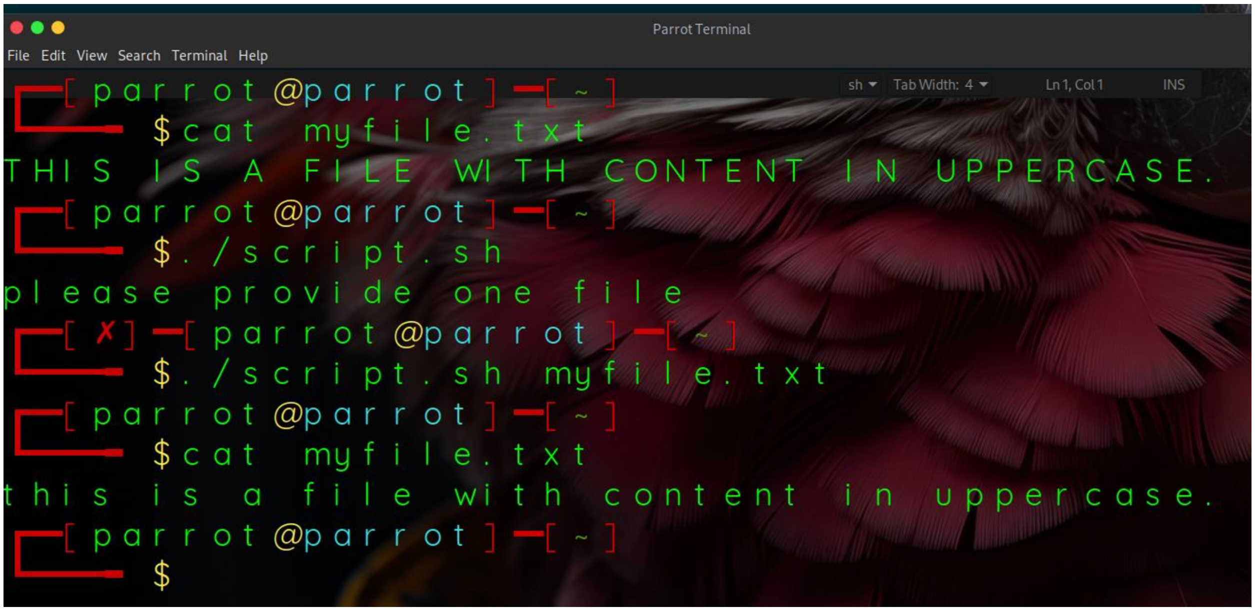
ASSINGNMENT 05 | AP22110010245

1. Write a shell script to translate all the characters to lower case in a given text file.

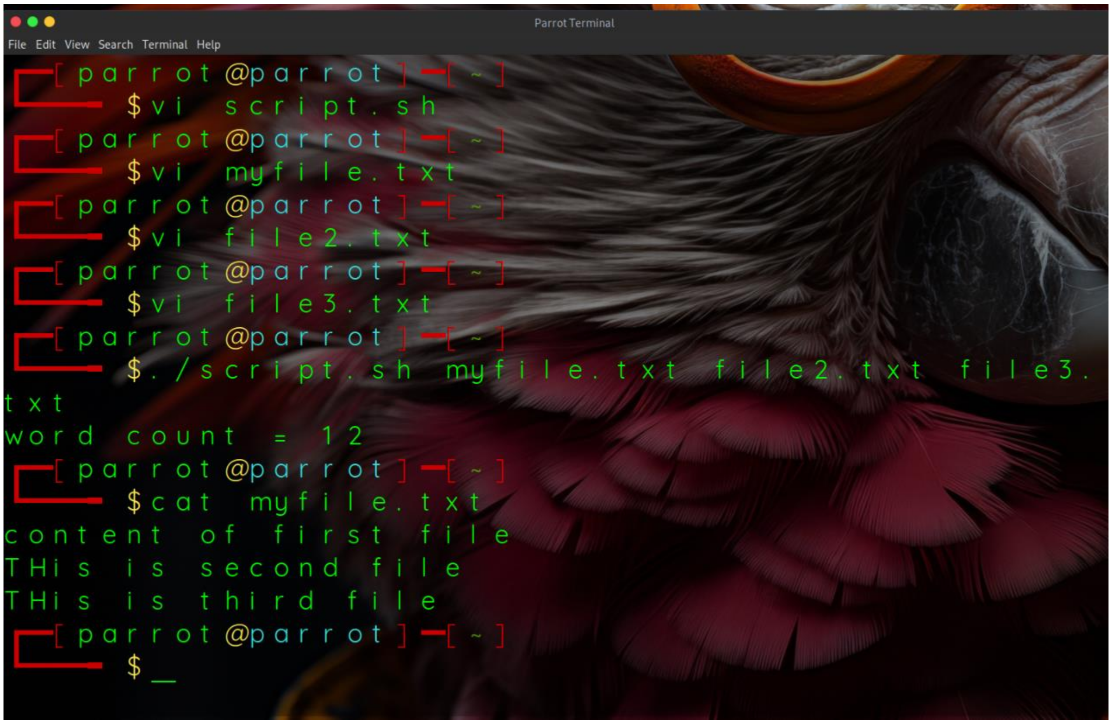
```
if [ $# -ne 1 ]; then
  echo "please provide one file"
  exit 1
fi

file=$1
content=$(cat "$file")
lowercase_content=$(echo "$content" | tr '[:upper:]' '[:lower:]')
echo "$lowercase_content" > myfile.txt
```



```
┌─[parrot@parrot]─[~]
└────  $cat myfile.txt
THIS IS A FILE WITH CONTENT IN UPPERCASE.
┌─[parrot@parrot]─[~]
└────  $./script.sh
please provide one file
┌─[✗]─[parrot@parrot]─[~]
└────  $./script.sh myfile.txt
┌─[parrot@parrot]─[~]
└────  $cat myfile.txt
this is a file with content in uppercase.
┌─[parrot@parrot]─[~]
└────  $
```

