Bash Project

Bash Case

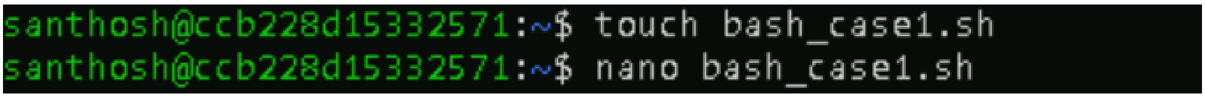
The **Bash case** statement is the simplest form of IF-THEN-ELSE with many ELIF elements. Using the case statement makes our bash script more readable and easier to maintain. These are generally applied to simplify the complex conditions having multiple different choices.

The Bash case statement follows a similar logic as the Javascript or C switch statement. There is a slight difference, as follows:

o The Bash case statement takes a value once and tests that value multiple times. It stops searching for a pattern once it has found it and executed the statement linked with it, which is almost opposite in case of the C switch statement.

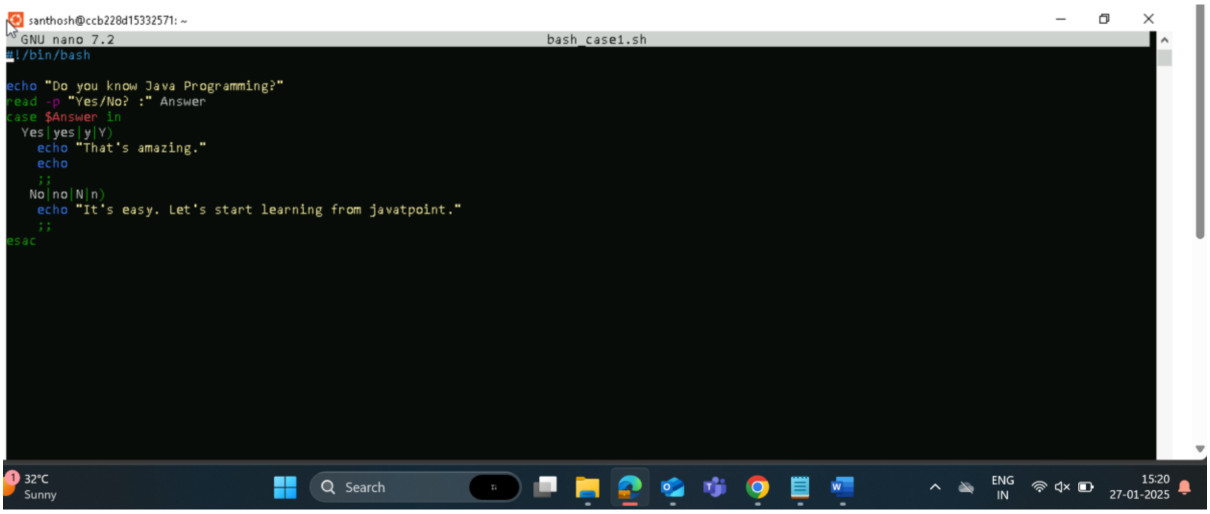
Example 1

Step 1: Creating a bash script using touch command and adding the script bby editing the file using nano command.



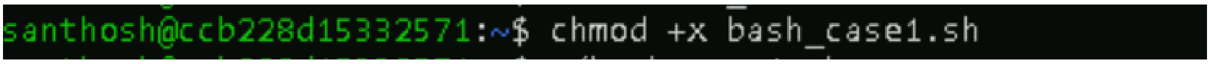
```
santhosh@ccb228d15332571:~$ touch bash_case1.sh
santhosh@ccb228d15332571:~$ nano bash_case1.sh
```

Step 2: Creating the script for a simple scenario to demonstrate the use of the case statement.



```
#!/bin/bash

echo "Do you know Java Programming?"
read -p "Yes/No? :" Answer
case $Answer in
  Yes|yes|y|Y)
    echo "That's amazing."
    echo
    ;;
   No|no|N|n)
    echo "It's easy. Let's start learning from javatpoint."
    ;;
esac
```

