



Week 3: Assignment 1

Help us analyse feedback of participants in eYRC

[Last Updated on: **21st April 2021, 19:00 Hrs**]

- Aim
- Given
- Procedure
 - Voila! You have now successfully completed this Assignment.
 - The deadline of this Assignment is **26th Apr 2021, 1700 Hrs**.
 - The total marks (10) will be updated after the deadline on or before **27th Apr 2021**, provided the test cases on both **master** and **feature** branches are **PASSED**.
- References

Aim

e-Yantra Robotics Competition (eYRC) is a unique annual competition for students in Engineering/ Science/ Polytechnic colleges. The entire competition is divided into small tasks which all the participating teams have to complete within a deadline.

In order to improve the theme, we take regular feedback from the students.

One such question in the feedback form is to **judge the understanding/knowledge (on a scale of 1-10) of participants** before and after completing the task.

To make sense of the data, we need your help to find the

- average knowledge of participants before & after completing the task
- percentage increase in knowledge between the number of participants who rated ≤ 2 before attempting the task and ≥ 5 after attempting the task

Now, write an **AWK (.awk)** script to scan through the data in the given CSV file and help us achieve this goal.

DISCLAIMER:

The data provided is to be solely used to solve this assignment. It does not relate to the actual figures of the competition.

Given

- Two files are provided to solve this assignment.
 - Skeleton program file: **week3_assignment1.awk** (Contents of this file on **master** and **feature** branch are different)
 - Sample CSV file: **week3_assignment1_sample.csv**

- Before moving on to the Procedure section, make sure you have gone through the **Week 3 Live Session on Git and GitHub** conducted on *April 21st, 2021*.
- You can find the link in the [Live Sessions](#) document.

Procedure

- Visit your **Private Repository of Assignment** that you must have received after performing the steps as mentioned in [Introduction to GitHub Classroom](#).
- Copy the URL of your repository as shown in Figure 1.

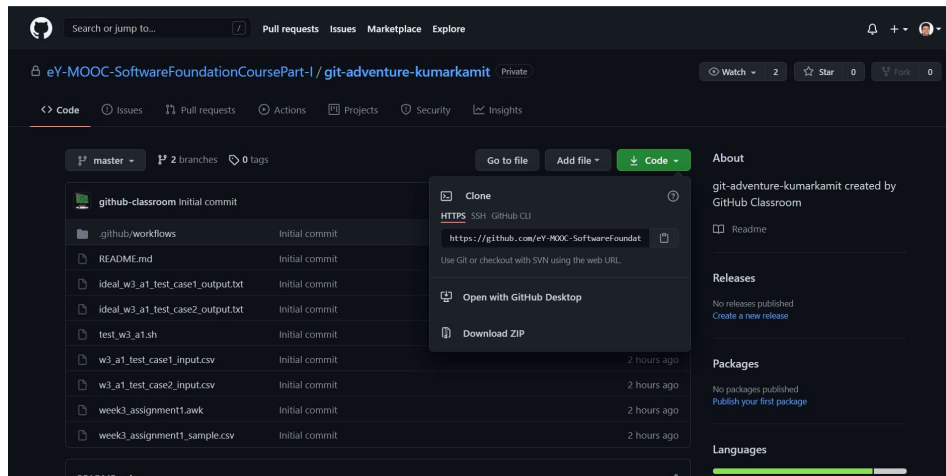


Figure 1: Copy Private Repository URL from the GitHub website.

- Open Terminal and navigate to the directory where you want to clone your repository.
- Now, clone the repository into your machine using the following command:

```
git clone https://github.com/eY-MOOC-SoftwareFoundationCoursePart-I/git-adventure-kumarkamit
```

- Navigate to the cloned repository and use the **ls** command to view the contents of the directory. The contents will be as shown in Figure 2.

