

1)

The screenshot shows a code editor with a menu bar (Files, New, Log, Find, Info, Settings) and a tab for 'assignment.bash'. The editor displays the following script:

```
1 echo "Enter the length of the rectangle: "  
2 read l  
3 echo "Enter the breadth of the rectangle: "  
4 read b  
5 echo "Area of rectangle is: "  
6 expr $l \* $b
```

To the right of the editor is a terminal window showing the execution of the script:

```
~$ bash assignment.bash  
Enter the length of the rectangle:  
15  
Enter the breadth of the rectangle:  
12  
Area of rectangle is:  
180  
~$
```

2)

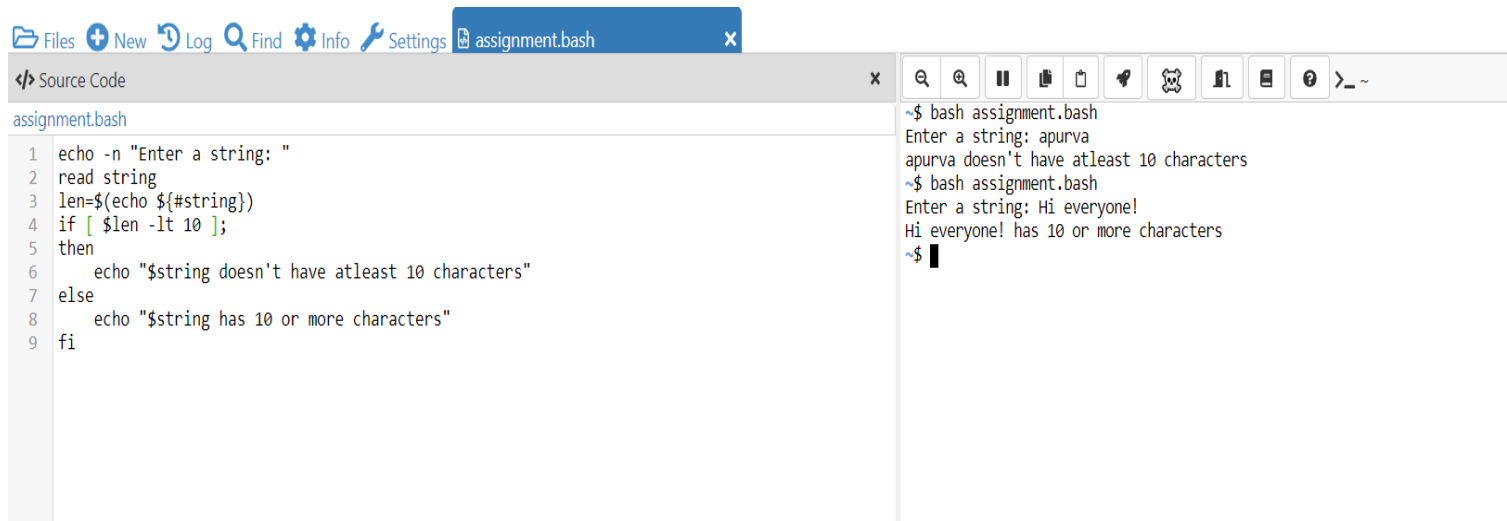
The screenshot shows a code editor with a menu bar (Files, New, Log, Find, Info, Settings) and a tab for 'assignment.bash'. The editor displays the following script:

```
1 echo -n "Enter a number: "  
2 read num  
3 fact=1  
4 for ((i=1;i<=num;i++))  
5 do  
6     fact=$((fact * $i))  
7 done  
8 echo "The factorial of the given number is: $fact"
```

To the right of the editor is a terminal window showing the execution of the script:

```
~$ bash assignment.bash  
Enter a number: 10  
The factorial of the given number is: 3628800  
~$ bash assignment.bash  
Enter a number: 6  
The factorial of the given number is: 720  
~$
```

3)