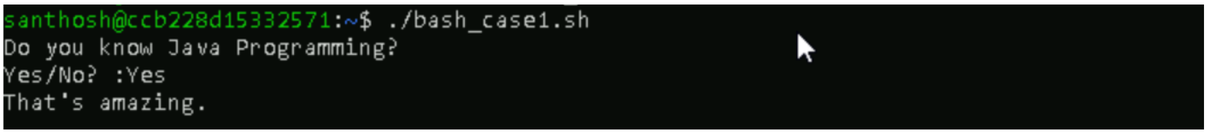
Step 3: Providing the necessary permissions for the base_case1.sh script.



```
santhosh@ccb228d15332571:~$ chmod +x bash_case1.sh
```

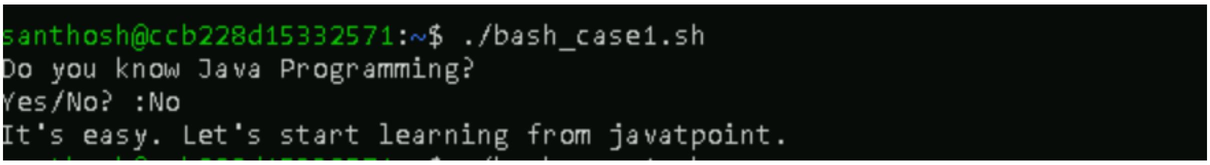
Step 4: Executing the output.

a. For Yes the output is.



```
santhosh@ccb228d15332571:~$ ./bash_case1.sh
Do you know Java Programming?
Yes/No? :Yes
That's amazing.
```

b. For No the output is.

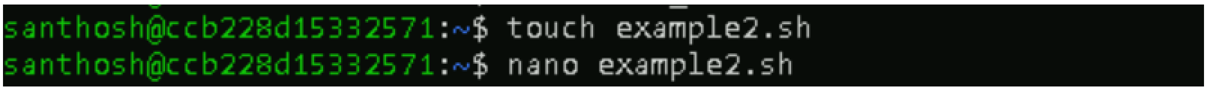


```
santhosh@ccb228d15332571:~$ ./bash_case1.sh
Do you know Java Programming?
Yes/No? :No
It's easy. Let's start learning from javatpoint.
```

Example 2

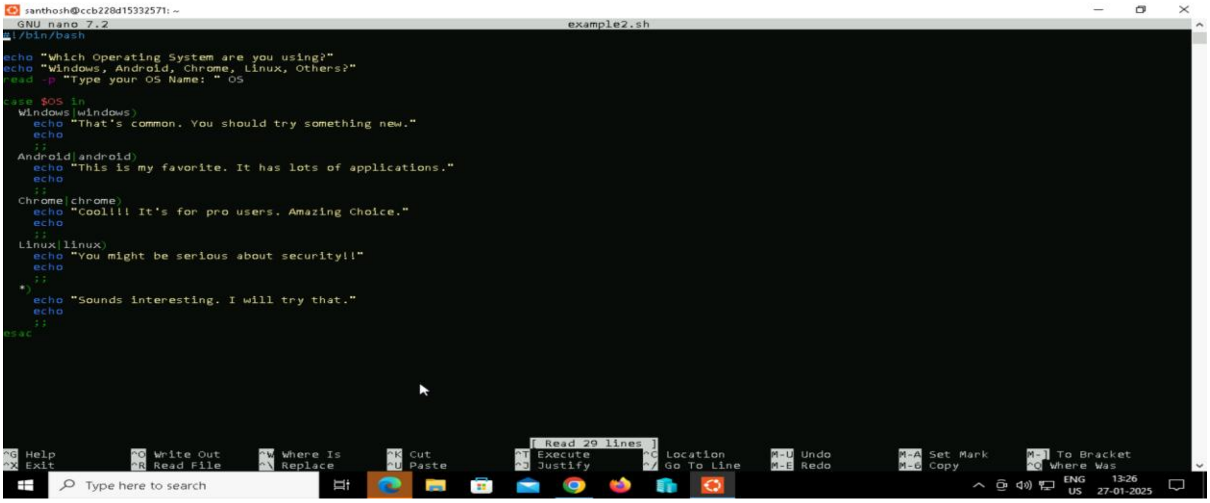
A combined scenario where there is also a default case when no previous matched case is found.

