**1.Create a shell script to print the HTTP error code of guvi.in & print, the success/failure message based on the error code response.**

**input - Create a shell script to print the HTTP error code of guvi.in**

**STEP 1**

```
[root@ip-172-31-34-112 ec2-user]# touch task4.sh
[root@ip-172-31-34-112 ec2-user]# ls -l
total 24
-rw-r--r--. 1 root     root       0 Sep  3 11:32 KARTHI
-rwxr-xr-x. 1 root     root     613 Sep  3 11:11 elseif.sh
-rwxr-xr-x. 1 root     root     263 Sep  3 11:10 forloop.sh
-rwxr-xr-x. 1 root     root     163 Sep  3 11:09 ifelse.sh
drwxr-xr-x. 2 ec2-user ec2-user  24 Sep  3 13:12 karth
-rwxr-xr-x. 1 root     root     108 Sep  3 12:12 myfun.sh
-rw-r--r--. 1 root     root       0 Sep  5 08:16 task4.sh
-rwxr-xr-x. 1 root     root      96 Sep  3 06:23 touch.sh
-rwxr-xr-x. 1 root     root     111 Sep  3 11:08 whileloop.sh
```

- ➢ **Touch** command is used to create a file
- ➢ **Sudo su** command is used to switch the ec2-user to root user
- ➢ **Ls -l** command is used to list all files & directory
- ➢ **The created file is task4.sh**

**STEP 2**

```
[root@ip-172-31-34-112 ec2-user]# chmod +x task4.sh
[root@ip-172-31-34-112 ec2-user]# ls -l
total 24
-rw-r--r--. 1 root     root       0 Sep  3 11:32 KARTHI
-rwxr-xr-x. 1 root     root     613 Sep  3 11:11 elseif.sh
-rwxr-xr-x. 1 root     root     263 Sep  3 11:10 forloop.sh
-rwxr-xr-x. 1 root     root     163 Sep  3 11:09 ifelse.sh
drwxr-xr-x. 2 ec2-user ec2-user  24 Sep  3 13:12 karth
-rwxr-xr-x. 1 root     root     108 Sep  3 12:12 myfun.sh
-rwxr-xr-x. 1 root     root       0 Sep  5 08:16 task4.sh
-rwxr-xr-x. 1 root     root      96 Sep  3 06:23 touch.sh
-rwxr-xr-x. 1 root     root     111 Sep  3 11:08 whileloop.sh
```

- ➢ **Chmod** command is used to convert read,write (r,w) files to (r,w,x - execute)
- ➢ now ,check task4.sh (highlighted),it shows permission was give to execute the file

**STEP 3**

```
[root@ip-172-31-34-112 ec2-user]# vim task4.sh
```

- ➢ **Vim** command is used to open the file to do scripting

**STEP 4**

```
#!/bin/bash

# URL to check

URL="https://www.guvi.in"
```

- **#! /bin/bash** it is an interpreter to kernel in os which language used
- As per question to check the http error code for the url is **www.guvi.in**

**STEP 5**

```
# Get the HTTP status code

http_status=$(curl -s -o  /dev/null -w "%{http_code}" $URL)
```

**Where,**

- **Curl is the** command is used to intsall, error,etcc,,
- **- s command** is operates to silent and hide process
- **- o /dev/null** command is to discard the output
- **-w "%(http_code)"** is to show the output of http status code
- Http_status take up url indirectly to show the http status code

**STEP 6**

```
# Print the HTTP status code

echo "http status code is $http_status"
```

- **Echo** command is used to print the given line

**STEP 7**

```
if [ "$http_status" -eq 200 ]
then

    echo "URL is SUCCESS"

else

    echo "URL is FAILURE"
fi
```

➢ Using if statement i got the result

**output -Then, print the success/failure message based on the error code response.**

```
[root@ip-172-31-34-112 ec2-user]# ./task4.sh
http status code is 200
URL is SUCCESS
```

**Here,**

➢ **200** command is used as a standard value to find the code status If **value is equal to 200** hence the **url is success, else failure.**

➢ **Answer is success.**

**2.Given a file, replace all occurrence of the word "give" with "learning" from 5th line till the end in only those lines that contain the word "welcome".**

