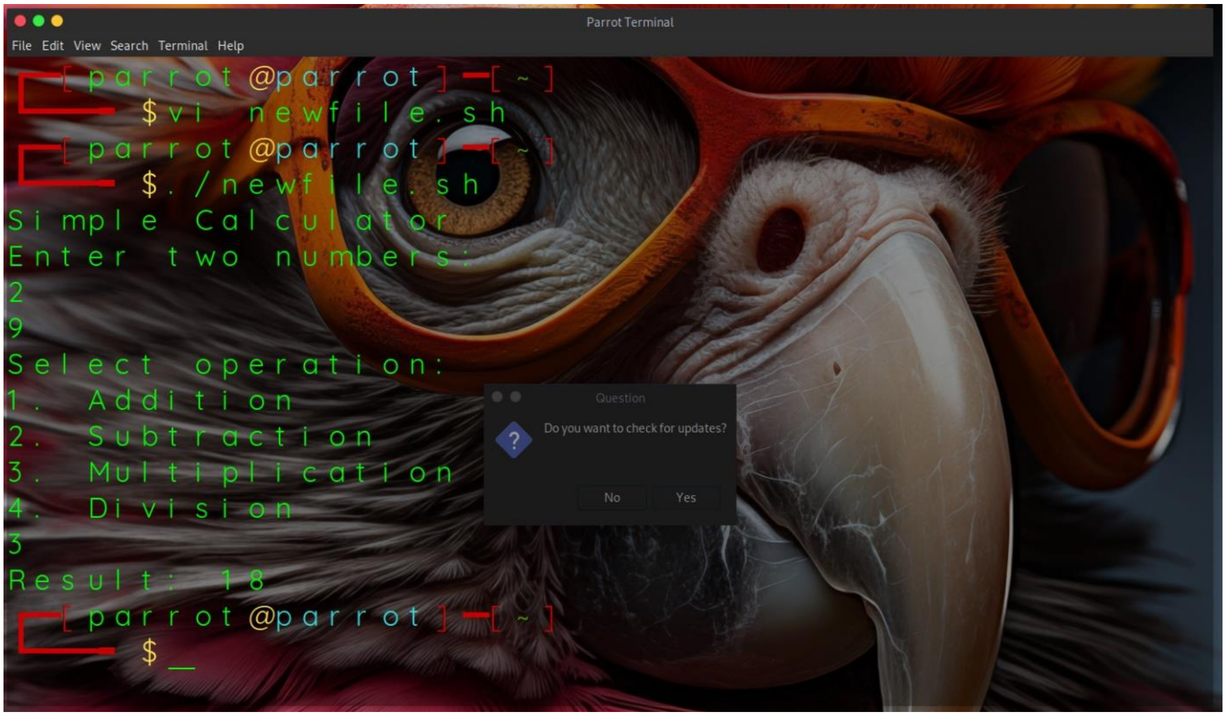
1. Write a shell program to simulate a simple calculator.

```
#!/bin/bash
echo "Simple Calculator"
echo "Enter two numbers:"
read num1
read num2

echo "Select operation:"
echo "1. Addition"
echo "2. Subtraction"
echo "3. Multiplication"
echo "4. Division"
read choice

case $choice in
  1) result=$(echo "$num1 + $num2" | bc)
     echo "Result: $result";;
  2) result=$(echo "$num1 - $num2" | bc)
     echo "Result: $result";;
  3) result=$(echo "$num1 * $num2" | bc)
     echo "Result: $result";;
  4) result=$(echo "scale=2; $num1 / $num2" | bc)
     echo "Result: $result";;
  *) echo "Invalid choice";;
Esac
```