Step 1: Creating a bash script using touch command and adding the script bby editing the file using nano command.



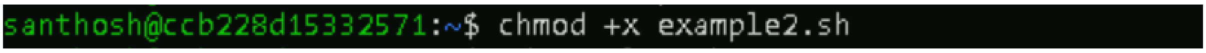
```
santhosh@ccb228d15332571:~$ touch example2.sh
santhosh@ccb228d15332571:~$ nano example2.sh
```

Step 2: Creating the script for a simple scenario to demonstrate the use of the case statement.



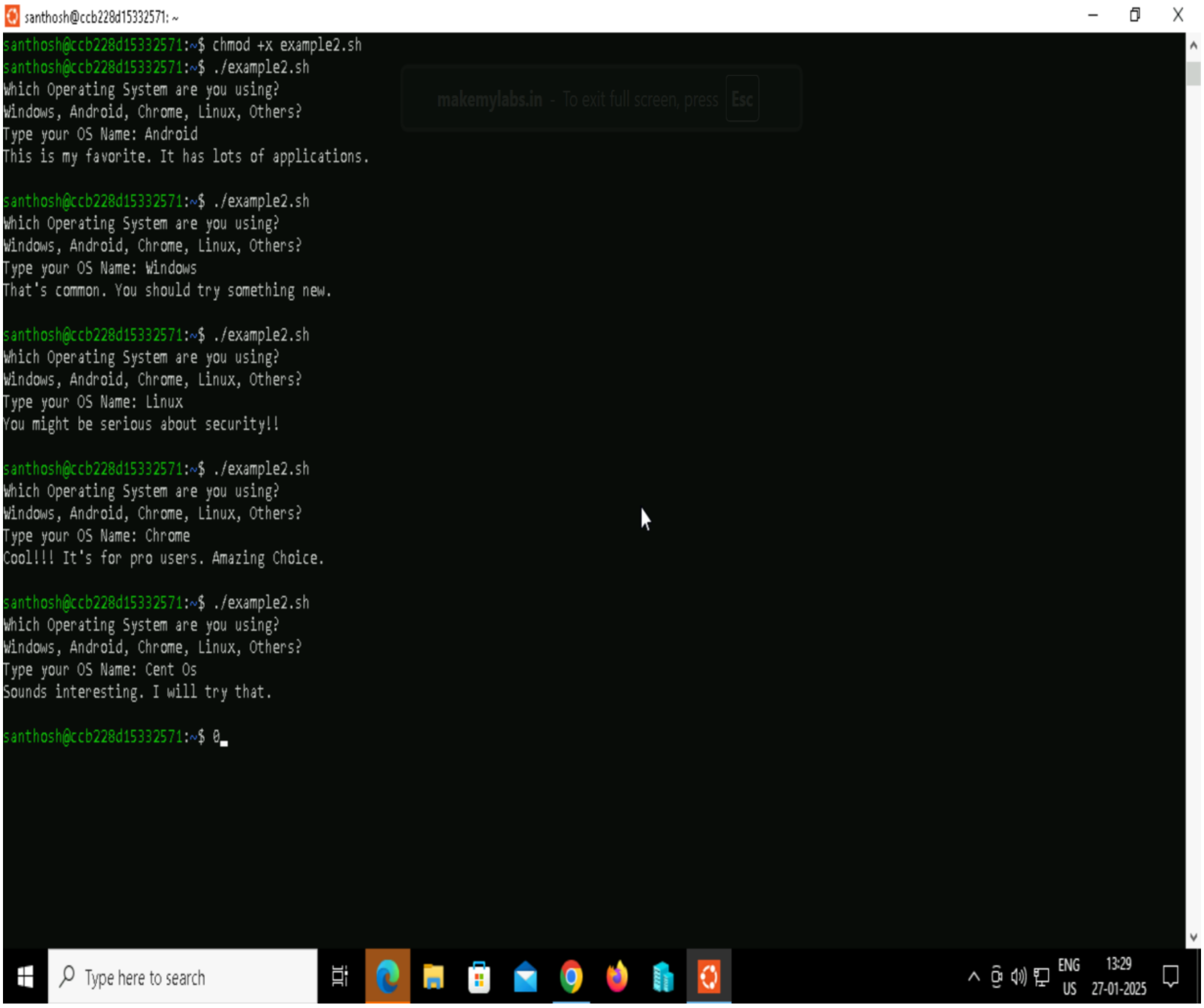
```
#!/bin/bash

echo "Which Operating System are you using?"
echo "Windows, Android, Chrome, Linux, Others?"
read -p "Type your OS Name: " OS

case $OS in
  Windows|windows)
    echo "That's common. You should try something new."
    echo
    ;;
  Android|android)
    echo "This is my favorite. It has lots of applications."
    echo
    ;;
  Chrome|chrome)
    echo "Cool!!! It's for pro users. Amazing Choice."
    echo
    ;;
  Linux|linux)
    echo "You might be serious about security!!"
    echo
    ;;
  *)
    echo "Sounds interesting. I will try that."
    echo
    ;;
esac
```