2. Write a shell script to combine any three text files into a single file (append them in the order as they appear in the arguments) and display the word count.

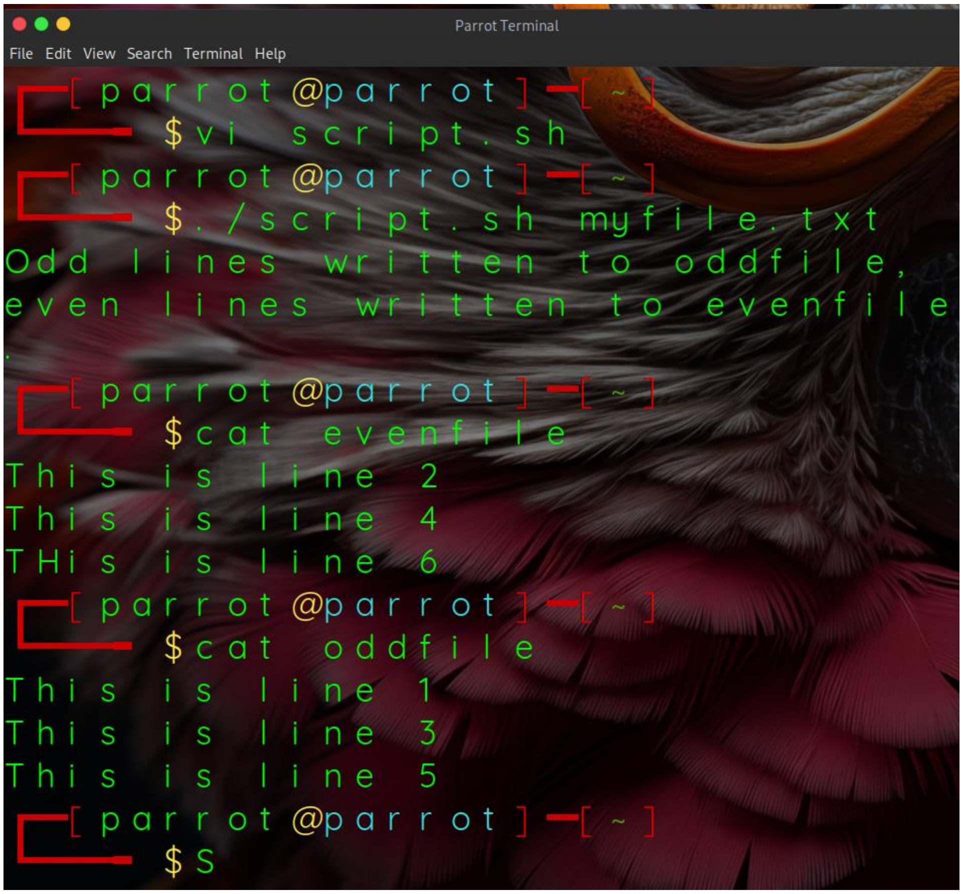
```
if [ $# -ne 3 ]; then
  echo "please provide three files"
  exit 1
fi

file1=$1
file2=$2
file3=$3
content2=$(cat "$file2")
content3=$(cat "$file3")

echo "$content2" >> $file1
echo "$content3" >> $file1
count=$(wc -w < "$file1")
echo "word count = $count"
```



```
┌─[parrot@parrot]─[~]
└──╼ $vi script.sh
┌─[parrot@parrot]─[~]
└──╼ $vi myfile.txt
┌─[parrot@parrot]─[~]
└──╼ $vi file2.txt
┌─[parrot@parrot]─[~]
└──╼ $vi file3.txt
┌─[parrot@parrot]─[~]
└──╼ $./script.sh myfile.txt file2.txt file3.
txt
word count = 12
┌─[parrot@parrot]─[~]
└──╼ $cat myfile.txt
content of first file
THis is second file
THis is third file
┌─[parrot@parrot]─[~]
└──╼ $_
```