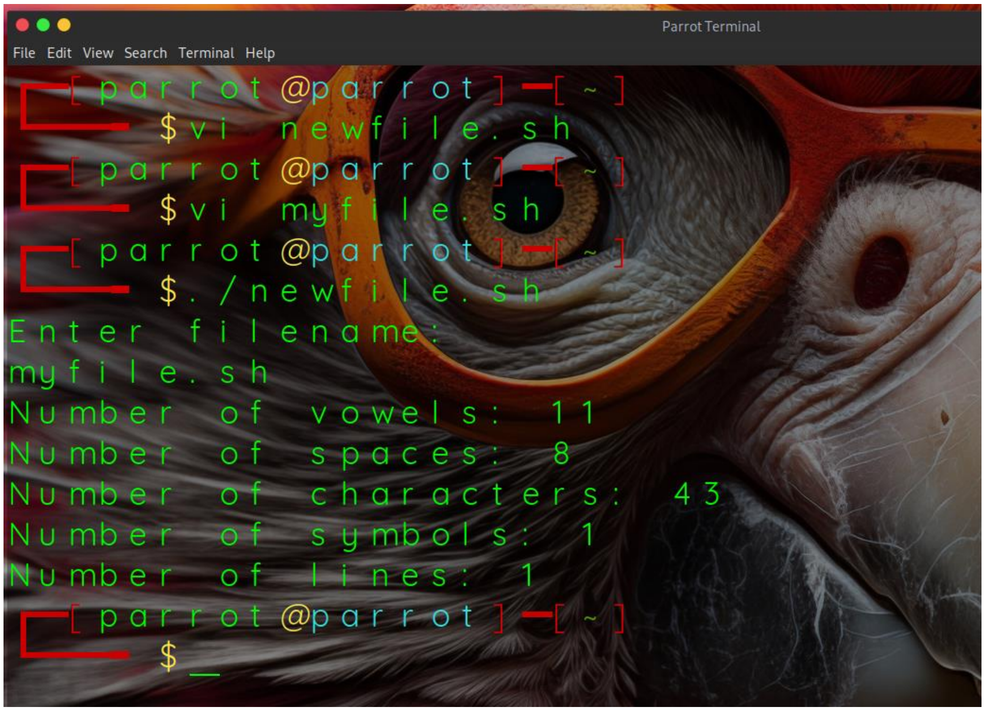


```
┌─[parrot@parrot]─[~]
└──╼ $vi newfile.sh
┌─[parrot@parrot]─[~]
└──╼ $./newfile.sh
Simple Calculator
Enter two numbers:
2
9
Select operation:
1. Addition
2. Subtraction
3. Multiplication
4. Division
3
Result: 18
┌─[parrot@parrot]─[~]
└──╼ $_
```

Do you want to check for updates?

2. Write a shell program to count the following in a text file.
   - Number of vowels in a given text file.
   - Number of blank spaces.
   - Number of characters.
   - Number of symbols.
   - Number of lines

