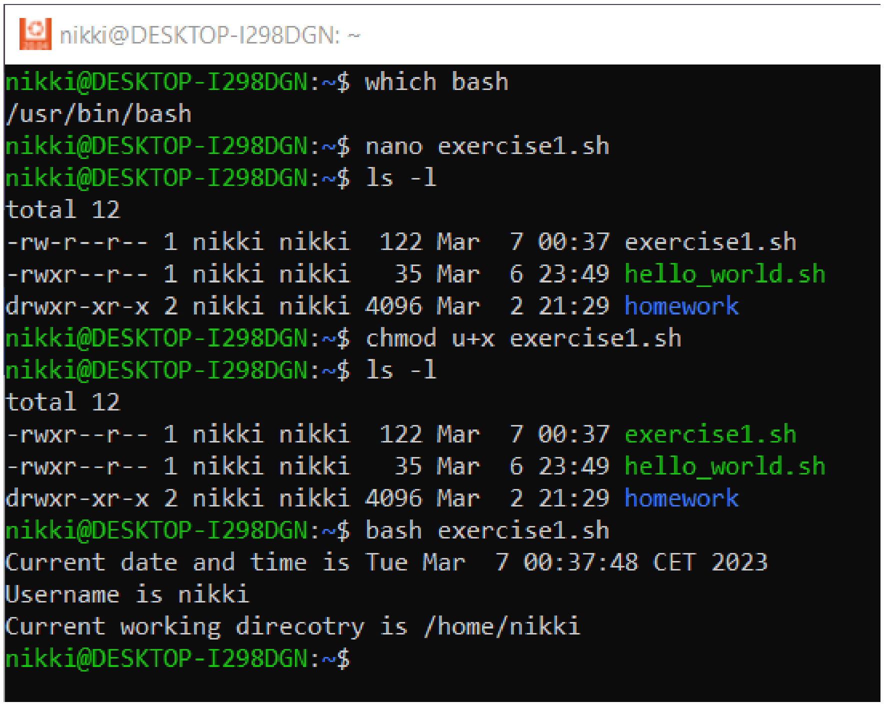
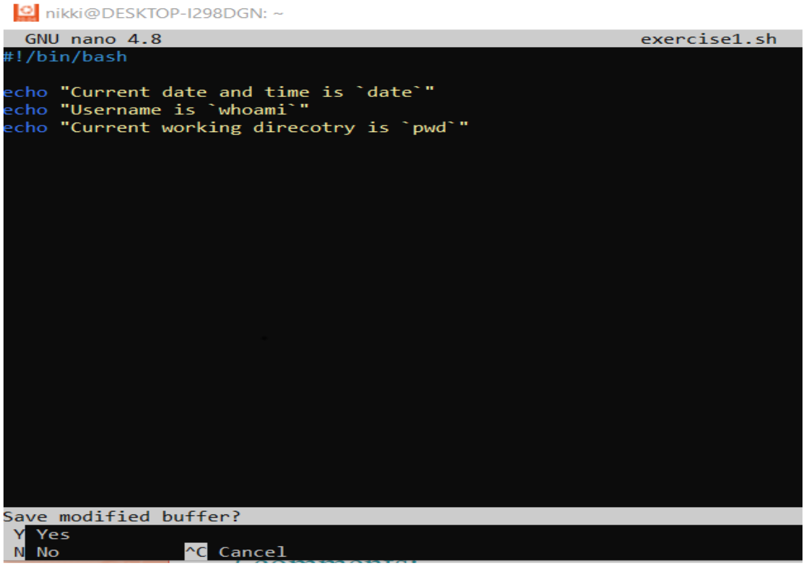
Homework No. 7 – Bash scripts

1. Write a shell script to get the current date, time, username and current working directory.
   - Before I start, I need to find the path to the bash shell. I do this by using which bash. First, I create the script using nano. Then I use ls -l, that lists directory contents. We see that my script doesn't have permission for executing. By using chmod command we give the script executable permissions. Then I run the script using bash exercise1.sh.



```
nikki@DESKTOP-I298DGN:~$ which bash
/usr/bin/bash
nikki@DESKTOP-I298DGN:~$ nano exercise1.sh
nikki@DESKTOP-I298DGN:~$ ls -l
total 12
-rw-r--r-- 1 nikki nikki  122 Mar  7 00:37 exercise1.sh
-rwxr--r-- 1 nikki nikki   35 Mar  6 23:49 hello_world.sh
drwxr-xr-x 2 nikki nikki 4096 Mar  2 21:29 homework
nikki@DESKTOP-I298DGN:~$ chmod u+x exercise1.sh
nikki@DESKTOP-I298DGN:~$ ls -l
total 12
-rwxr--r-- 1 nikki nikki  122 Mar  7 00:37 exercise1.sh
-rwxr--r-- 1 nikki nikki   35 Mar  6 23:49 hello_world.sh
drwxr-xr-x 2 nikki nikki 4096 Mar  2 21:29 homework
nikki@DESKTOP-I298DGN:~$ bash exercise1.sh
Current date and time is Tue Mar  7 00:37:48 CET 2023
Username is nikki
Current working direcotry is /home/nikki
nikki@DESKTOP-I298DGN:~$
```