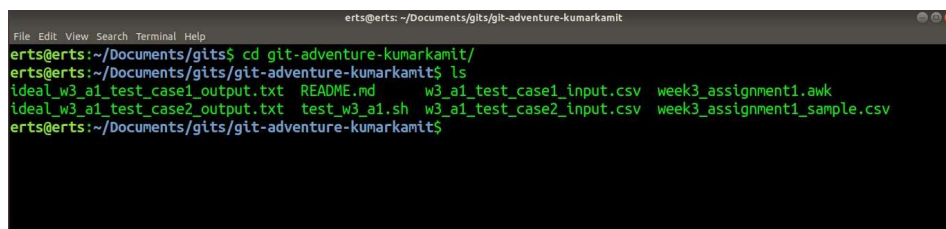
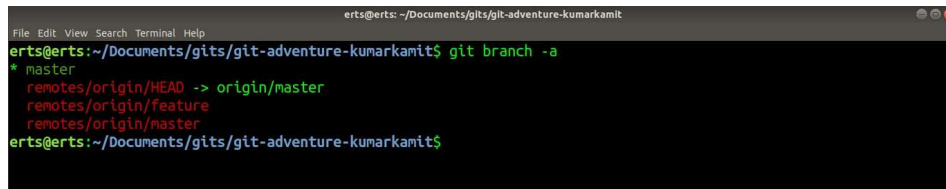


Figure 2: Contents of the 'Master' branch in the cloned repository.

- Use the following command to find the branches in your repository.

```
git branch -a
```



```

erts@erts: ~/Documents/gits/git-adventure-kumarkamit
File Edit View Search Terminal Help
erts@erts:~/Documents/gits/git-adventure-kumarkamit$ git branch -a
* master
  remotes/origin/HEAD -> origin/master
  remotes/origin/feature
  remotes/origin/master
erts@erts:~/Documents/gits/git-adventure-kumarkamit$

```

Figure 3: View the branches in the cloned repository.

- We will talk about the contents of **master** branch later. Let us first switch to the **feature** branch.
- Use the **checkout** command as mentioned below:

```
git checkout feature
```



- Use the **ls** command to view the contents of the directory. The contents will be as shown in Figure 4.



```

erts@erts: ~/Documents/gits/git-adventure-kumarkamit
File Edit View Search Terminal Help
erts@erts:~/Documents/gits/git-adventure-kumarkamit$ ls
feature_ideal_w3_a1_test_case1_output.txt  test_w3_a1_feature.sh      week3_assignment1.awk
feature_ideal_w3_a1_test_case2_output.txt  w3_a1_test_case1_input.csv week3_assignment1_sample.csv
README.md                                  w3_a1_test_case2_input.csv
erts@erts:~/Documents/gits/git-adventure-kumarkamit$

```

Figure 4: Contents of the **feature** branch after using **checkout** command.

- An AWK script by the name **week3_assignment1.awk** will be present.
- You need to edit this file in order to solve the **2nd part of the assignment**.
- The contents of **week3_assignment1.awk** will be as follows:

```

#!/bin/awk -f

BEGIN{
    FS=",";
    know_less_than_eq_two_before = 0;
    know_great_than_eq_five_after = 0;
}

{
    if(NR > 0)
    {
        if($1 <= 1)
            know_less_than_eq_two_before = 1;

        if($2 >= 8)
            know_great_than_eq_five_after = 1;
    }
}

END{
    percen_incr = ( ( know_great_than_eq_five_after - know_less_than_eq_two_befo
    printf "\nPercentage increase in knowledge gained from level 2 or less befor

```

- After you have edited the file, you can check your output by running the following command:

```
bash test_w3_a1_feature.sh week3_assignment1.awk
```



- If all the test cases are passed, you will find the following statement printed in your terminal:

Congrats! All **test** cases PASSED.



- If the test case(s) is/are not passed, you will find the following statement printed in your terminal-

Test case number 1: FAIL



- After you are satisfied with your output and want to push your code to the repo, follow the below steps:

1. Run,

```
git status
```



to check the status of your file. You will observe that state is **modified**.

2. Now run,

```
git add .
```



to stage all the modified files in the directory.

3. Run

```
git status
```



again, to verify whether the files are staged.