3. Write a shell script that, given a file name as the argument will write the even numbered line to a file with name even-file and odd numbered lines to a file called odd-file.

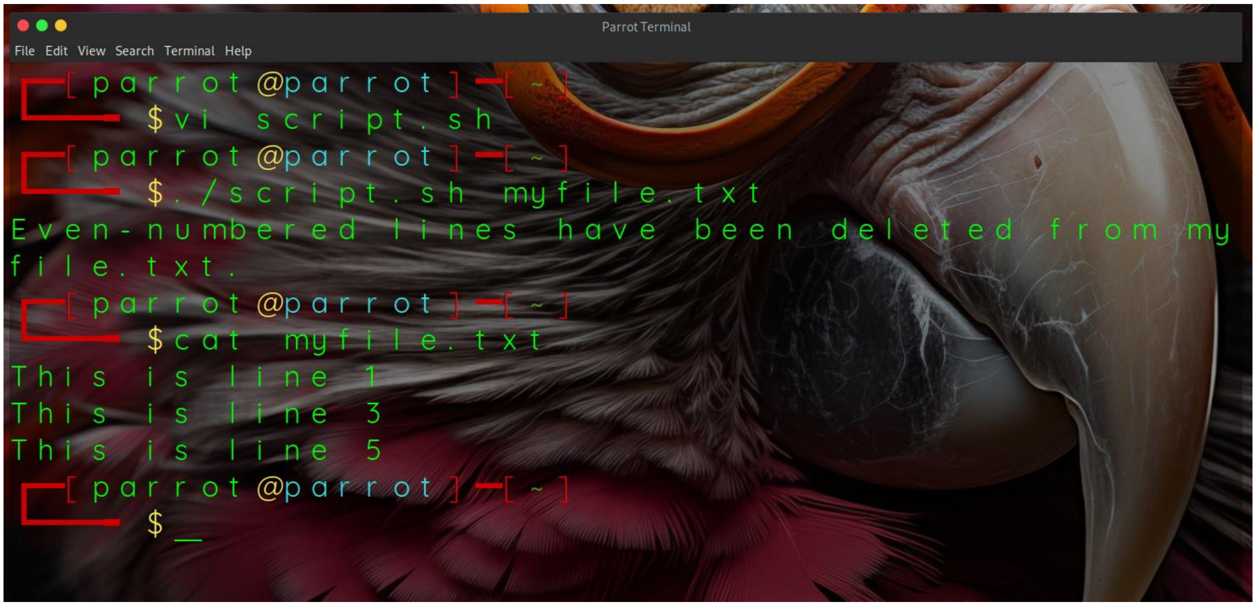
```
if [ $# -ne 1 ]; then
  echo "please provide one file"
  exit 1
fi
file=$1
# Odd lines to oddfile
sed -n '1~2p' "$file" > oddfile
# Even lines to evenfile
sed -n '2~2p' "$file" > evenfile
echo "Odd lines written to oddfile, even lines written to evenfile."
```



```
┌─[parrot@parrot]─[~]
└──$vi script.sh
┌─[parrot@parrot]─[~]
└──$./script.sh myfile.txt
Odd lines written to oddfile,
even lines written to evenfile
.
┌─[parrot@parrot]─[~]
└──$cat evenfile
This is line 2
This is line 4
THis is line 6
┌─[parrot@parrot]─[~]
└──$cat oddfile
This is line 1
This is line 3
This is line 5
┌─[parrot@parrot]─[~]
└──$S
```