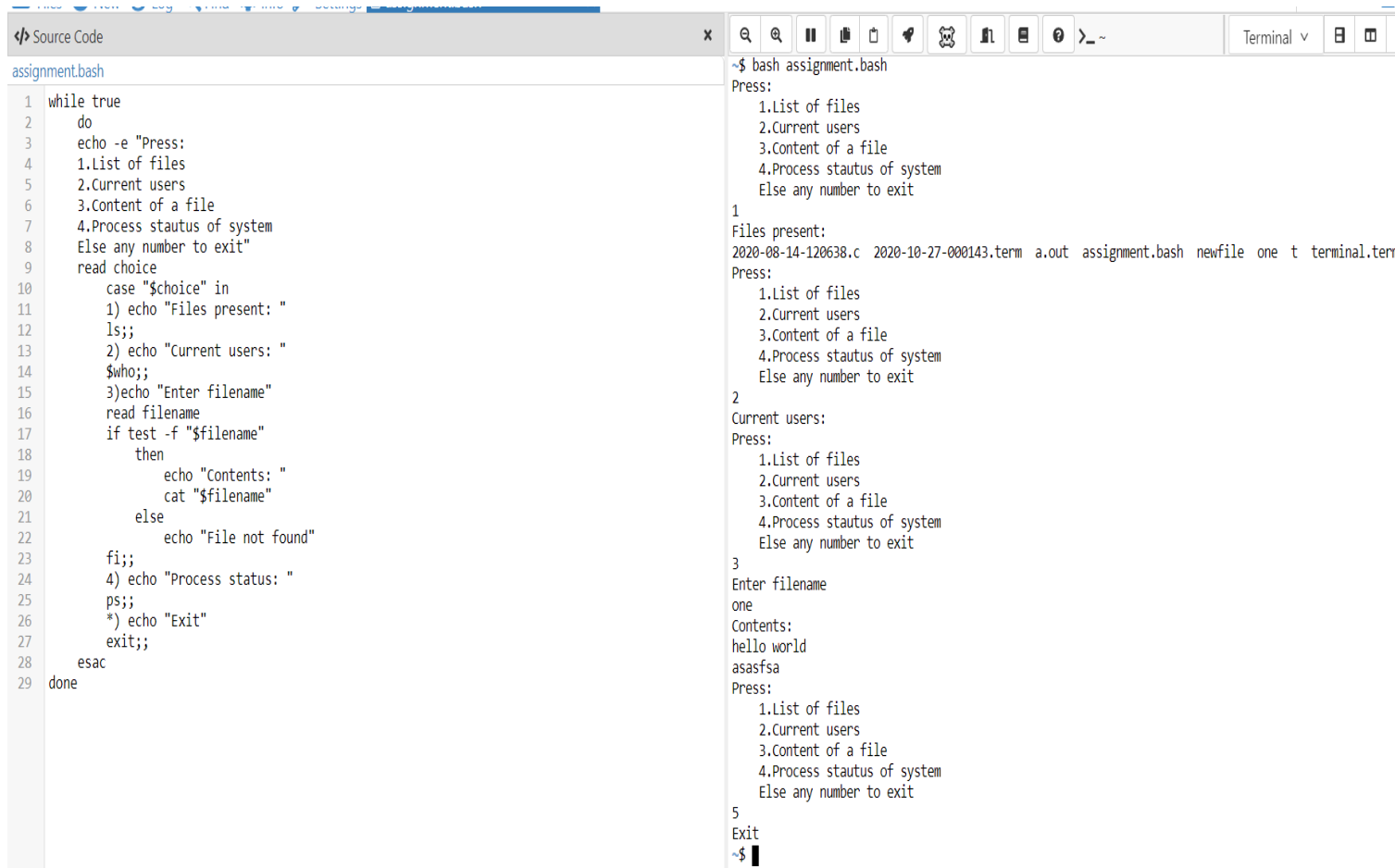


The screenshot shows a code editor with a file named `assignment.bash`. The script checks if a string has at least 10 characters. The terminal output shows the script being run twice: first with the input "apurva" (which is less than 10 characters) and then with "Hi everyone!" (which is 10 or more characters).

```
1 echo -n "Enter a string: "  
2 read string  
3 len=$(echo ${#string})  
4 if [ $len -lt 10 ];  
5 then  
6     echo "$string doesn't have atleast 10 characters"  
7 else  
8     echo "$string has 10 or more characters"  
9 fi
```

```
~$ bash assignment.bash  
Enter a string: apurva  
apurva doesn't have atleast 10 characters  
~$ bash assignment.bash  
Enter a string: Hi everyone!  
Hi everyone! has 10 or more characters  
~$
```

4)



The screenshot shows a code editor with a file named `assignment.bash`. The script is a menu-driven program that allows users to list files, show current users, display file contents, or show process status. The terminal output shows the script being run, displaying the menu, selecting option 1 (list files), option 2 (current users), and option 3 (enter filename), and then selecting option 4 (process status).

```
1 while true  
2 do  
3     echo -e "Press:  
4     1.List of files  
5     2.Current users  
6     3.Content of a file  
7     4.Process stautus of system  
8     Else any number to exit"  
9     read choice  
10    case "$choice" in  
11        1) echo "Files present: "  
12        ls;;  
13        2) echo "Current users: "  
14        $who;;  
15        3)echo "Enter filename"  
16        read filename  
17        if test -f "$filename"  
18            then  
19            echo "Contents: "  
20            cat "$filename"  
21        else  
22            echo "File not found"  
23        fi;;  
24        4) echo "Process status: "  
25        ps;;  
26        *) echo "Exit"  
27        exit;;  
28    esac  
29 done
```

```
~$ bash assignment.bash  
Press:  
 1.List of files  
 2.Current users  
 3.Content of a file  
 4.Process stautus of system  
 Else any number to exit  
1  
Files present:  
2020-08-14-120638.c 2020-10-27-000143.term a.out assignment.bash newfile one t terminal.term  
Press:  
 1.List of files  
 2.Current users  
 3.Content of a file  
 4.Process stautus of system  
 Else any number to exit  
2  
Current users:  
Press:  
 1.List of files  
 2.Current users  
 3.Content of a file  
 4.Process stautus of system  
 Else any number to exit  
3  
Enter filename  
one  
Contents:  
hello world  
asasfsa  
Press:  
 1.List of files  
 2.Current users  
 3.Content of a file  
 4.Process stautus of system  
 Else any number to exit  
5  
Exit  
~$
```