The bash script:

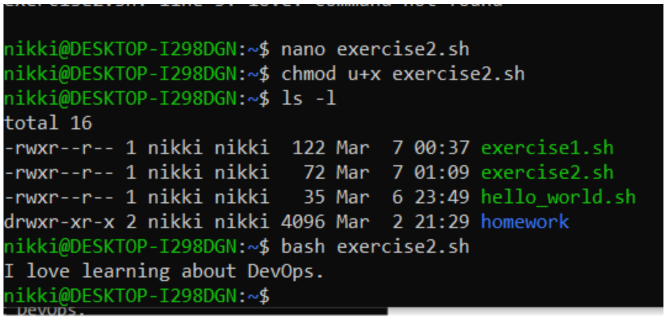


```
#!/bin/bash

echo "Current date and time is `date`"
echo "Username is `whoami`"
echo "Current working direcotry is `pwd`"
```

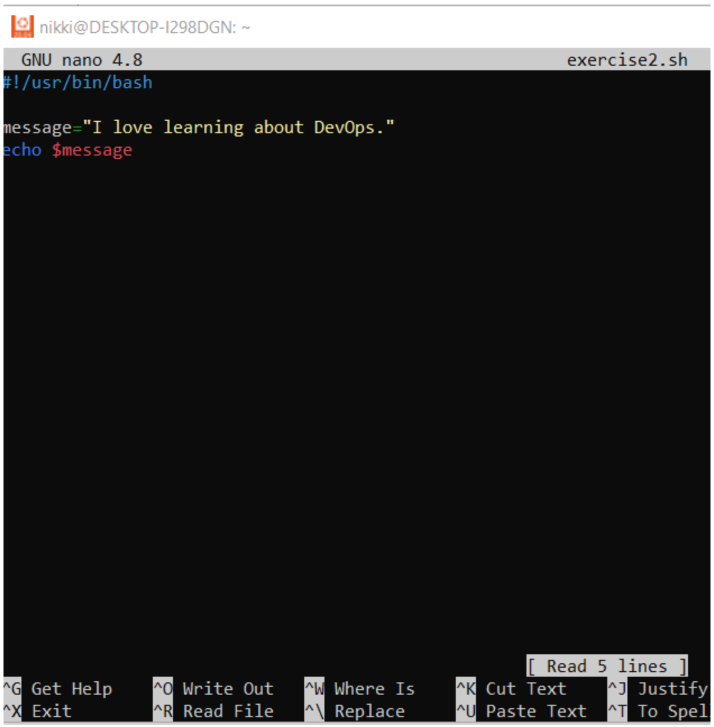
Save modified buffer?

2. Write a shell script that prints "I love learning about DevOps" on the screen. Message should be a variable.
   - First we create a script using nano. Then we give it executable privileges. And at the and we execute it.



```
nikki@DESKTOP-I298DGN:~$ nano exercise2.sh
nikki@DESKTOP-I298DGN:~$ chmod u+x exercise2.sh
nikki@DESKTOP-I298DGN:~$ ls -l
total 16
-rwxr--r-- 1 nikki nikki  122 Mar  7 00:37 exercise1.sh
-rwxr--r-- 1 nikki nikki   72 Mar  7 01:09 exercise2.sh
-rwxr--r-- 1 nikki nikki   35 Mar  6 23:49 hello_world.sh
drwxr-xr-x 2 nikki nikki 4096 Mar  2 21:29 homework
nikki@DESKTOP-I298DGN:~$ bash exercise2.sh
I love learning about DevOps.
nikki@DESKTOP-I298DGN:~$
```

The script:

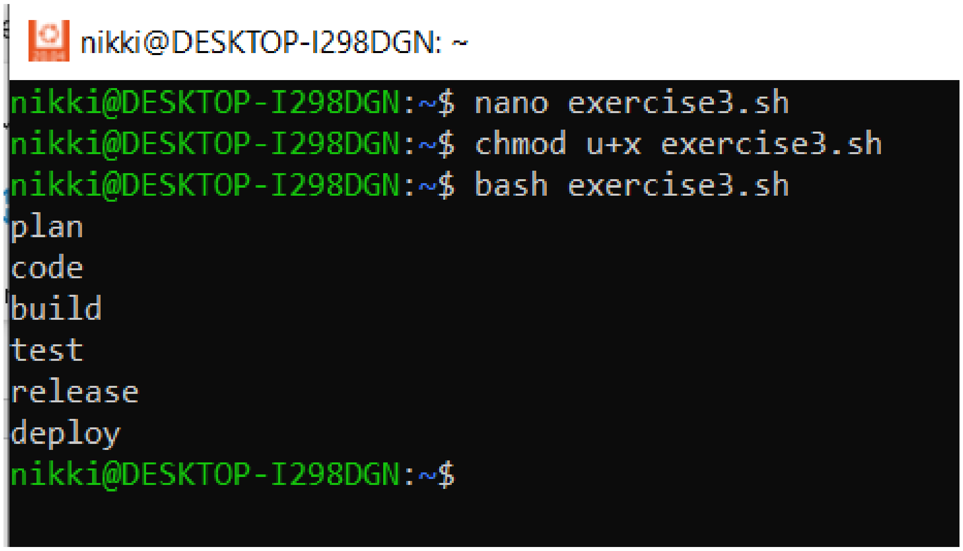


```
#!/usr/bin/bash

message="I love learning about DevOps."
echo $message
```

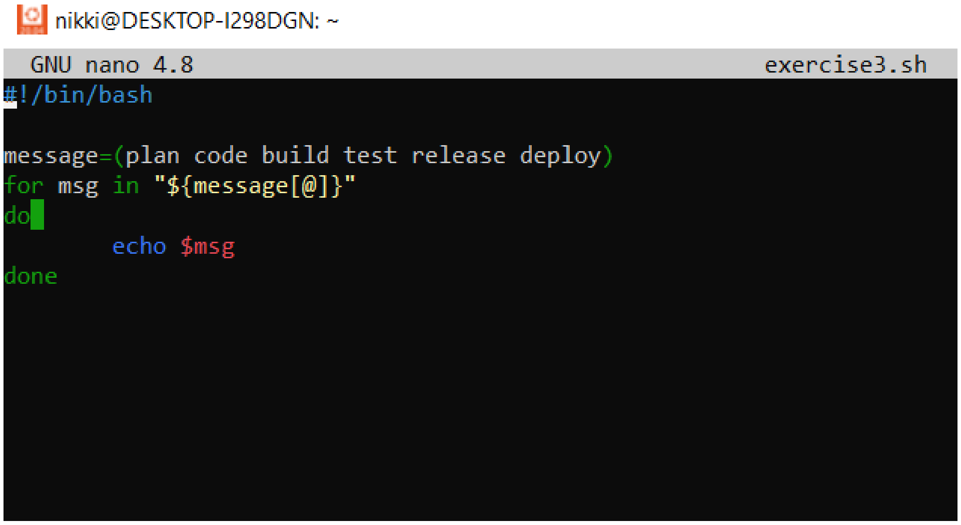
3. Write a shell script that displays “plan code build test release deploy” on the screen with each appearing on a separate line.

   - Again, we create a script by using the nano command. We are using the chmod command to give the script executable permission. By using the bash command, we execute the script.



```
nikki@DESKTOP-I298DGN:~$ nano exercise3.sh
nikki@DESKTOP-I298DGN:~$ chmod u+x exercise3.sh
nikki@DESKTOP-I298DGN:~$ bash exercise3.sh
plan
code
build
test
release
deploy
nikki@DESKTOP-I298DGN:~$
```

The script:



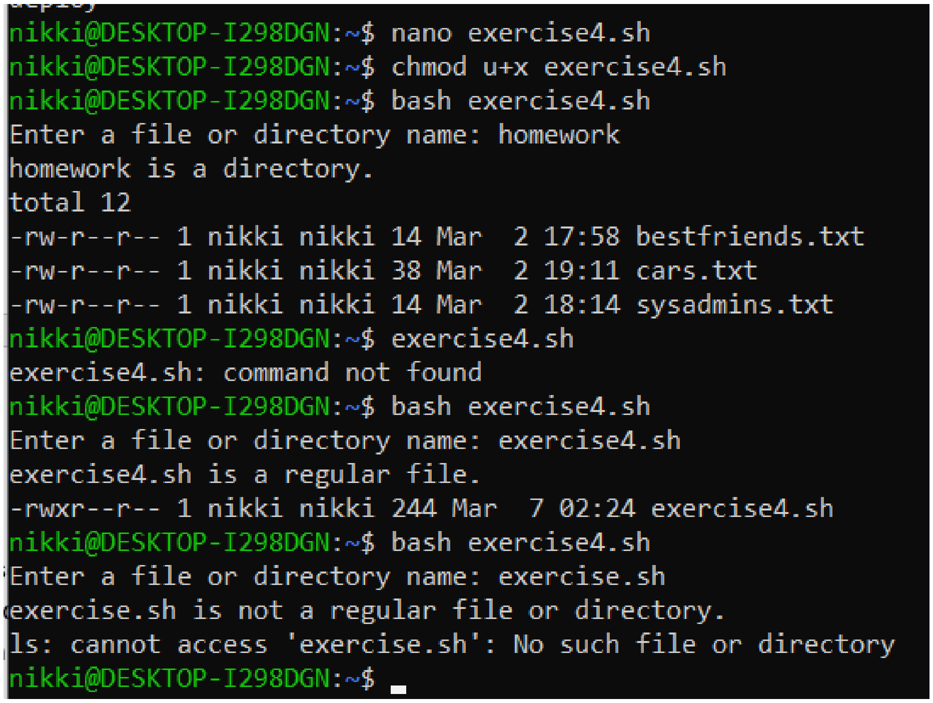
```
  GNU nano 4.8                                        exercise3.sh
#!/bin/bash

message=(plan code build test release deploy)
for msg in "${message[@]}"
do
        echo $msg
done
```

Here we use a loop in order to list the items, every word in new line.

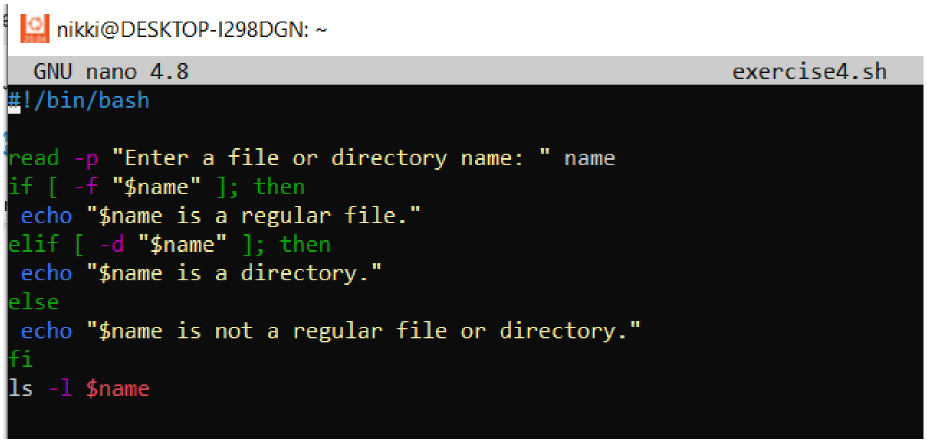
4. Write a shell script that prompts the user for a name of a file or directory and reports if it is a regular file, a directory, or another type of file. Also perform a ls command against the file or directory with the long listing option.

   Again, we start by creating the script. Like before we use the nano command. And we use the chmod to give the file executable permission. By using the bash command, we execute the script.



```
nikki@DESKTOP-I298DGN:~$ nano exercise4.sh
nikki@DESKTOP-I298DGN:~$ chmod u+x exercise4.sh
nikki@DESKTOP-I298DGN:~$ bash exercise4.sh
Enter a file or directory name: homework
homework is a directory.
total 12
-rw-r--r-- 1 nikki nikki 14 Mar  2 17:58 bestfriends.txt
-rw-r--r-- 1 nikki nikki 38 Mar  2 19:11 cars.txt
-rw-r--r-- 1 nikki nikki 14 Mar  2 18:14 sysadmins.txt
nikki@DESKTOP-I298DGN:~$ exercise4.sh
exercise4.sh: command not found
nikki@DESKTOP-I298DGN:~$ bash exercise4.sh
Enter a file or directory name: exercise4.sh
exercise4.sh is a regular file.
-rwxr--r-- 1 nikki nikki 244 Mar  7 02:24 exercise4.sh
nikki@DESKTOP-I298DGN:~$ bash exercise4.sh
Enter a file or directory name: exercise.sh
exercise.sh is not a regular file or directory.
ls: cannot access 'exercise.sh': No such file or directory
nikki@DESKTOP-I298DGN:~$
```

The bash script.



