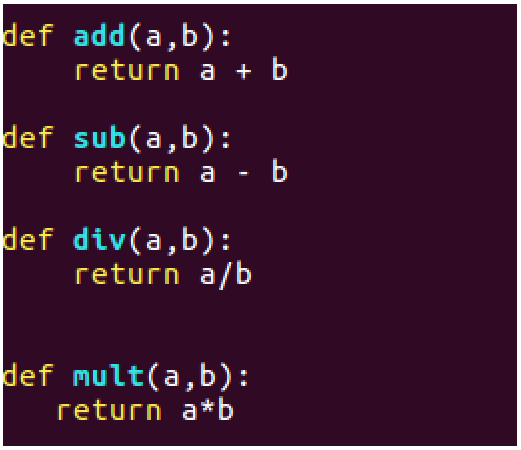
```
commit ddaff3c6ba8eb4b353b9a783d1b167a907edddee (HEAD -> new_feature)
Author: ArunM <arunm@gmail.com>
Date:   Wed Nov 3 19:55:19 2021 +0530

    added cube of a number feature from new_feature branch

commit 42e8a7e74e672485c6f10bad36bdb2c1c1fac6c3 (master)
Author: ArunM <arunm@gmail.com>
Date:   Wed Nov 3 19:52:51 2021 +0530

    added square of a number feature from new_feature branch

commit d55e6d60eb3899a74a40a42fe665e3b8093f8608
Author: ArunM <arunm@gmail.com>
Date:   Wed Nov 3 19:51:20 2021 +0530

    added div feature

commit 69ca4b9cd475cecf5f95a9b811e0fd7ca24287e0
Author: ArunM <arunm@gmail.com>
Date:   Wed Nov 3 19:50:48 2021 +0530

    added subract feature
```

We can see that we had made a commit from the new_feature branch.

Now switch back to master branch and make edits in calc.py

# git checkout master

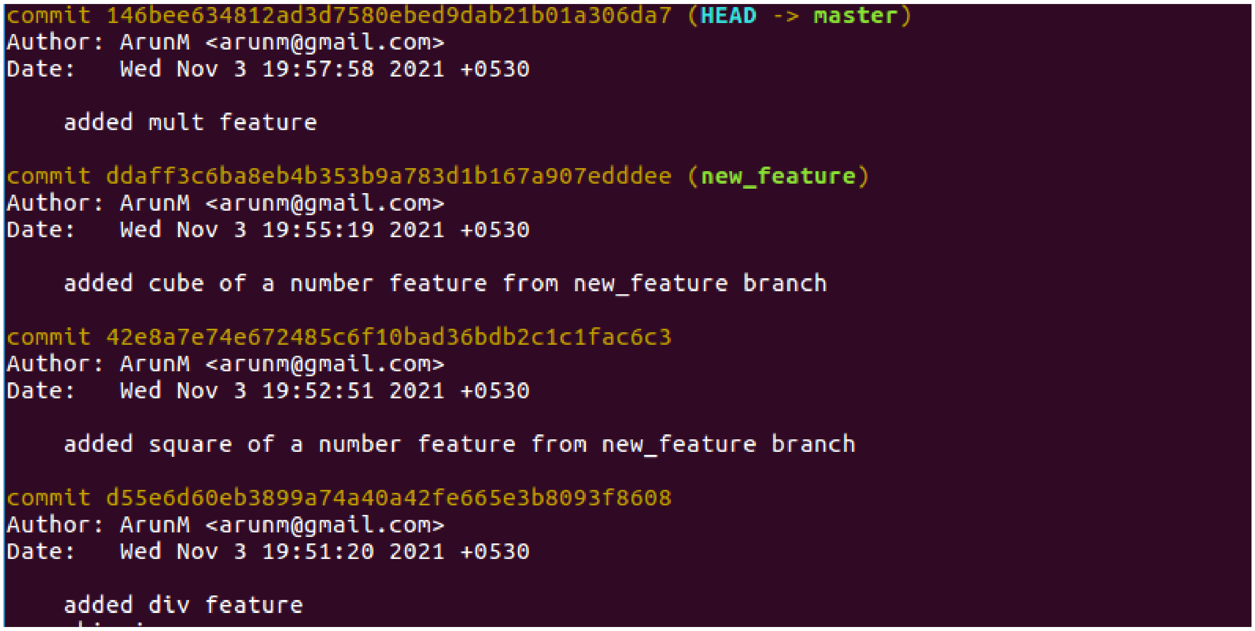
# vi calc.py



```
def add(a,b):
    return a + b

def sub(a,b):
    return a - b

def div(a,b):
    return a/b


def mult(a,b):
   return a*b
```