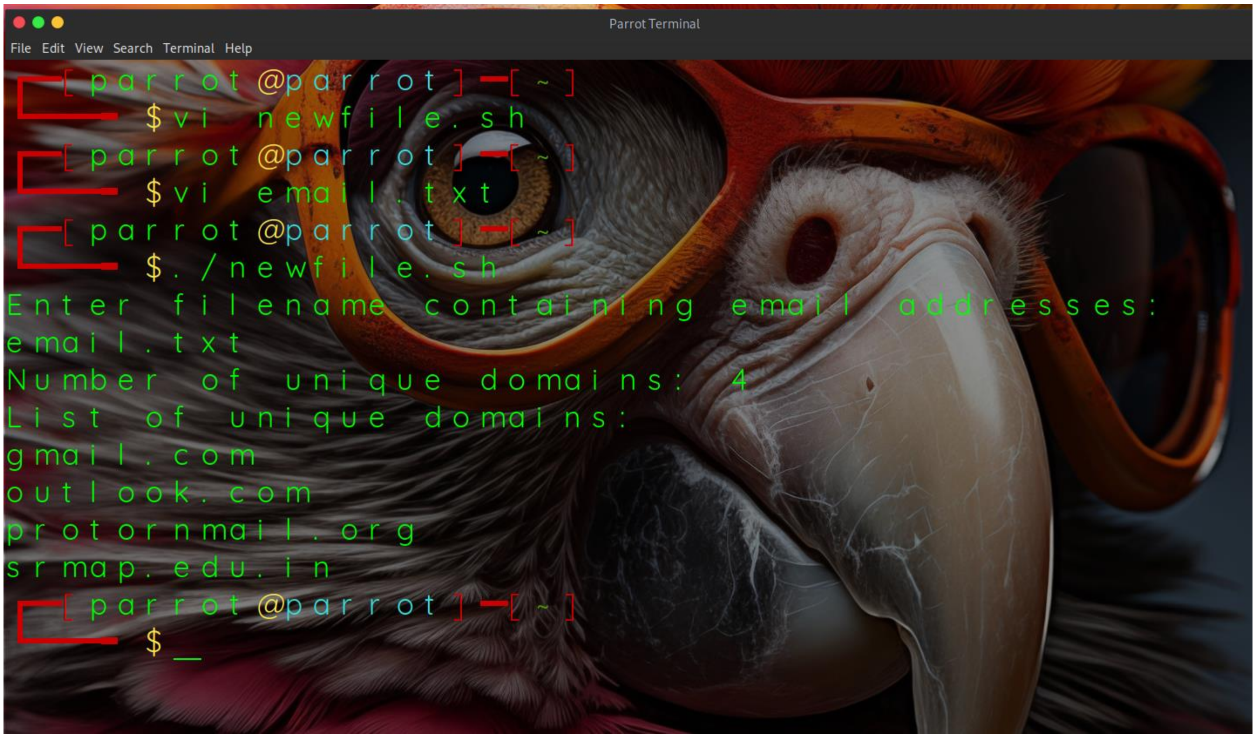
```
#!/bin/bash

echo "Enter filename:"
read filename

if [ ! -f "$filename" ]; then
   echo "File not found!"
   exit 1
fi

vowels=$(tr -cd 'aeiouAEIOU' < "$filename" | wc -c)
spaces=$(grep -o ' ' "$filename" | wc -l)
characters=$(wc -m < "$filename")
symbols=$(grep -o '[[:punct:]]' "$filename" | wc -l)
lines=$(wc -l < "$filename")

echo "Number of vowels: $vowels"
echo "Number of spaces: $spaces"
echo "Number of characters: $characters"
echo "Number of symbols: $symbols"
echo "Number of lines: $lines"
```



```
┌─[parrot@parrot]─[~]
└────$vi newfile.sh
┌─[parrot@parrot]─[~]
└────$vi myfile.sh
┌─[parrot@parrot]─[~]
└────$./newfile.sh
Enter filename:
myfile.sh
Number of vowels: 11
Number of spaces: 8
Number of characters: 43
Number of symbols: 1
Number of lines: 1
┌─[parrot@parrot]─[~]
└────$_
```

3. Write a shell script that reads a file containing a list of email addresses (one per line) and counts how many unique domains exist in the file.