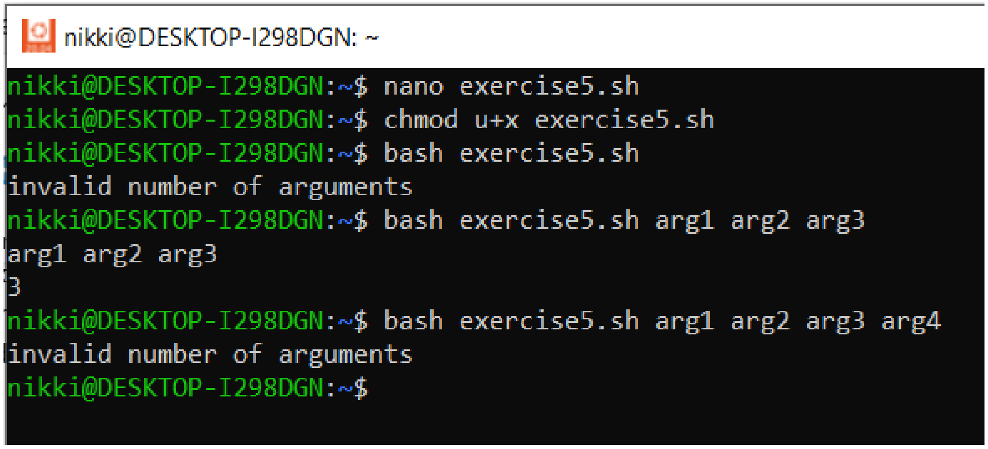
```
#!/bin/bash

read -p "Enter a file or directory name: " name
if [ -f "$name" ]; then
 echo "$name is a regular file."
elif [ -d "$name" ]; then
 echo "$name is a directory."
else
 echo "$name is not a regular file or directory."
fi
ls -l $name
```

The script reads a user input. By using an if statement , we are checking for a regular file, directory or another type of file that is not a regular file/directory. At the and we use the ls -l command and flag to list the files in the directory or the file name.

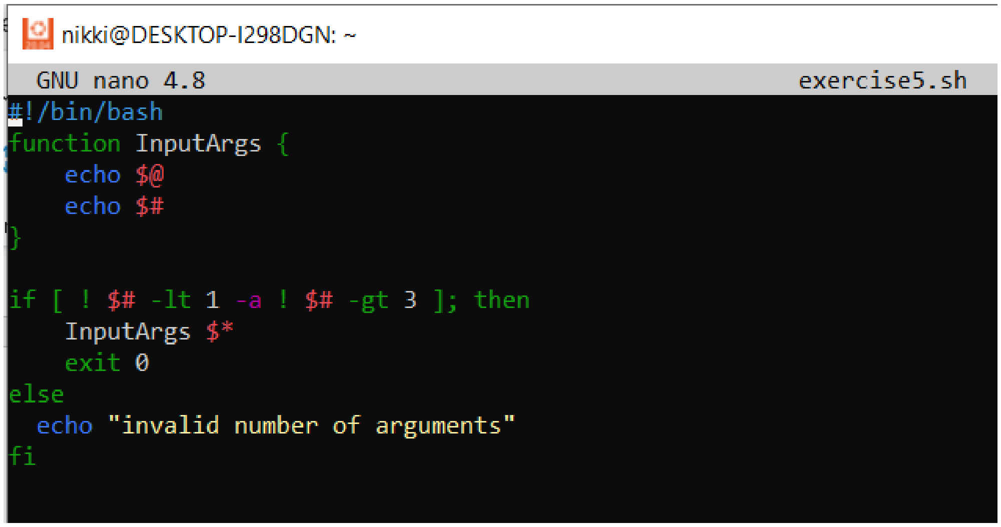
5. Use arguments in a script. Total number of arguments should be three.