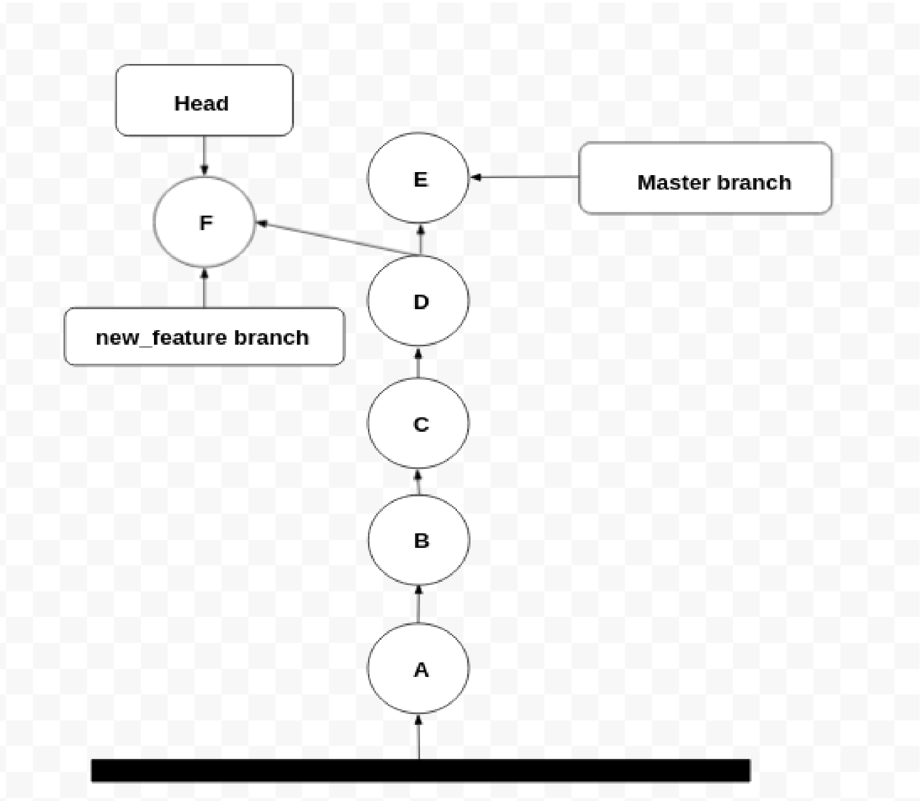
# git add .
# git commit -m “added mult feature”

Check for logs
#git log --all --decorate



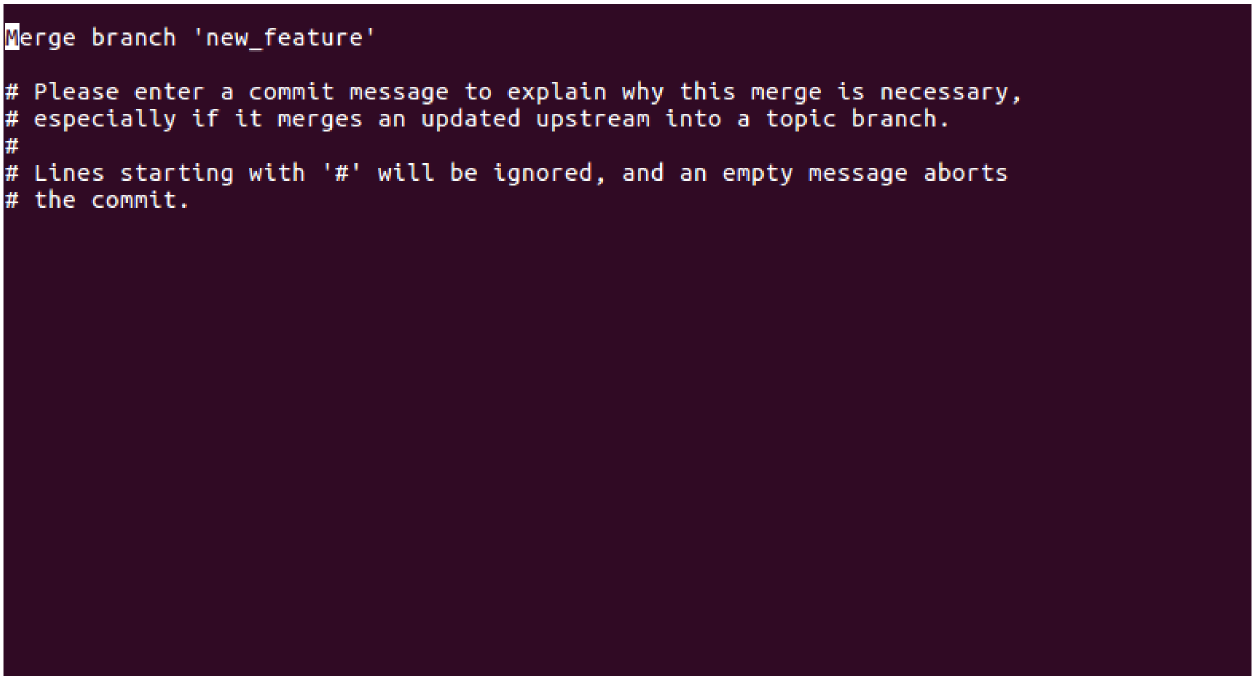
```
commit 146bee634812ad3d7580ebed9dab21b01a306da7 (HEAD -> master)
Author: ArunM <arunm@gmail.com>
Date:   Wed Nov 3 19:57:58 2021 +0530

    added mult feature

commit ddaff3c6ba8eb4b353b9a783d1b167a907edddee (new_feature)
Author: ArunM <arunm@gmail.com>
Date:   Wed Nov 3 19:55:19 2021 +0530

    added cube of a number feature from new_feature branch

commit 42e8a7e74e672485c6f10bad36bdb2c1c1fac6c3
Author: ArunM <arunm@gmail.com>
Date:   Wed Nov 3 19:52:51 2021 +0530

    added square of a number feature from new_feature branch

commit d55e6d60eb3899a74a40a42fe665e3b8093f8608
Author: ArunM <arunm@gmail.com>
Date:   Wed Nov 3 19:51:20 2021 +0530

    added div feature
```

Also we can see commits made from the master branch.