4. Write a shell script which deletes all the even numbered lines in a text file.

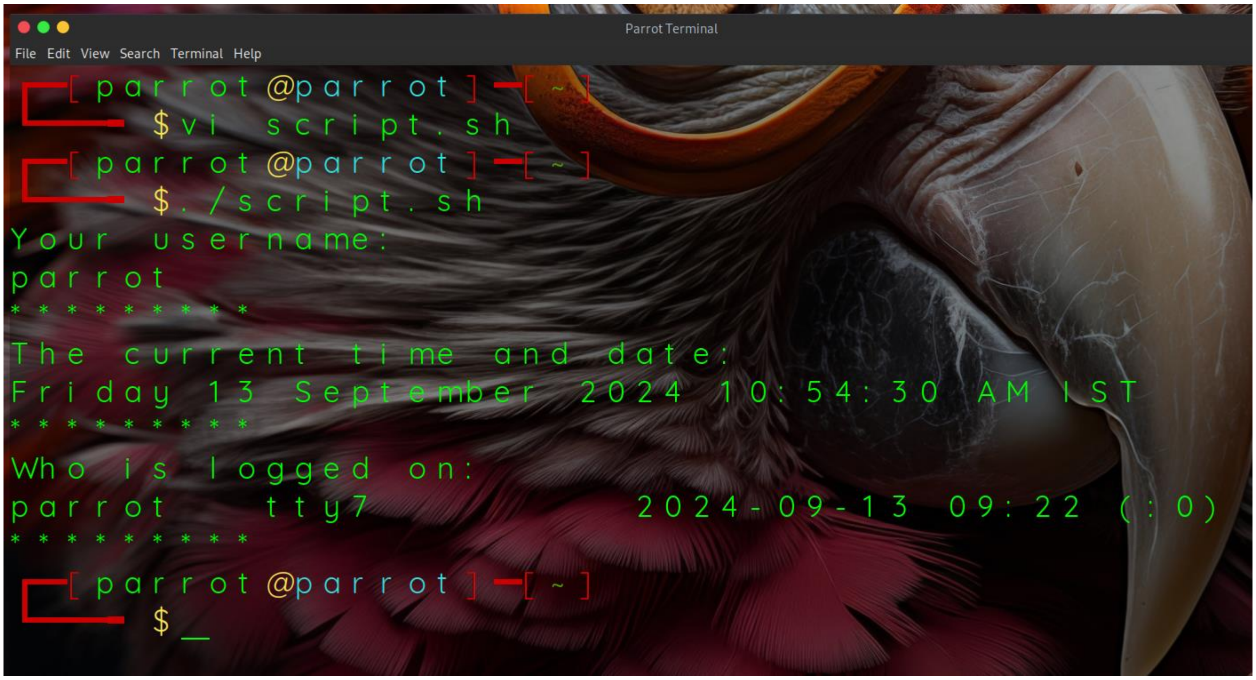
```
#!/bin/bash
if [ $# -ne 1 ]; then
  echo "Please provide a file name"
  exit 1
fi
file=$1
sed -i '2~2d' "$file"
echo "Even-numbered lines have been deleted from $file."
```



```
┌─[parrot@parrot]─[~]
└──╼ $vi script.sh
┌─[parrot@parrot]─[~]
└──╼ $./script.sh myfile.txt
Even-numbered lines have been deleted from my
file.txt.
┌─[parrot@parrot]─[~]
└──╼ $cat myfile.txt
This is line 1
This is line 3
This is line 5
┌─[parrot@parrot]─[~]
└──╼ $_
```

5. Write a script called hello which outputs the following: • your username • the time and date • who is logged on • also output a line of asterices (*********) after each section.