- We create the script using nano and give the script executable permissions by using the chmod command. By using the bash command, we execute the script.



```
nikki@DESKTOP-I298DGN:~$ nano exercise5.sh
nikki@DESKTOP-I298DGN:~$ chmod u+x exercise5.sh
nikki@DESKTOP-I298DGN:~$ bash exercise5.sh
invalid number of arguments
nikki@DESKTOP-I298DGN:~$ bash exercise5.sh arg1 arg2 arg3
arg1 arg2 arg3
3
nikki@DESKTOP-I298DGN:~$ bash exercise5.sh arg1 arg2 arg3 arg4
invalid number of arguments
nikki@DESKTOP-I298DGN:~$
```

The bash script.



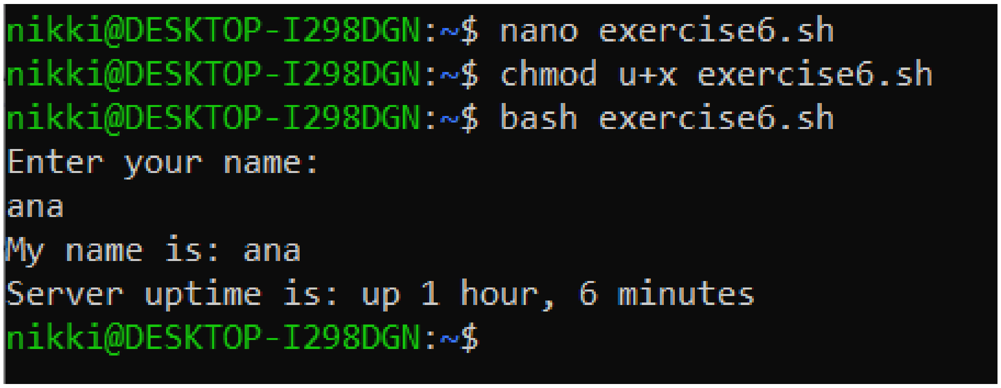
```
  GNU nano 4.8                                          exercise5.sh
#!/bin/bash
function InputArgs {
    echo $@
    echo $#
}

if [ ! $# -lt 1 -a ! $# -gt 3 ]; then
    InputArgs $*
    exit 0
else
  echo "invalid number of arguments"
fi
```

The bash script takes in a number of arguments. Then the arguments are listed to no more than 3, and no less than 1. If one of the conditions are not met, then we will display an error message.

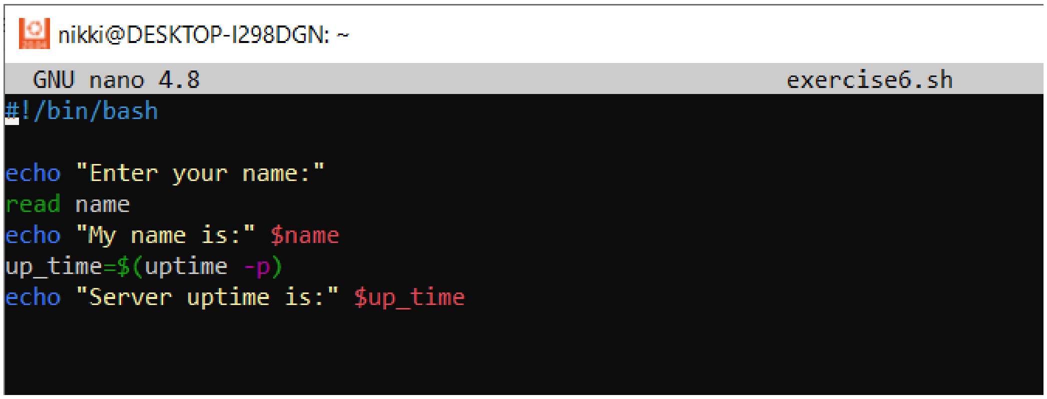
6. Write a script that till output your name out of a variable and will display the server uptime.

- We create the script using nano and give the script executable permissions by using the chmod command. By using the bash command, we execute the script.



```
nikki@DESKTOP-I298DGN:~$ nano exercise6.sh
nikki@DESKTOP-I298DGN:~$ chmod u+x exercise6.sh
nikki@DESKTOP-I298DGN:~$ bash exercise6.sh
Enter your name:
ana
My name is: ana
Server uptime is: up 1 hour, 6 minutes
nikki@DESKTOP-I298DGN:~$
```

- This is the code for the bash script.



```
  GNU nano 4.8                                       exercise6.sh
#!/bin/bash

echo "Enter your name:"
read name
echo "My name is:" $name
up_time=$(uptime -p)
echo "Server uptime is:" $up_time
```

We use the read command in order to have user input. By using the uptime command and the -p argument we get the server uptime. -p argument is used to show only the running time of the system.