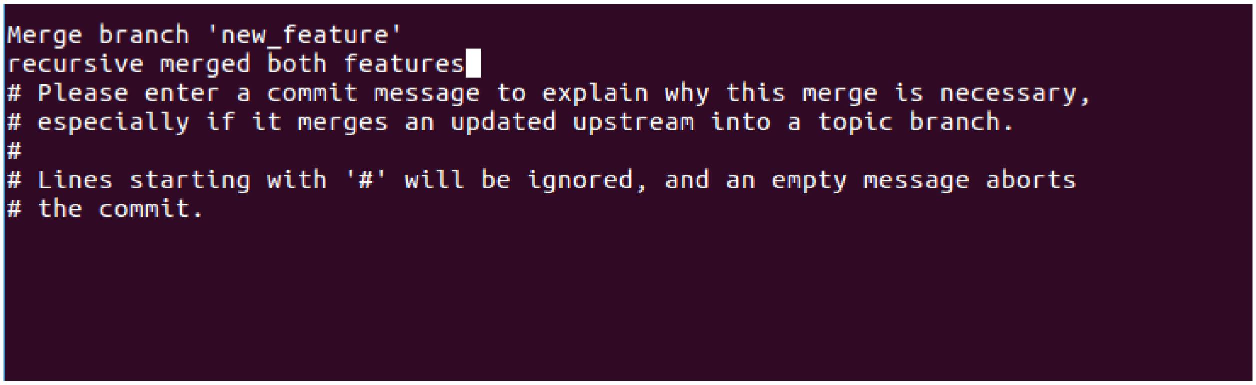
Currently the HEAD is pointing with masterbranch. After merging it will be aligned with both.

# git merge new_feature
A blank window will come

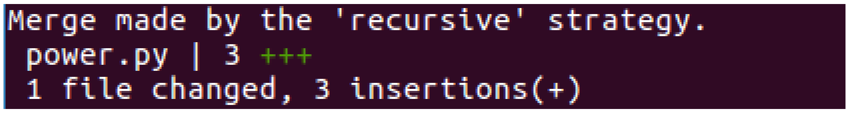


```
Merge branch 'new_feature'

# Please enter a commit message to explain why this merge is necessary,
# especially if it merges an updated upstream into a topic branch.
#
# Lines starting with '#' will be ignored, and an empty message aborts
# the commit.
```

Type the required message and save it



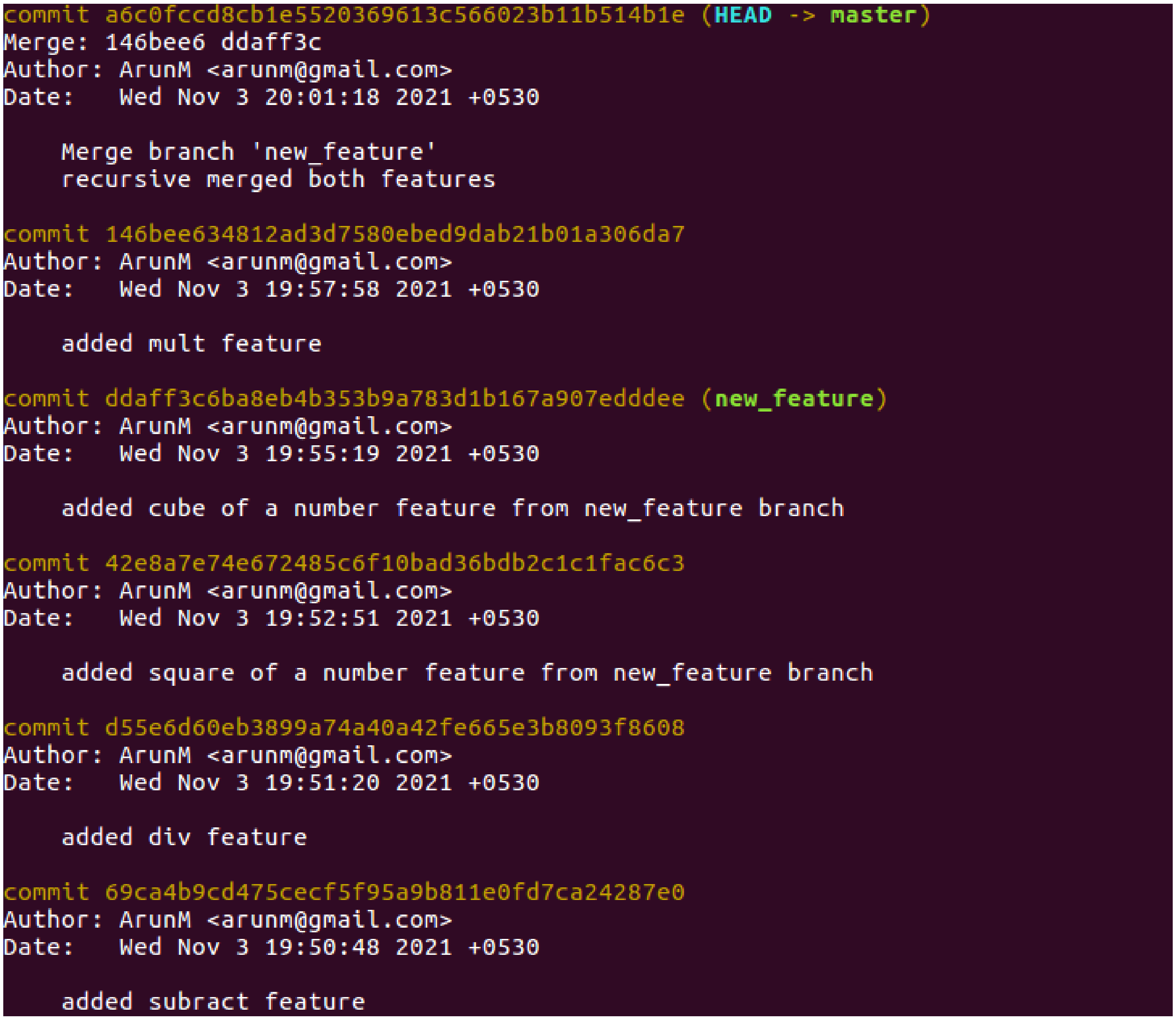
```
Merge branch 'new_feature'
recursive merged both features
# Please enter a commit message to explain why this merge is necessary,
# especially if it merges an updated upstream into a topic branch.
#
# Lines starting with '#' will be ignored, and an empty message aborts
# the commit.
```