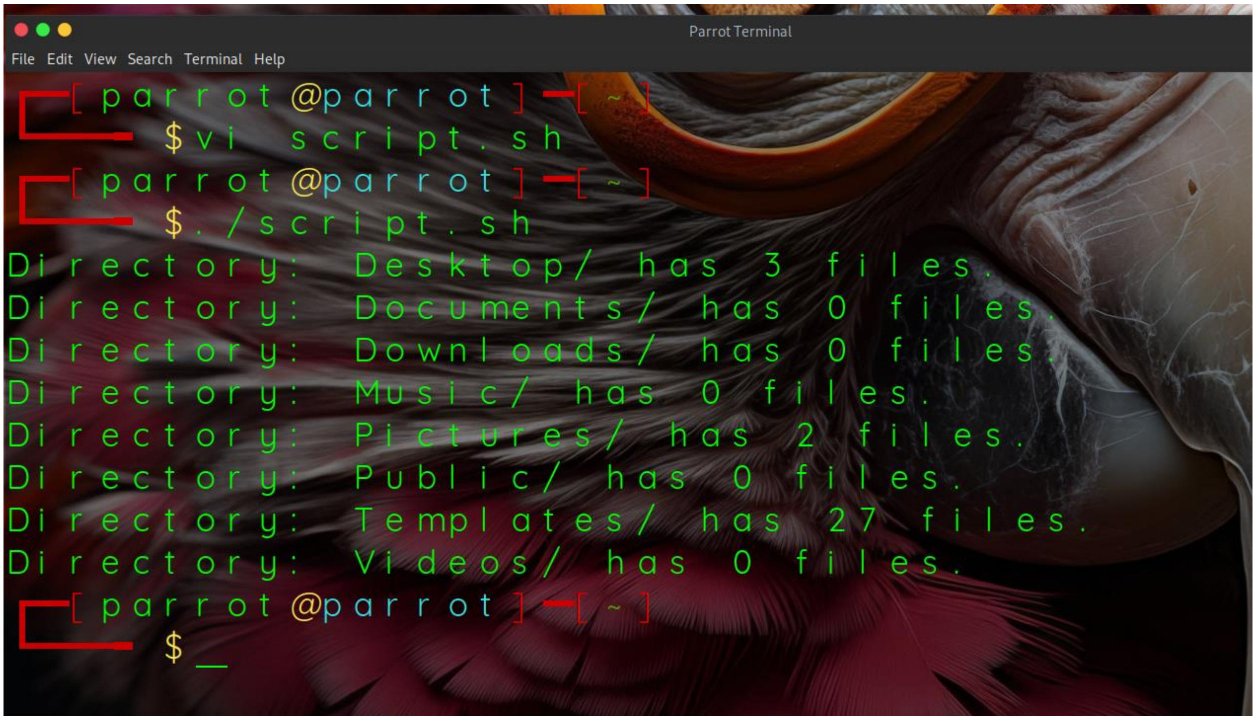
```
#!/bin/bash
echo "Your username:"
echo "$USER"
echo "*********"
echo "The current time and date:"
date
echo "*********"
echo "Who is logged on:"
who
echo "*********"
```



```
┌─[parrot@parrot]─[~]
└──── $vi  script.sh
┌─[parrot@parrot]─[~]
└──── $./script.sh
Your username:
parrot
*********
The current time and date:
Friday 13 September 2024 10:54:30 AM IST
*********
Who is logged on:
parrot   tty7          2024-09-13 09:22 (:0)
*********
┌─[parrot@parrot]─[~]
└──── $_
```

6. Write a script that will count the number of files in each of your subdirectories.

```
#!/bin/bash
for dir in */; do
  count=$(find "$dir" -type f | wc -l)
  echo "Directory: $dir has $count files."
Done
```

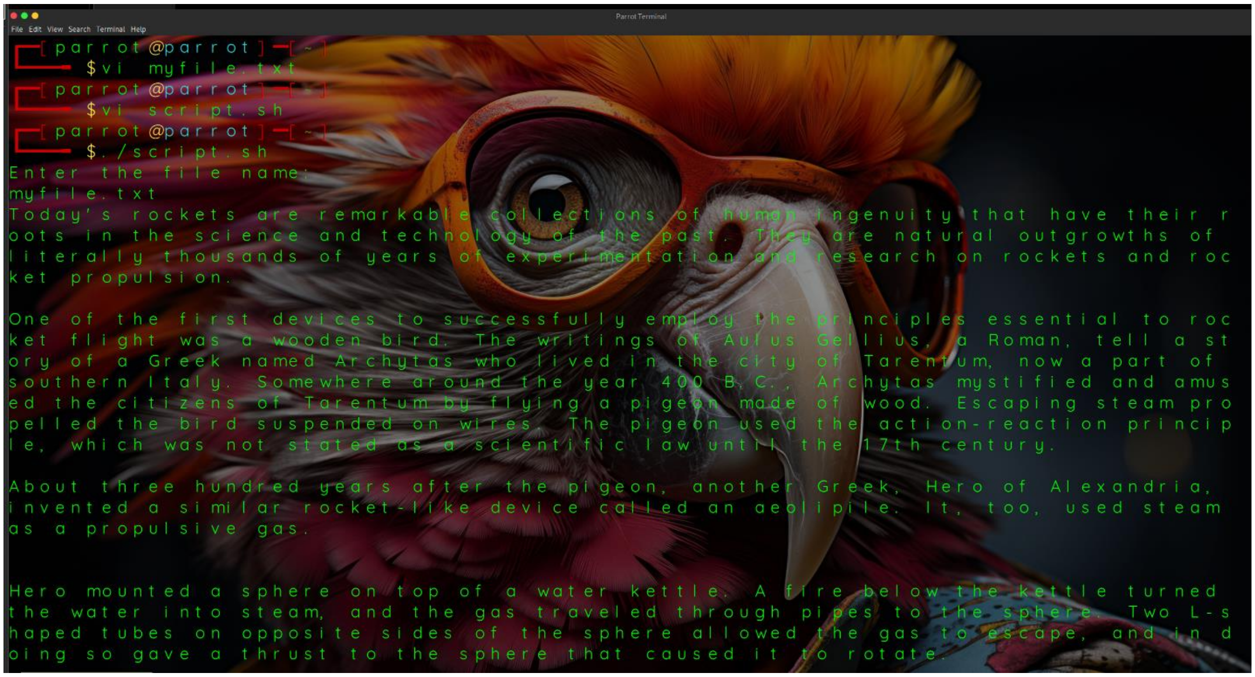


```
┌─[parrot@parrot]─[~]
└──── $vi script.sh
┌─[parrot@parrot]─[~]
└──── $./script.sh
Directory: Desktop/ has 3 files.
Directory: Documents/ has 0 files.
Directory: Downloads/ has 0 files.
Directory: Music/ has 0 files.
Directory: Pictures/ has 2 files.
Directory: Public/ has 0 files.
Directory: Templates/ has 27 files.
Directory: Videos/ has 0 files.
┌─[parrot@parrot]─[~]
└──── $_
```