**STEP 1**

```
[ec2-user@ip-172-31-39-229 ~]$ sudo su
[root@ip-172-31-39-229 ec2-user]# mkdir task 4
[root@ip-172-31-39-229 ec2-user]# cd task 4
bash: cd: too many arguments
[root@ip-172-31-39-229 ec2-user]# pwd
/home/ec2-user
[root@ip-172-31-39-229 ec2-user]# mkdir task4
[root@ip-172-31-39-229 ec2-user]# cd task4
[root@ip-172-31-39-229 task4]# pwd
/home/ec2-user/task4
```

➢ **Sudo su** command switch to root user
➢ **Mkdir** command for create a folder
➢ **Cd** command for open a folder
➢ **Pwd** command for to show present working folder

## STEP 2

```
[root@ip-172-31-39-229 task4]# touch file.txt
[root@ip-172-31-39-229 task4]# vim file.txt
[root@ip-172-31-39-229 task4]# cat file.txt
L1 :yesterday i went to movie
L2 :i saw my friend
L3 :we watch movie together
L4 :after that we went to hotel
L5 :she ordered pizaaz and burger
L6 :server welcome and give the order to chef
L7 :then we give gifts and welcome  eachother
L8 :suddenyl her friends came i welocome and give  them also
L9 :i give some instruction to them as an welcome address
L10: at the day end we give bye each other and left and all our welcome goes to another day
```

➢ **Touch** command for create a file
➢ **Vim** command for write a text  in file
➢ **Cat** command for show all the text inside a file
➢ **.txt** command for text file

## STEP 3

```
[root@ip-172-31-39-229 task4]# vim replace.sh
```

➢ **Vim** command for write a text inside a file
➢ **.sh** command for bash script file

## STEP 4

```
#!/bin/bash
input_file="file.txt"
output_file="output.txt"

awk 'NR < 5 { print; next } /welcome/ { gsub( /give/, "learning" ) }1' "$input_file" > "$output_file"

echo "Replacement completed. check the output'$output_file'"
~
```

➢ **#!/bin/bash** command language interpreter kernel to os
➢ **Input_file & output_file** is an key variable

- ➢ **“File.txt”** is an value which already created as 10 line statement
- ➢ **“Output.txt”** is an value which our output should present here it will automatiucaly created a file when it bring output
- ➢ Echo command for print a text in ne file
- ➢ **Awk ‘NR < 5 { print;next }** is for packet search. Replace and process
- ➢ **/welcome/ { gsub ( /give/, “learning” ) }** it will check all lines where both welcome and give present in same line then it replace give with learning **(replacement,string.var)**
- ➢ **1’“$input_file” > “$output_file”** 1 is for print command in awk ls to print output file from input file

**STEP 5**

```
[root@ip-172-31-39-229 task4]# ./replace.sh
Replacement completed. check the output'output.txt'
```

- ➢ **./replace.sh** command used for execute a output

**STEP 6**

```
[root@ip-172-31-39-229 task4]# ls -l
total 12
-rw-r--r--. 1 root root 448 Sep  5 13:19 file.txt
-rw-r--r--. 1 root root 464 Sep  6 04:53 output.txt
-rwxr-xr-x. 1 root root 224 Sep  6 04:53 replace.sh
```

- ➢ **Ls -l** command is to check for new and old files and folder but now we inside the folder so its shows only the files present inside folder

**OUTPUT**

**As per question, replace all occurrence of the word "give" with "learning" from 5th line till the end in only those lines that contain the word "welcome"**

**Using while statement** i did a **bash script** using the following command for replace a word **learning** in place of **give** from **5th line** wherever the **give and welcome** should present in **same line**

```
L1 :yesterday i went to movie
L2 :i saw my friend
L3 :we watch movie together
L4 :after that we went to hotel
L5 :she ordered pizaaz and burger
L6 :server welcome and learning the order to chef
L7 :then we learning gifts and welcome  eachother
L8 :suddenyl her friends came i welocome and give  them also
L9 :i learning some instruction to them as an welcome address
L10: at the day end we learning bye each other and left and all our welcome goes to another day

~
```