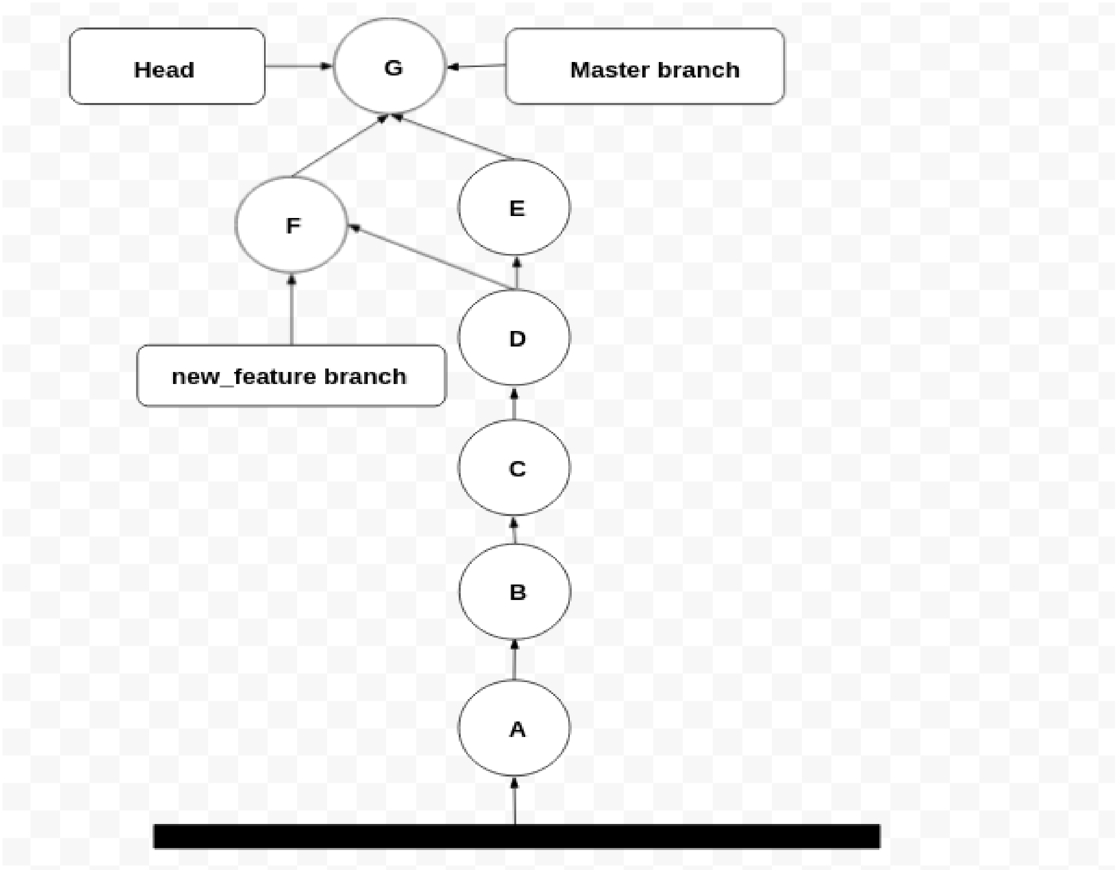
```
Merge made by the 'recursive' strategy.
 power.py | 3 +++
 1 file changed, 3 insertions(+)
```

Now our merge is successful.
Check for logs

# git log --all --decorate



```
commit a6c0fccd8cb1e5520369613c566023b11b514b1e (HEAD -> master)
Merge: 146bee6 ddaff3c
Author: ArunM <arunm@gmail.com>
Date:   Wed Nov 3 20:01:18 2021 +0530

    Merge branch 'new_feature'
    recursive merged both features

commit 146bee634812ad3d7580ebed9dab21b01a306da7
Author: ArunM <arunm@gmail.com>
Date:   Wed Nov 3 19:57:58 2021 +0530

    added mult feature

commit ddaff3c6ba8eb4b353b9a783d1b167a907edddee (new_feature)
Author: ArunM <arunm@gmail.com>
Date:   Wed Nov 3 19:55:19 2021 +0530

    added cube of a number feature from new_feature branch

commit 42e8a7e74e672485c6f10bad36bdb2c1c1fac6c3
Author: ArunM <arunm@gmail.com>
Date:   Wed Nov 3 19:52:51 2021 +0530

    added square of a number feature from new_feature branch

commit d55e6d60eb3899a74a40a42fe665e3b8093f8608
Author: ArunM <arunm@gmail.com>
Date:   Wed Nov 3 19:51:20 2021 +0530

    added div feature

commit 69ca4b9cd475cecf5f95a9b811e0fd7ca24287e0
Author: ArunM <arunm@gmail.com>
Date:   Wed Nov 3 19:50:48 2021 +0530

    added subract feature
```