4. Run

```
git commit -m "type message for your commit here"
```



to commit all the files which are staged.

5. Run

```
git status
```



to verify that your working tree is clean.

6. Lastly, run

```
git push origin feature
```



Make sure that you enter your **GitHub username and password**.

- Refer Figure 5 to check the output of above commands.

```

erts@erts: ~/Documents/gits/git-adventure-kumarkamit
File Edit View Search Terminal Help
erts@erts:~/Documents/gits/git-adventure-kumarkamit$ git status
On branch feature
Your branch is up to date with 'origin/feature'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   week3_assignment1.awk

no changes added to commit (use "git add" and/or "git commit -a")
erts@erts:~/Documents/gits/git-adventure-kumarkamit$ git add .
erts@erts:~/Documents/gits/git-adventure-kumarkamit$ git status
On branch feature
Your branch is up to date with 'origin/feature'.

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   week3_assignment1.awk

erts@erts:~/Documents/gits/git-adventure-kumarkamit$ git commit -m "SFC A1 Week 3 Solution"
[feature c4dba05] SFC A1 Week 3 Solution
 1 file changed, 1 insertion(+), 1 deletion(-)
erts@erts:~/Documents/gits/git-adventure-kumarkamit$ git status
On branch feature
Your branch is ahead of 'origin/feature' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
erts@erts:~/Documents/gits/git-adventure-kumarkamit$ git push

```

Figure 5: Output of commands required to push a file on remote repository.

- If your code **did not pass all the test cases**, you will observe a :x: next to the commit as shown in Figure 6.

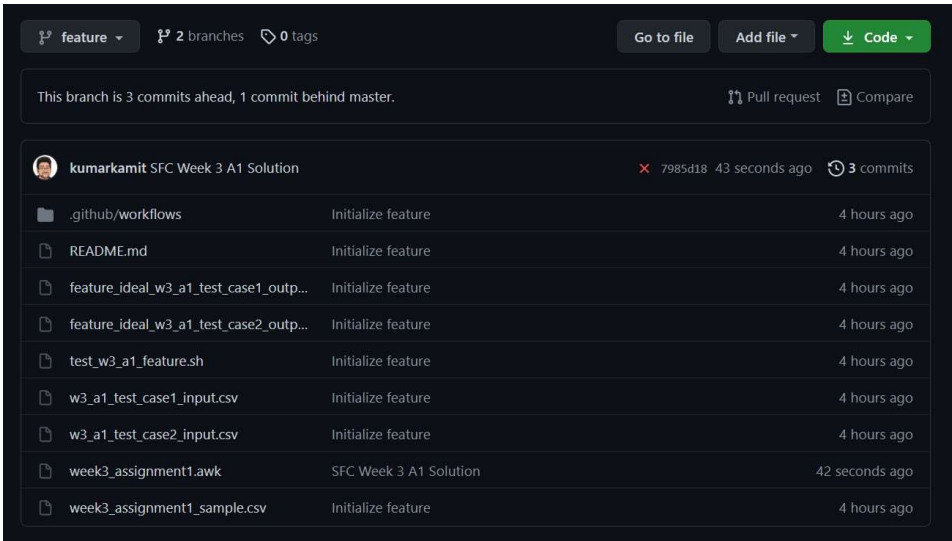


Figure 6: Unsuccessful attempt to clear all the test cases.

- If your code passed all the test cases, you will observe a :heavy_check_mark: next to the commit as shown in Figure 7.