7. Write a shell script like a more command. It asks the user name, the name of the file on command prompt and displays only the 15 lines of the file at a time on the screen. Further, next 15 lines will be displayed only when the user presses the enter key / any other key.

```
#!/bin/bash
echo "Enter the file name:"
read filename
if [ ! -f "$filename" ]; then
  echo "File not found!"
  exit 1
fi
# Display the file content 15 lines at a time
lines_per_page=15
```

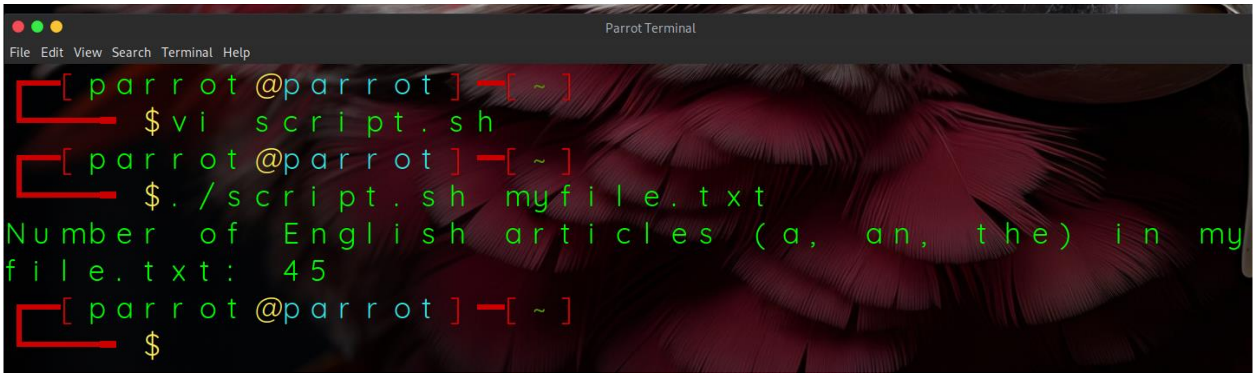
```
total_lines=$(wc -l < "$filename")
start=1
while [ $start -le $total_lines ]; do
  end=$((start + lines_per_page - 1))
  sed -n "${start},${end}p" "$filename"
  start=$((end + 1))
  if [ $start -le $total_lines ]; then
    read -p "Press Enter to continue..."
  fi
done
```



```
┌─[parrot@parrot]─[~]
└──╼ $vi myfile.txt
┌─[parrot@parrot]─[~]
└──╼ $vi script.sh
┌─[parrot@parrot]─[~]
└──╼ $./script.sh
Enter the file name:
myfile.txt
Today's rockets are remarkable collections of human ingenuity that have their r
oots in the science and technology of the past. They are natural outgrowths of
literally thousands of years of experimentation and research on rockets and roc
ket propulsion.

One of the first devices to successfully employ the principles essential to roc
ket flight was a wooden bird. The writings of Aulus Gellius, a Roman, tell a st
ory of a Greek named Archytas who lived in the city of Tarentum, now a part of
southern Italy. Somewhere around the year 400 B.C., Archytas mystified and amus
ed the citizens of Tarentum by flying a pigeon made of wood. Escaping steam pro
pelled the bird suspended on wires. The pigeon used the action-reaction princip
le, which was not stated as a scientific law until the 17th century.

About three hundred years after the pigeon, another Greek, Hero of Alexandria,
invented a similar rocket-like device called an aeolipile. It, too, used steam
as a propulsive gas.


Hero mounted a sphere on top of a water kettle. A fire below the kettle turned
the water into steam, and the gas traveled through pipes to the sphere. Two L-s
haped tubes on opposite sides of the sphere allowed the gas to escape, and in d
oing so gave a thrust to the sphere that caused it to rotate.
```