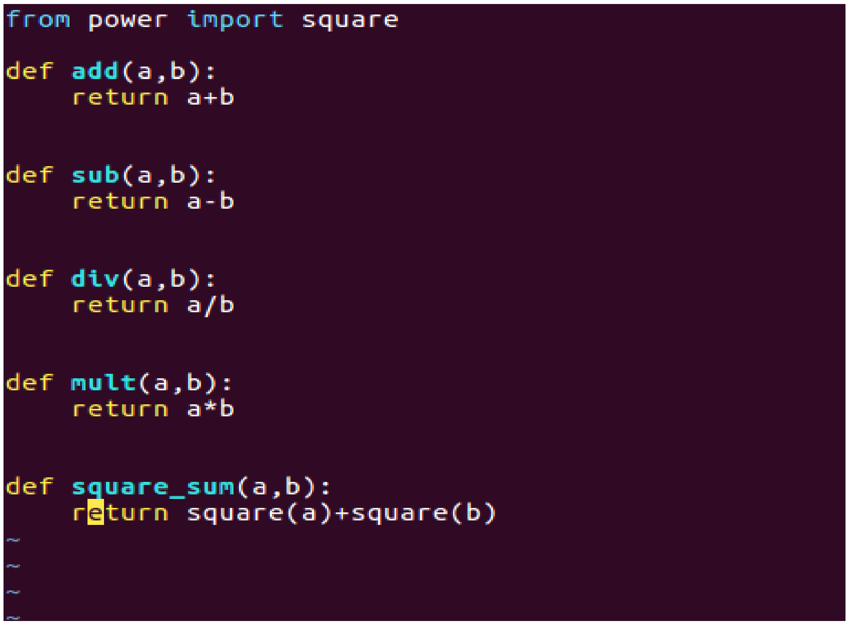
Note: Actually during such a merge a new commit is created.

Scenario 3:(Solving recursive merge conflict)

Suppose we made some changes to file calc.py from master and we committed it. Then we made some changes to the same file from new_feature and committed it. Then if we merge these , conflict can occur since we are working on the same file. We don't know which code to keep and which to remove.

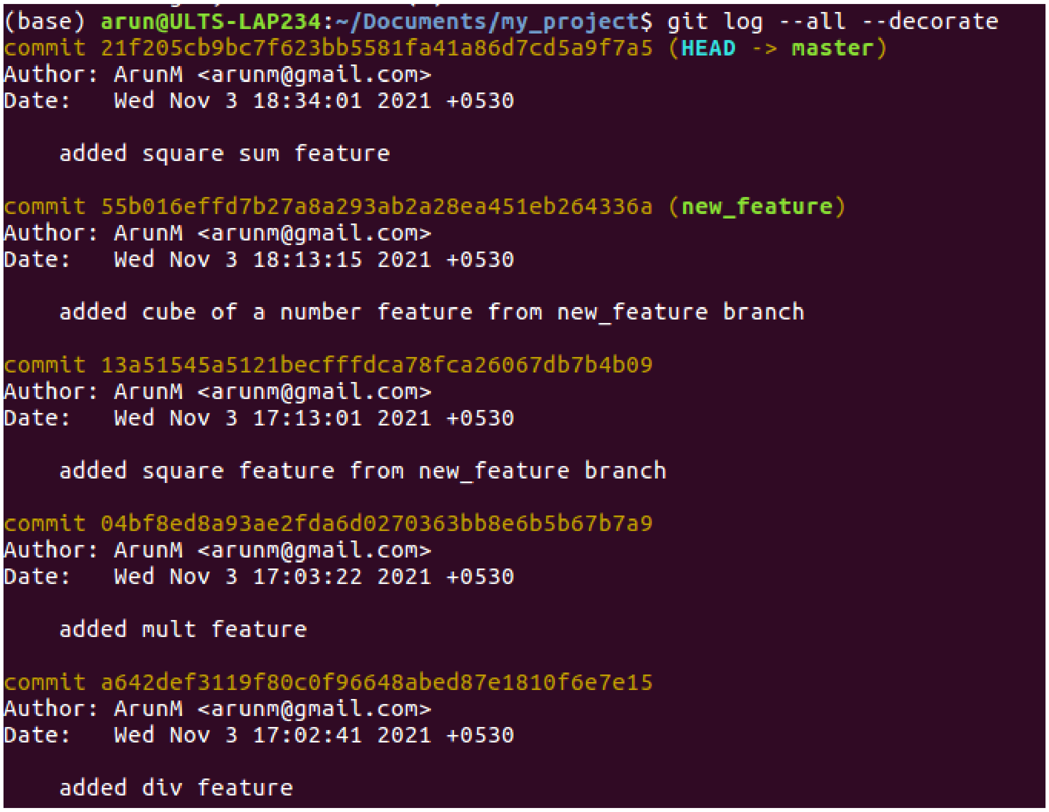
Open calc.py and add some lines. Here we add a function to find the sum of square of two numbers. For that we imported square function from power.py.