Step 3: Providing the necessary permissions for the example2.sh script.



```
santhosh@ccb228d15332571:~$ chmod +x example2.sh
```

Step 4: Executing the output.



santhosh@ccb228d15332571: ~
makemylabs.in - To exit full screen, press Esc
Type here to search
ENG
US
13:29
27-01-2025

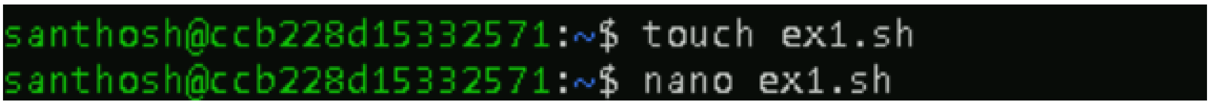
```
santhosh@ccb228d15332571:~$ chmod +x example2.sh
santhosh@ccb228d15332571:~$ ./example2.sh
Which Operating System are you using?
Windows, Android, Chrome, Linux, Others?
Type your OS Name: Android
This is my favorite. It has lots of applications.

santhosh@ccb228d15332571:~$ ./example2.sh
Which Operating System are you using?
Windows, Android, Chrome, Linux, Others?
Type your OS Name: Windows
That's common. You should try something new.

santhosh@ccb228d15332571:~$ ./example2.sh
Which Operating System are you using?
Windows, Android, Chrome, Linux, Others?
Type your OS Name: Linux
You might be serious about security!!

santhosh@ccb228d15332571:~$ ./example2.sh
Which Operating System are you using?
Windows, Android, Chrome, Linux, Others?
Type your OS Name: Chrome
Cool!!! It's for pro users. Amazing Choice.

santhosh@ccb228d15332571:~$ ./example2.sh
Which Operating System are you using?
Windows, Android, Chrome, Linux, Others?
Type your OS Name: Cent Os
Sounds interesting. I will try that.

santhosh@ccb228d15332571:~$ 0_
```

Bash For Loop

Like any other programming language, bash shell scripting also supports 'for loops' to perform repetitive tasks. It helps us to iterate a particular set of statements over a series of words in a string, or elements in an array. For example, you can either run UNIX command (or task) many times or just read and process the list of commands using a 'for loop'.