8. Write a shell script that counts English language articles (a, an, the) in a given text file.

```
#!/bin/bash
if [ $# -ne 1 ]; then
  echo "Please provide a file name"
  exit 1
fi
file=$1
articles=$(grep -o -i -w '\ba\b\|\ban\b\|\bthe\b' "$file" | wc -l)
echo "Number of English articles (a, an, the) in $file: $articles"
```

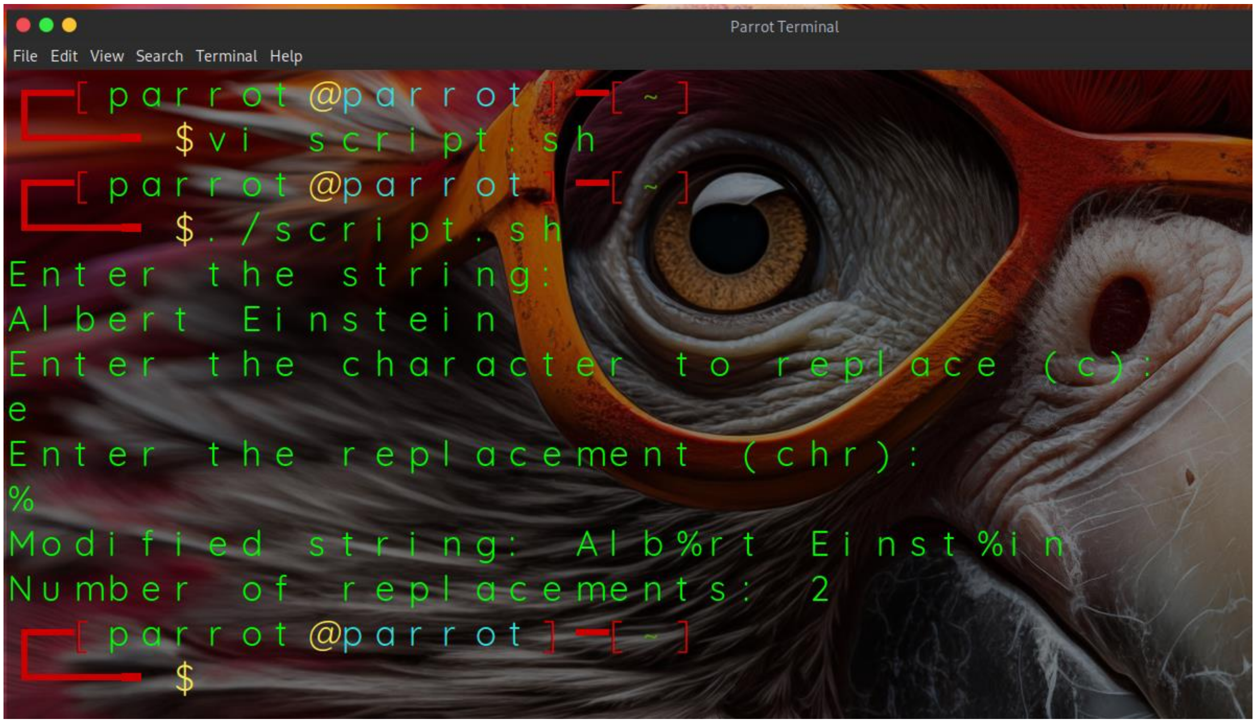


```
┌─[parrot@parrot]─[~]
└──╼ $vi script.sh
┌─[parrot@parrot]─[~]
└──╼ $./script.sh myfile.txt
Number of English articles (a, an, the) in my
file.txt: 45
┌─[parrot@parrot]─[~]
└──╼ $
```

9. Write the shell script which will replace each occurrence of character c with the characters chr in a string s. It should also display the number of replacements.

```
#!/bin/bash

echo "Enter the string:"
read s
echo "Enter the character to replace (c):"
read c
echo "Enter the replacement (chr):"
read chr
```