# vim calc.py



```
from power import square

def add(a,b):
    return a+b


def sub(a,b):
    return a-b


def div(a,b):
    return a/b


def mult(a,b):
    return a*b


def square_sum(a,b):
    return square(a)+square(b)
~
~
~
~
```

# git add .
# git commit -m "added square sum feature"
Now check for logs
# git log --all --decorate



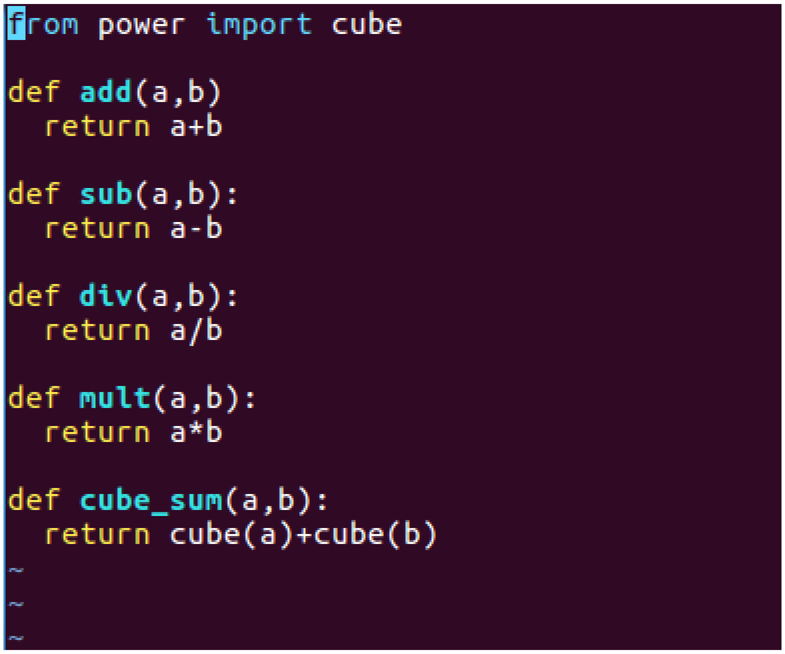
```
(base) arun@ULTS-LAP234:~/Documents/my_project$ git log --all --decorate
commit 21f205cb9bc7f623bb5581fa41a86d7cd5a9f7a5 (HEAD -> master)
Author: ArunM <arunm@gmail.com>
Date:   Wed Nov 3 18:34:01 2021 +0530

    added square sum feature

commit 55b016effd7b27a8a293ab2a28ea451eb264336a (new_feature)
Author: ArunM <arunm@gmail.com>
Date:   Wed Nov 3 18:13:15 2021 +0530

    added cube of a number feature from new_feature branch

commit 13a51545a5121becfffdca78fca26067db7b4b09
Author: ArunM <arunm@gmail.com>
Date:   Wed Nov 3 17:13:01 2021 +0530

    added square feature from new_feature branch

commit 04bf8ed8a93ae2fda6d0270363bb8e6b5b67b7a9
Author: ArunM <arunm@gmail.com>
Date:   Wed Nov 3 17:03:22 2021 +0530

    added mult feature

commit a642def3119f80c0f96648abed87e1810f6e7e15
Author: ArunM <arunm@gmail.com>
Date:   Wed Nov 3 17:02:41 2021 +0530

    added div feature
```

Then go to new_feature branch and make changes to the same file.

# git checkout new_feature

Open calc.py and add some lines. Here we add a function to find the sum of cube of two numbers. For that we imported the cube function from power.py.

# vim calc.py



```
from power import cube

def add(a,b)
  return a+b

def sub(a,b):
  return a-b

def div(a,b):
  return a/b

def mult(a,b):
  return a*b

def cube_sum(a,b):
  return cube(a)+cube(b)
~
~
~
```

Now add the file to staging

# git add .

Finally move the file to commit area

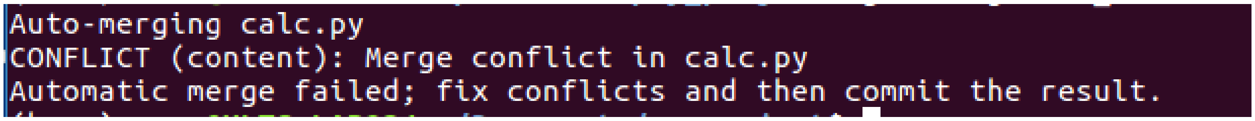
#git commit -m "added cube sum feature from new_feature branch"

Now as a developer once we finish the new feature, if the code is fine, we have to merge it with master. For that go to the master branch.