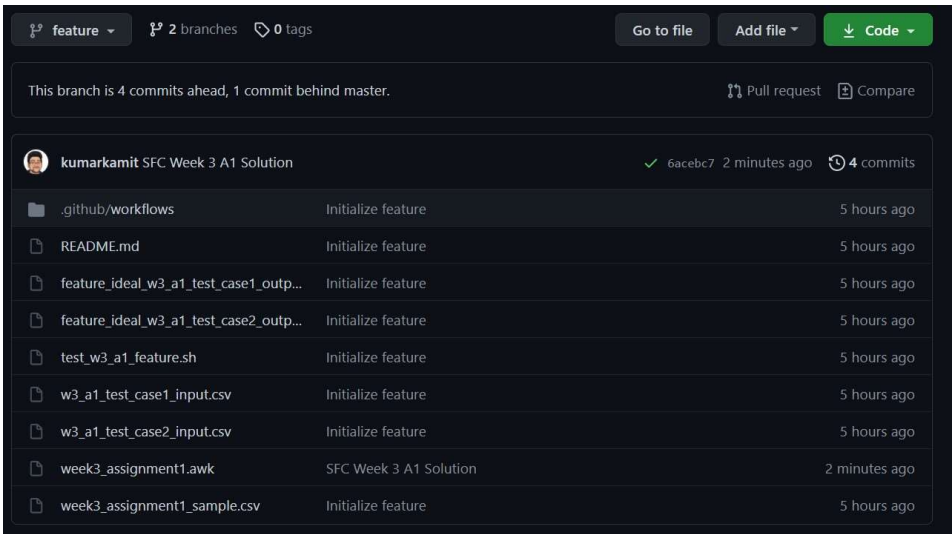


Figure 7: Successful attempt to clear all the test cases.

- You can review the result of all your commits by visiting the **Actions** tab in your Assignment repository as shown in Figure 8.

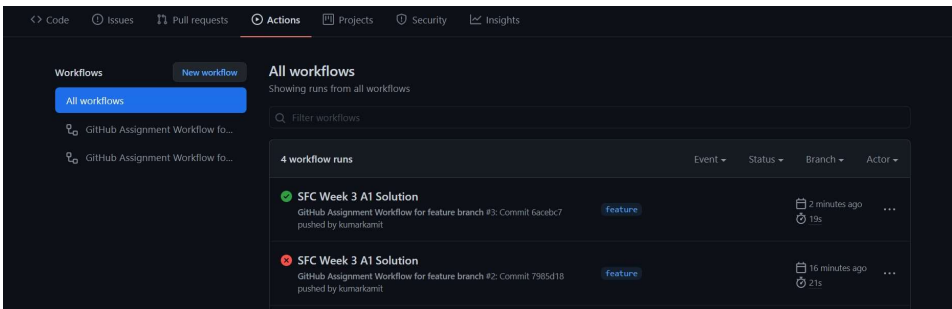


Figure 8: Reviewing the result of each commit on GitHub.

- Once you have passed all test cases on **feature** branch, then move to the **master** branch of the repository. Use the **checkout** command as mentioned below:

```
git checkout master
```



- Use the **ls** command to view the contents of the directory. The contents will be as shown in Figure 9.

```
erts@erts: ~/Documents/gits/git-adventure-kumarkamit
erts@erts:~/Documents/gits/git-adventure-kumarkamit$ git branch
* feature
  master
erts@erts:~/Documents/gits/git-adventure-kumarkamit$ ls
ideal_w3_a1_test_case1_output.txt  README.md      w3_a1_test_case1_input.csv  week3_assignment1.awk
ideal_w3_a1_test_case2_output.txt  test_w3_a1.sh  w3_a1_test_case2_input.csv  week3_assignment1_sample.csv
erts@erts:~/Documents/gits/git-adventure-kumarkamit$
```

Figure 9: Contents of the **master** branch after using **checkout** command.

- You will notice that another **week3_assignment1.awk** is present here. The contents of this file will be as follows:

```
#!/bin/awk -f

BEGIN{
    printf "Average value of knowledge gained (on a scale of 1 to 10) before and

    FS=",";
    sum_scale_know_before = 0;
    sum_scale_know_after = 0;
}

{
    if(NR > 0)
    {
        sum_scale_know_before = sum_scale_know_before + 1;
        sum_scale_know_after = sum_scale_know_after + 2;
    }
}

END{
    printf "Before: %.3f\n", sum_scale_know_before/NR;
    printf "After: %.3f\n", sum_scale_know_after/NR;
}
```