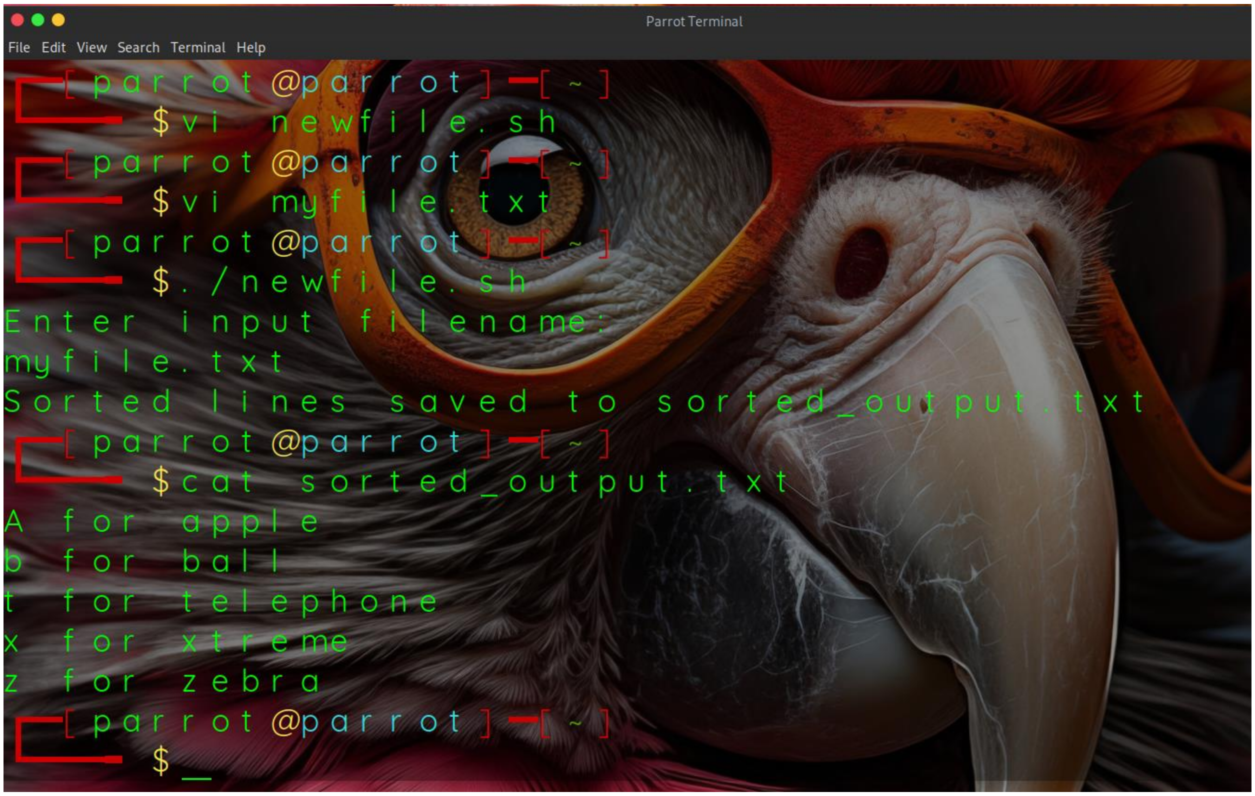
```
#!/bin/bash
echo "Enter filename containing email addresses:"
read filename

if [ ! -f "$filename" ]; then
   echo "File not found!"
   exit 1
fi
domains=$(grep -oE '[[:alnum:]_.-]+@[[:alnum:]_.-]+' "$filename" | cut -d '@' -f 2 | sort -u)
num_domains=$(echo "$domains" | wc -l)

echo "Number of unique domains: $num_domains"
echo "List of unique domains:"
echo "$domains"
```



```
┌─[parrot@parrot]─[~]
└──╼ $vi newfile.sh
┌─[parrot@parrot]─[~]
└──╼ $vi email.txt
┌─[parrot@parrot]─[~]
└──╼ $./newfile.sh
Enter filename containing email addresses:
email.txt
Number of unique domains: 4
List of unique domains:
gmail.com
outlook.com
protonmail.org
srmap.edu.in
┌─[parrot@parrot]─[~]
└──╼ $_
```

4. Write a script that reads the contents of a text file, sorts the lines in alphabetical order, and writes the sorted lines into a new file named sorted_output.txt.