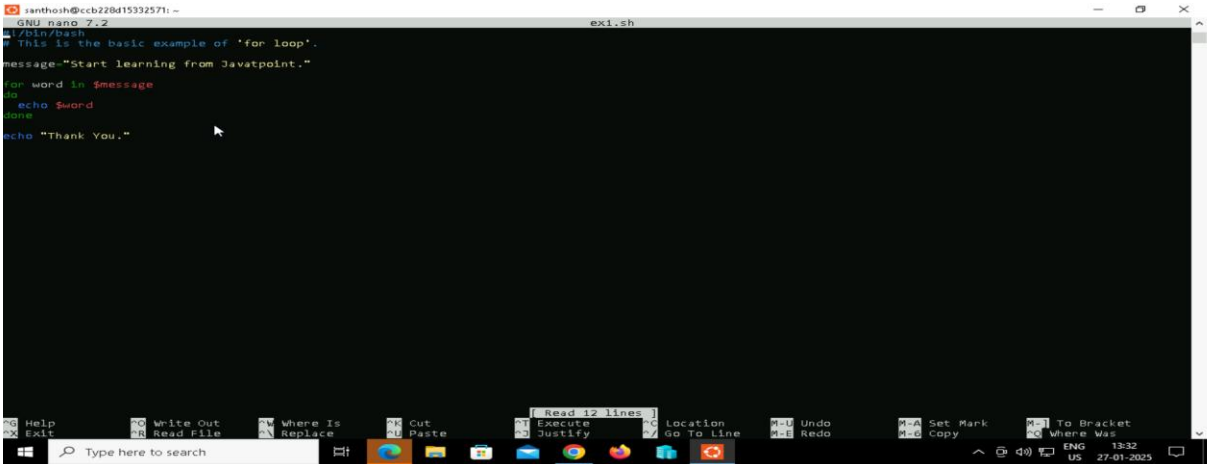
Example 1

Step 1: Creating a bash script using touch command and adding the script bby editing the file using nano command.



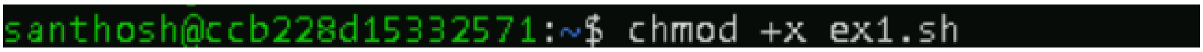
```
santhosh@ccb228d15332571:~$ touch ex1.sh
santhosh@ccb228d15332571:~$ nano ex1.sh
```

Step 2: Creating the script for a simple scenario to demonstrate the use of the case statement.



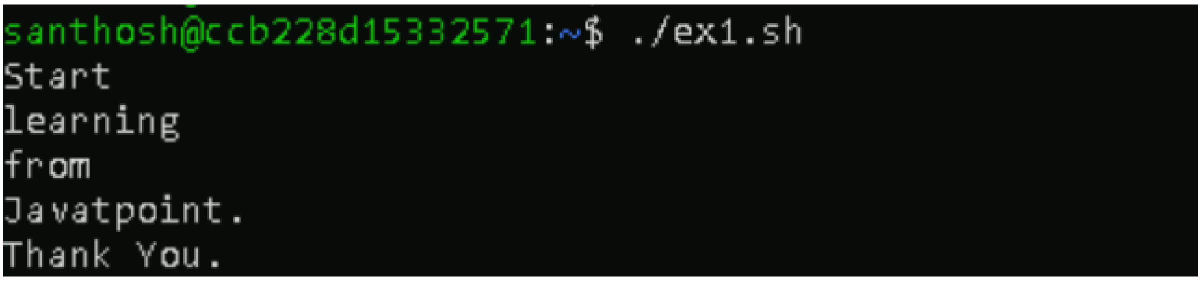
```
#!/bin/bash
# This is the basic example of 'for loop'.

message="Start learning from Javatpoint."

for word in $message
do
  echo $word
done

echo "Thank You."
```

Step 3: Providing the necessary permissions for the ex1.sh script.



```
santhosh@ccb228d15332571:~$ chmod +x ex1.sh
```

Step 4: Executing the output.

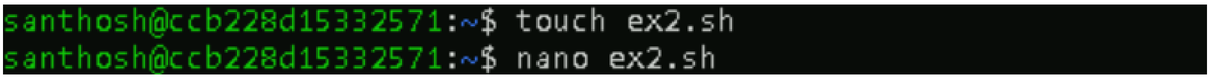


```
santhosh@ccb228d15332571:~$ ./ex1.sh
Start
learning
from
Javatpoint.
Thank You.
```