Switch to master branch

# git checkout master

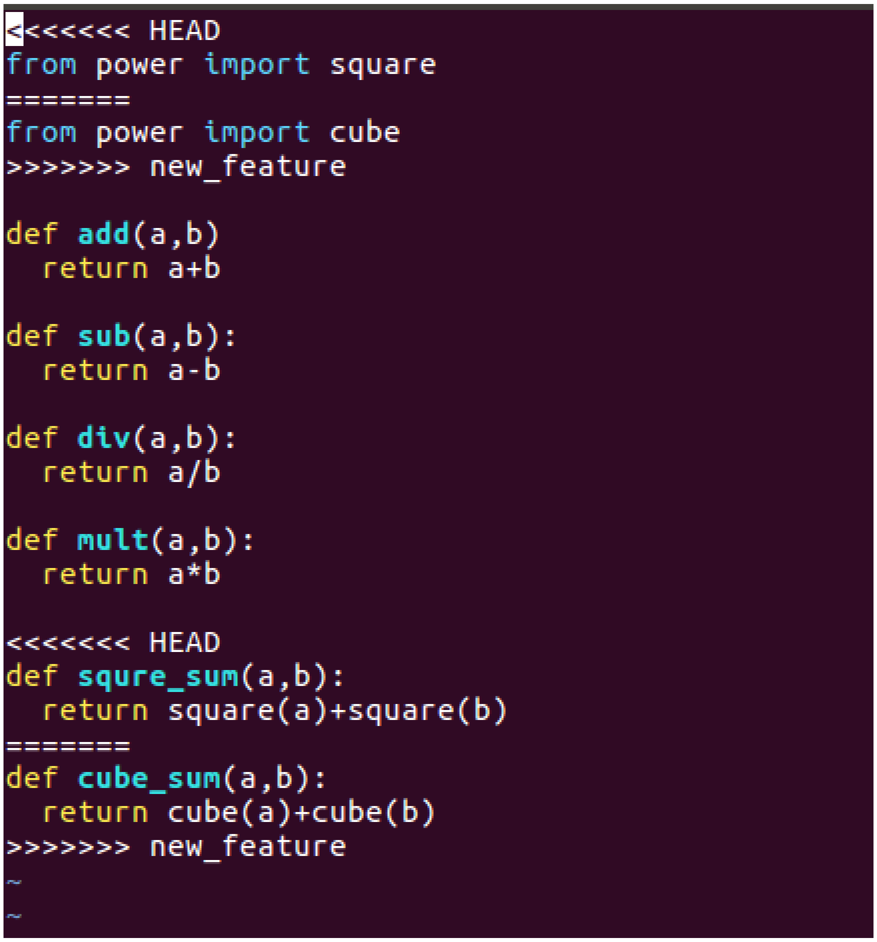
# git merge new_feature



```
Auto-merging calc.py
CONFLICT (content): Merge conflict in calc.py
Automatic merge failed; fix conflicts and then commit the result.
```

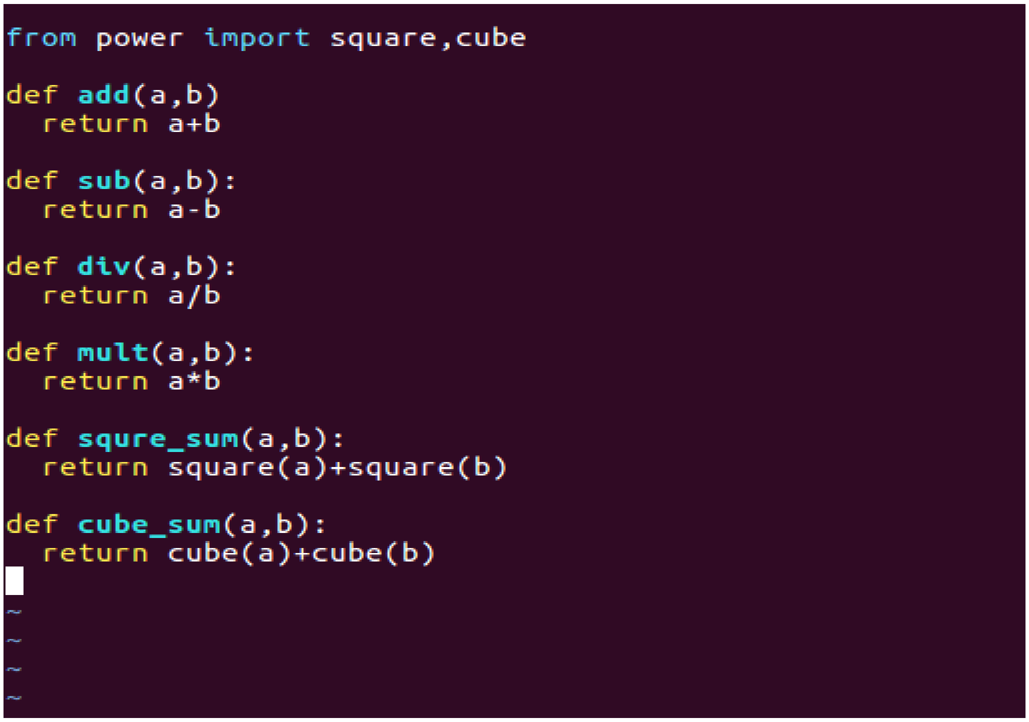
While merging we can see a conflict message. It is because the same file gets edited from both the master branch and the new_feature branch. Git doesn't know which code to keep. So we have to manually decide what to keep. For that open the file where conflict exist

# vim calc.py



```
<<<<<<< HEAD
from power import square
=======
from power import cube
>>>>>>> new_feature

def add(a,b)
  return a+b

def sub(a,b):
  return a-b

def div(a,b):
  return a/b

def mult(a,b):
  return a*b

<<<<<<< HEAD
def squre_sum(a,b):
  return square(a)+square(b)
=======
def cube_sum(a,b):
  return cube(a)+cube(b)
>>>>>>> new_feature
~
~
```