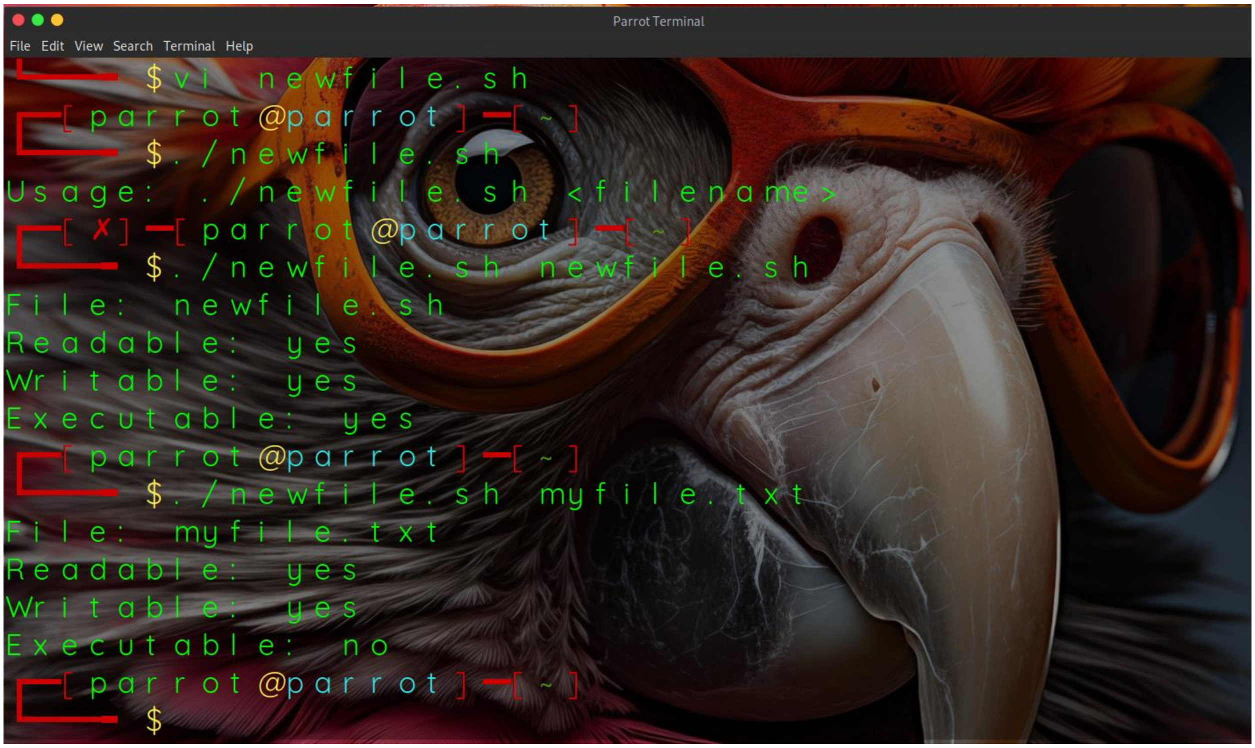
```
#!/bin/bash
echo "Enter input filename:"
read input_file

if [ ! -f "$input_file" ]; then
   echo "File not found!"
   exit 1
fi
output_file="sorted_output.txt"
sort "$input_file" > "$output_file"

echo "Sorted lines saved to $output_file"
```



```
┌─[parrot@parrot]─[~]
└──╼ $vi newfile.sh
┌─[parrot@parrot]─[~]
└──╼ $vi myfile.txt
┌─[parrot@parrot]─[~]
└──╼ $./newfile.sh
Enter input filename:
myfile.txt
Sorted lines saved to sorted_output.txt
┌─[parrot@parrot]─[~]
└──╼ $cat sorted_output.txt
A for apple
b for ball
t for telephone
x for xtreme
z for zebra
┌─[parrot@parrot]─[~]
└──╼ $_
```

5. Write a shell script that takes a filename as an argument and checks if the file is readable, writable, and executable.

```
#!/bin/bash

# Checking file permissions

if [ $# -ne 1 ]; then
   echo "Usage: $0 <filename>"
   exit 1
fi

filename="$1"

if [ ! -e "$filename" ]; then
   echo "File $filename does not exist."
   exit 1
fi

echo "File: $filename"
echo "Readable: $( [ -r "$filename" ] && echo "yes" || echo "no" )"
echo "Writable: $( [ -w "$filename" ] && echo "yes" || echo "no" )"
echo "Executable: $( [ -x "$filename" ] && echo "yes" || echo "no" )"
```



```
└──╼ $vi newfile.sh
┌─[parrot@parrot]─[~]
└──╼ $./newfile.sh
Usage: ./newfile.sh <filename>
┌─[✗]─[parrot@parrot]─[~]
└──╼ $./newfile.sh newfile.sh
File: newfile.sh
Readable: yes
Writable: yes
Executable: yes
┌─[parrot@parrot]─[~]
└──╼ $./newfile.sh myfile.txt
File: myfile.txt
Readable: yes
Writable: yes
Executable: no
┌─[parrot@parrot]─[~]
└──╼ $
```