- Before we edit the contents of this file, we need to ****merge** this file with the contents of the same file present** in the **feature** branch.
- To merge the contents of both the files, use the following command:

```
git merge feature --allow-unrelated-histories
```



- You will observe, that some conflicts arise when you try to merge as shown in Figure 10. To learn more about this, make sure you have thoroughly gone through the **Week 3 Live Session on Git and GitHub**.

```
erts@erts:~/Documents/gits/git-adventure-kumarkamit$ git merge feature --allow-unrelated-histories
Auto-merging week3_assignment1.awk
CONFLICT (add/add): Merge conflict in week3_assignment1.awk
Automatic merge failed; fix conflicts and then commit the result.
```

Figure 10: Conflicts shown in terminal when trying to merge.

- After you have resolved these conflicts, you can test your code with the following command:

```
bash test_w3_a1.sh week3_assignment1.awk
```



- If your code **passed all the test cases**, you will get the output as shown below:

Running tests...



=====

Evaluating for Test case: 1

Pass => Program executed correctly.

Pass => Output is correct!

Test case number 1: PASS

=====

=====

Evaluating for Test case: 2

Pass => Program executed correctly.

Pass => Output is correct!

Test case number 2: PASS

=====

Congrats! All test cases PASSED.

- If your code **did not pass all the test cases**, you may get the output as shown below:

Running tests...



=====

Evaluating for Test case: 1

Pass => Program executed correctly.

Fail => Expected output is:

Average value of knowledge gained (on a scale of 1 to 10) before and after the task

Before: 2.673

After: 5.925

Percentage increase in knowledge gained from level 2 or less before to level 5 or more

But got the output:

Average value of knowledge gained (on a scale of 1 to 10) before and after the task

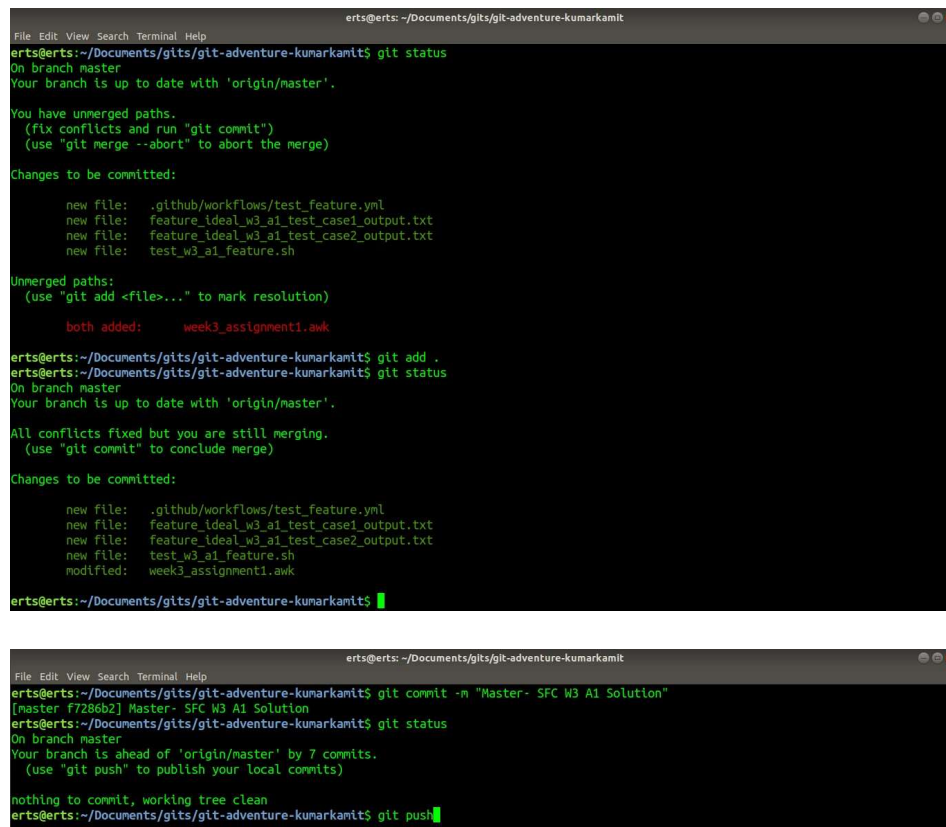
Before: 2.669

After: 5.918

Percentage increase in knowledge gained from level 2 or less before to level 5 or more