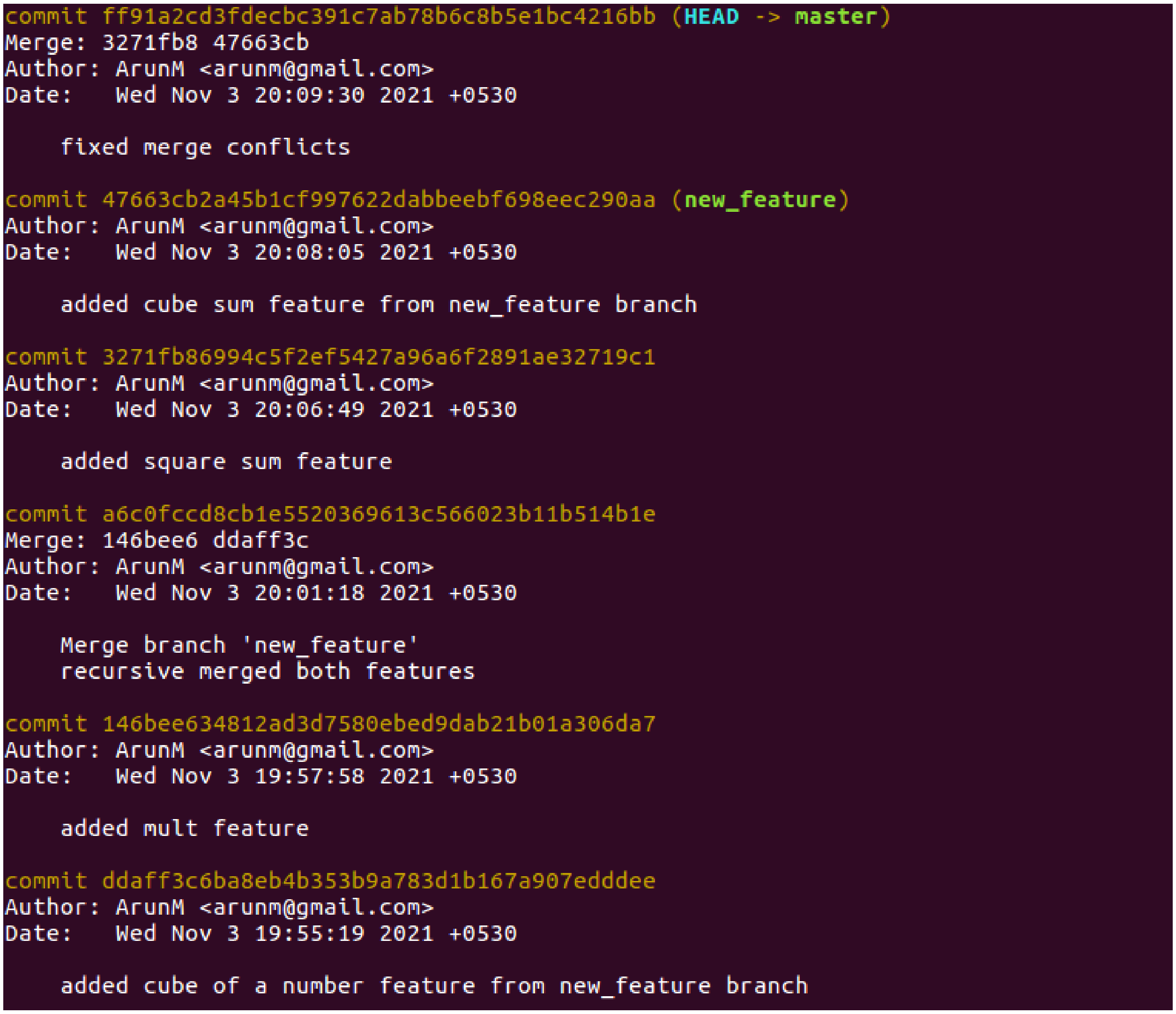
In the above file we can see the edits made from both the branches. Edit the file to keep only the necessary code.



```
from power import square,cube

def add(a,b)
  return a+b

def sub(a,b):
  return a-b

def div(a,b):
  return a/b

def mult(a,b):
  return a*b

def squre_sum(a,b):
  return square(a)+square(b)

def cube_sum(a,b):
  return cube(a)+cube(b)

~
~
~
~
```

Now I have edited the file. Finally move the file to staging and commit it.

```
#git add calc.py
#git commit -m “fixed merge conflicts”
# git merge new_feature
```



```
commit ff91a2cd3fdecbc391c7ab78b6c8b5e1bc4216bb (HEAD -> master)
Merge: 3271fb8 47663cb
Author: ArunM <arunm@gmail.com>
Date:   Wed Nov 3 20:09:30 2021 +0530

    fixed merge conflicts

commit 47663cb2a45b1cf997622dabbeebf698eec290aa (new_feature)
Author: ArunM <arunm@gmail.com>
Date:   Wed Nov 3 20:08:05 2021 +0530

    added cube sum feature from new_feature branch

commit 3271fb86994c5f2ef5427a96a6f2891ae32719c1
Author: ArunM <arunm@gmail.com>
Date:   Wed Nov 3 20:06:49 2021 +0530

    added square sum feature

commit a6c0fccd8cb1e5520369613c566023b11b514b1e
Merge: 146bee6 ddaff3c
Author: ArunM <arunm@gmail.com>
Date:   Wed Nov 3 20:01:18 2021 +0530

    Merge branch 'new_feature'
    recursive merged both features

commit 146bee634812ad3d7580ebed9dab21b01a306da7
Author: ArunM <arunm@gmail.com>
Date:   Wed Nov 3 19:57:58 2021 +0530

    added mult feature

commit ddaff3c6ba8eb4b353b9a783d1b167a907edddee
Author: ArunM <arunm@gmail.com>
Date:   Wed Nov 3 19:55:19 2021 +0530

    added cube of a number feature from new_feature branch
```

We can see it is up to date.