Example 2

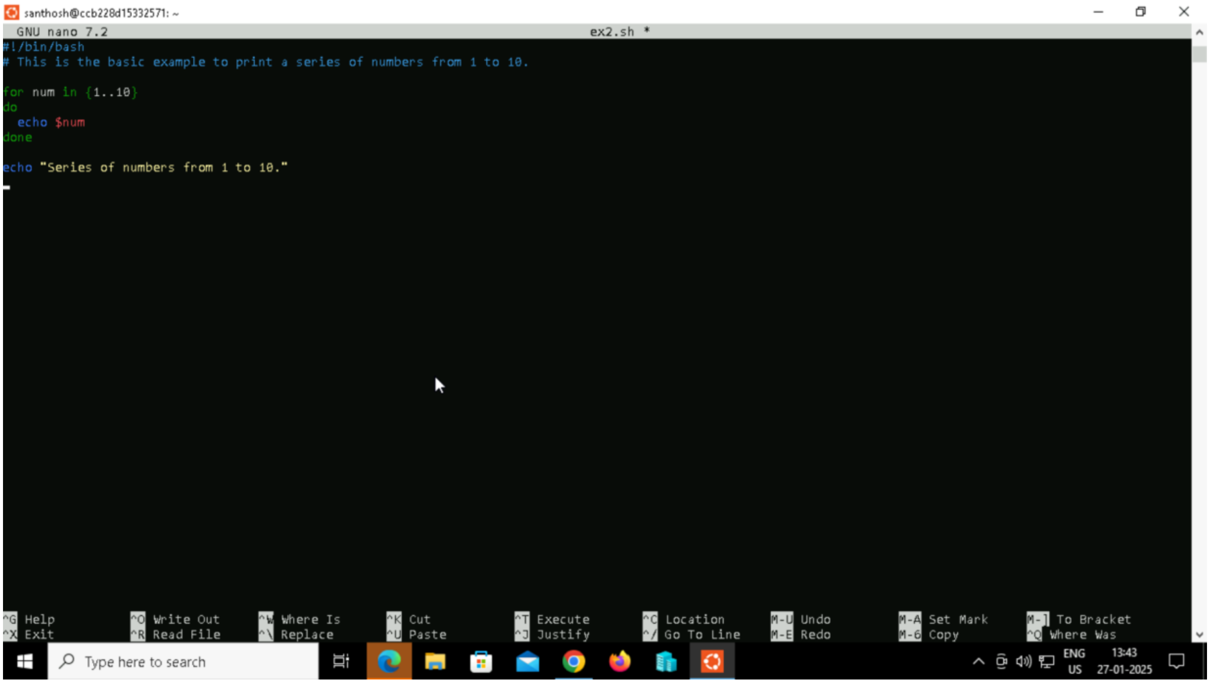
For Loop to Read a Range

Step 1: Creating a bash script using touch command and adding the script bby editing the file using nano command.



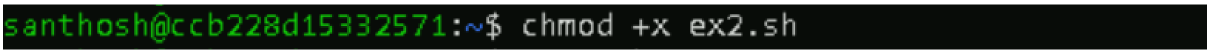
```
santhosh@ccb228d15332571:~$ touch ex2.sh
santhosh@ccb228d15332571:~$ nano ex2.sh
```

Step 2: Creating the script for a simple scenario to demonstrate the use of the case statement.