Test case number 1: FAIL

- After you are satisfied with your output, follow the steps mentioned above to push your code to GitHub. You can refer Figure 11 for these steps as well.



```

erts@erts: ~/Documents/gits/git-adventure-kumarkamit
File Edit View Search Terminal Help
erts@erts:~/Documents/gits/git-adventure-kumarkamit$ git status
On branch master
Your branch is up to date with 'origin/master'.

You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Changes to be committed:
  new file:   .github/workflows/test_feature.yml
  new file:   feature_ideal_w3_a1_test_case1_output.txt
  new file:   feature_ideal_w3_a1_test_case2_output.txt
  new file:   test_w3_a1_feature.sh

Unmerged paths:
  (use "git add <file>..." to mark resolution)

        both added:   week3_assignment1.awk

erts@erts:~/Documents/gits/git-adventure-kumarkamit$ git add .
erts@erts:~/Documents/gits/git-adventure-kumarkamit$ git status
On branch master
Your branch is up to date with 'origin/master'.

All conflicts fixed but you are still merging.
  (use "git commit" to conclude merge)

Changes to be committed:
  new file:   .github/workflows/test_feature.yml
  new file:   feature_ideal_w3_a1_test_case1_output.txt
  new file:   feature_ideal_w3_a1_test_case2_output.txt
  new file:   test_w3_a1_feature.sh
  modified:   week3_assignment1.awk

erts@erts:~/Documents/gits/git-adventure-kumarkamit$

```

```

erts@erts:~/Documents/gits/git-adventure-kumarkamit
File Edit View Search Terminal Help
erts@erts:~/Documents/gits/git-adventure-kumarkamit$ git commit -m "Master- SFC W3 A1 Solution"
[master f7286b2] Master- SFC W3 A1 Solution
erts@erts:~/Documents/gits/git-adventure-kumarkamit$ git status
On branch master
Your branch is ahead of 'origin/master' by 7 commits.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
erts@erts:~/Documents/gits/git-adventure-kumarkamit$ git push

```

Figure 11 (a) and (b): Git commands required to push code on 'master' branch.

- Visit the **Actions** tab in your Assignment repository to verify the test results of your code.

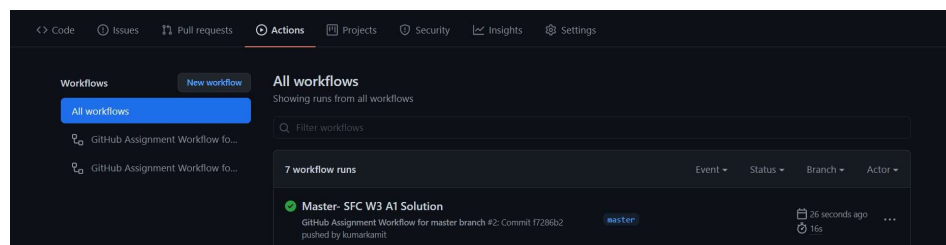


Figure 12: Reviewing the final result of the assignment.

Voila! You have now successfully completed this Assignment.

The deadline of this Assignment is 26th Apr 2021, 1700 Hrs .

The total marks (10) will be updated after the deadline on or before 27th Apr 2021, provided the test cases on both master and feature branches are PASSED .

References

- Nano Editor
 - [How to use Nano Text Editor](#)
 - [Nano Editor Official Docs](#)

- Vim Editor
 - [Interactive Vim Tutorial](#)
 - AWK
 - [Advanced Bash Scripting Guide](#)
 - [AWK: Introduction and Tutorial](#)
 - [Very Useful Command Line Utilities](#)
 - [SED AWK Examples by Unix School](#)
-

All The Best!