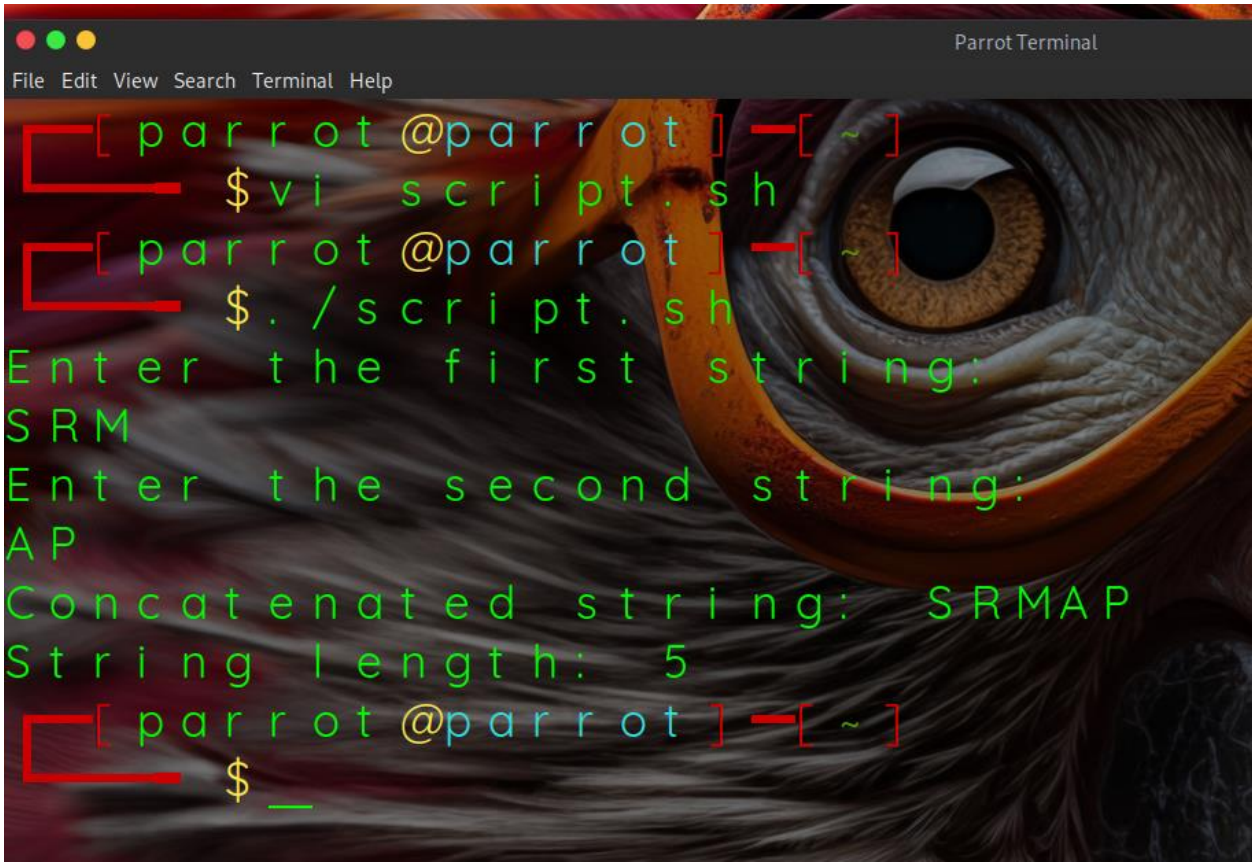
```
# Count occurrences of 'c' before replacement
count=$(echo "$s" | grep -o "$c" | wc -l)

# Replace occurrences
new_string=$(echo "$s" | sed "s/$c/$chr/g")

echo "Modified string: $new_string"
echo "Number of replacements: $count"
```



```
┌─[parrot@parrot]─[~]
└──── $vi script.sh
┌─[parrot@parrot]─[~]
└──── $./script.sh
Enter the string:
Albert Einstein
Enter the character to replace (c):
e
Enter the replacement (chr):
%
Modified string: Alb%rt Einst%in
Number of replacements: 2
┌─[parrot@parrot]─[~]
└──── $
```

10. Write a shell program to concatenate to two strings given as input and display the resultant string along with its string length.

```
#!/bin/bash

echo "Enter the first string:"
read str1
echo "Enter the second string:"
read str2

# Concatenate the strings
result="$str1$str2"

# Calculate the length of the resultant string
length=${#result}

echo "Concatenated string: $result"
echo "String length: $length"
```



```
┌─[parrot@parrot]─[~]
└──── $vi script.sh
┌─[parrot@parrot]─[~]
└──── $./script.sh
Enter the first string:
SRM
Enter the second string:
AP
Concatenated string: SRMAP
String length: 5
┌─[parrot@parrot]─[~]
└──── $
```