

Question number one

Question: Show and explain, using branches and commits, how a Bash script may create a file for storing temporary output, and how to ensure that this file is removed upon terminating the script.

Answer:

Code:

Here is one way to create a file for storing temporary output. I found following to be working for me:

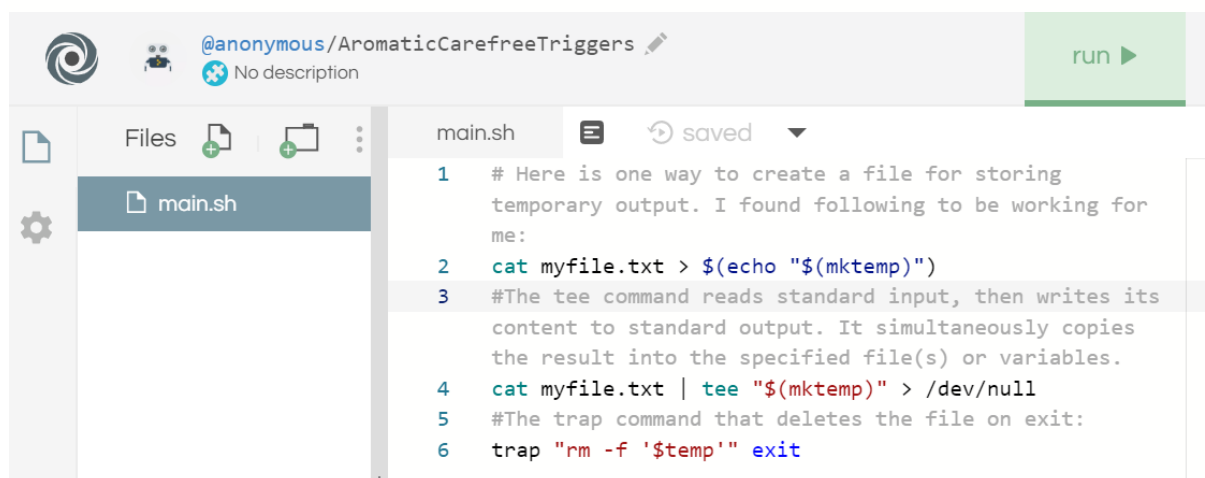
```
cat myfile.txt > $(echo "$(mktemp)")
```

#The tee command reads standard input, then writes its content to standard output. It simultaneously copies the result into the specified file(s) or variables.

```
cat myfile.txt | tee "$(mktemp)" > /dev/null
```

#The trap command that deletes the file on exit:

```
trap "rm -f '$temp'" exit
```

A screenshot of a web-based code editor interface. At the top, there's a header with a logo, a username '@anonymous/AromaticCarefreeTriggers', and a 'run' button. Below the header, there's a sidebar with a 'Files' tab and a file named 'main.sh'. The main area shows the content of 'main.sh' with line numbers 1 through 6. The code is a Bash script that creates a temporary file, writes to it using 'tee', and sets a trap to remove the file on exit.

```
1 # Here is one way to create a file for storing
   temporary output. I found following to be working for
   me:
2 cat myfile.txt > $(echo "$(mktemp)")
3 #The tee command reads standard input, then writes its
   content to standard output. It simultaneously copies
   the result into the specified file(s) or variables.
4 cat myfile.txt | tee "$(mktemp)" > /dev/null
5 #The trap command that deletes the file on exit:
6 trap "rm -f '$temp'" exit
```