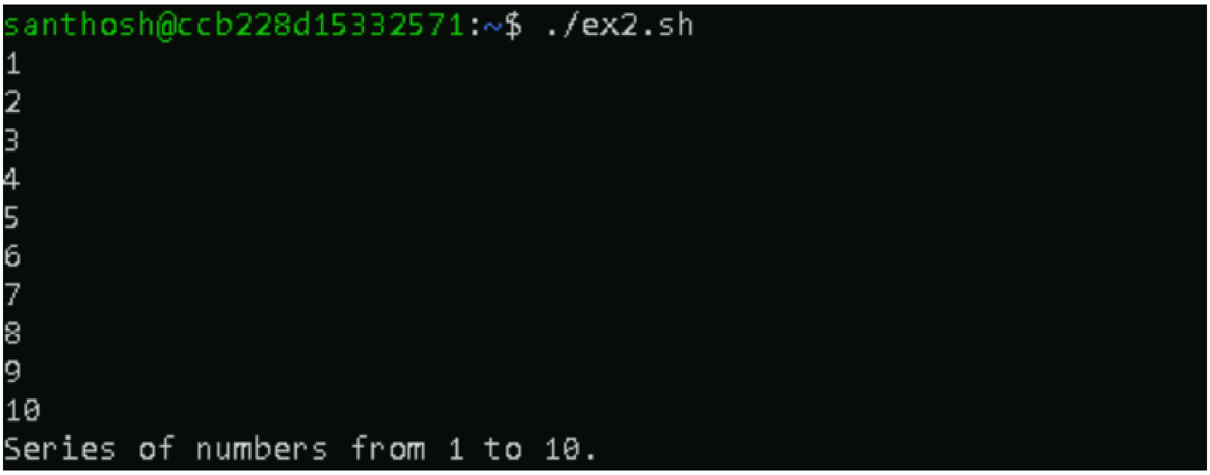
```
#!/bin/bash
# This is the basic example to print a series of numbers from 1 to 10.

for num in {1..10}
do
  echo $num
done

echo "Series of numbers from 1 to 10."
```

Step 3: Providing the necessary permissions for the ex2.sh script.



```
santhosh@ccb228d15332571:~$ chmod +x ex2.sh
```

Step 4: Executing the output.

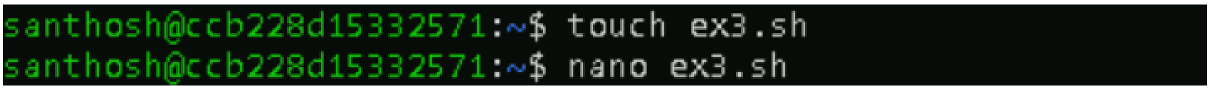


```
santhosh@ccb228d15332571:~$ ./ex2.sh
1
2
3
4
5
6
7
8
9
10
Series of numbers from 1 to 10.
```

Example 3

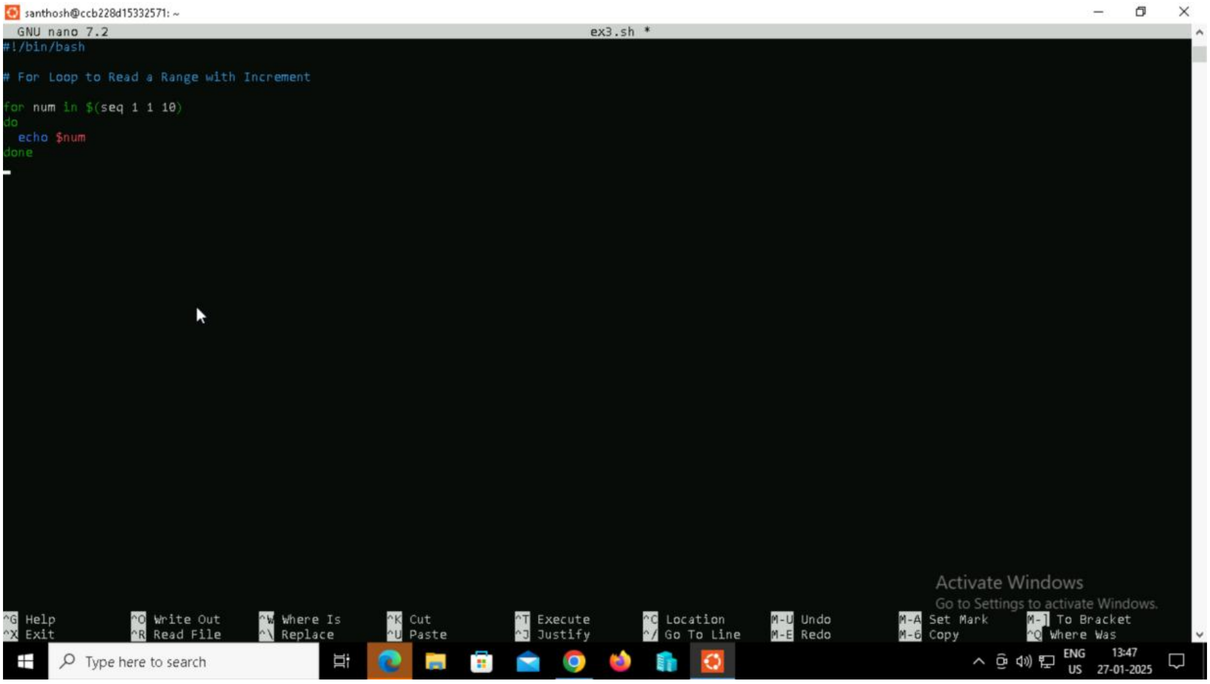
For Loop to Read a Range with Increment/Decrement

Step 1: Creating a bash script using touch command and adding the script bby editing the file using nano command.



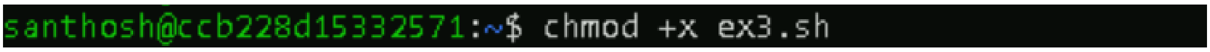
```
santhosh@ccb228d15332571:~$ touch ex3.sh
santhosh@ccb228d15332571:~$ nano ex3.sh
```

Step 2: Creating the script for a simple scenario to demonstrate the use of the case statement.



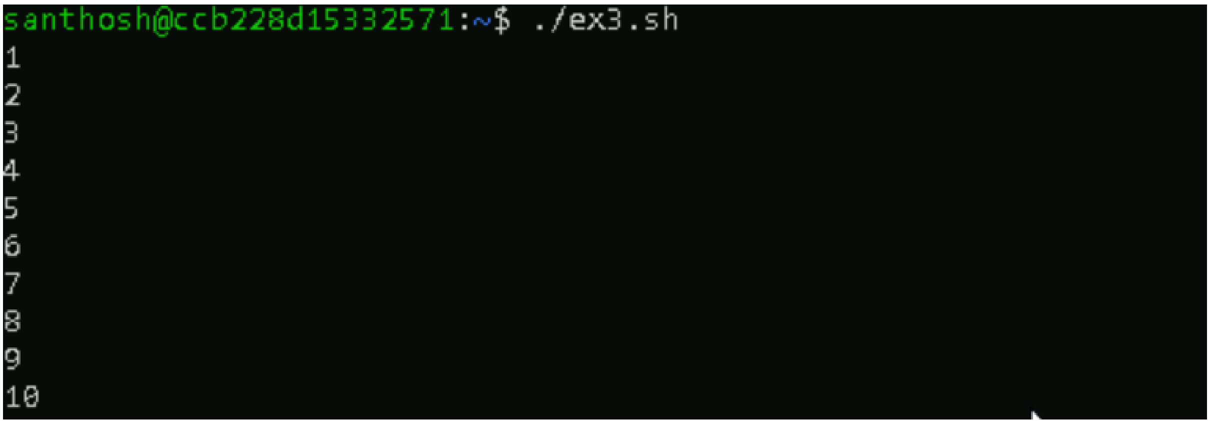
```
  GNU nano 7.2                                ex3.sh *
#!/bin/bash

# For Loop to Read a Range with Increment

for num in $(seq 1 1 10)
do
  echo $num
done
```

Step 3: Providing the necessary permissions for the ex3.sh script.



```
santhosh@ccb228d15332571:~$ chmod +x ex3.sh
```

Step 4: Executing the output.

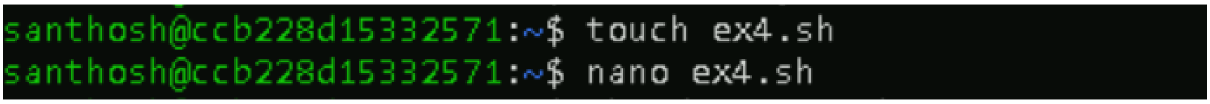


```
santhosh@ccb228d15332571:~$ ./ex3.sh
1
2
3
4
5
6
7
8
9
10
```

Example 4

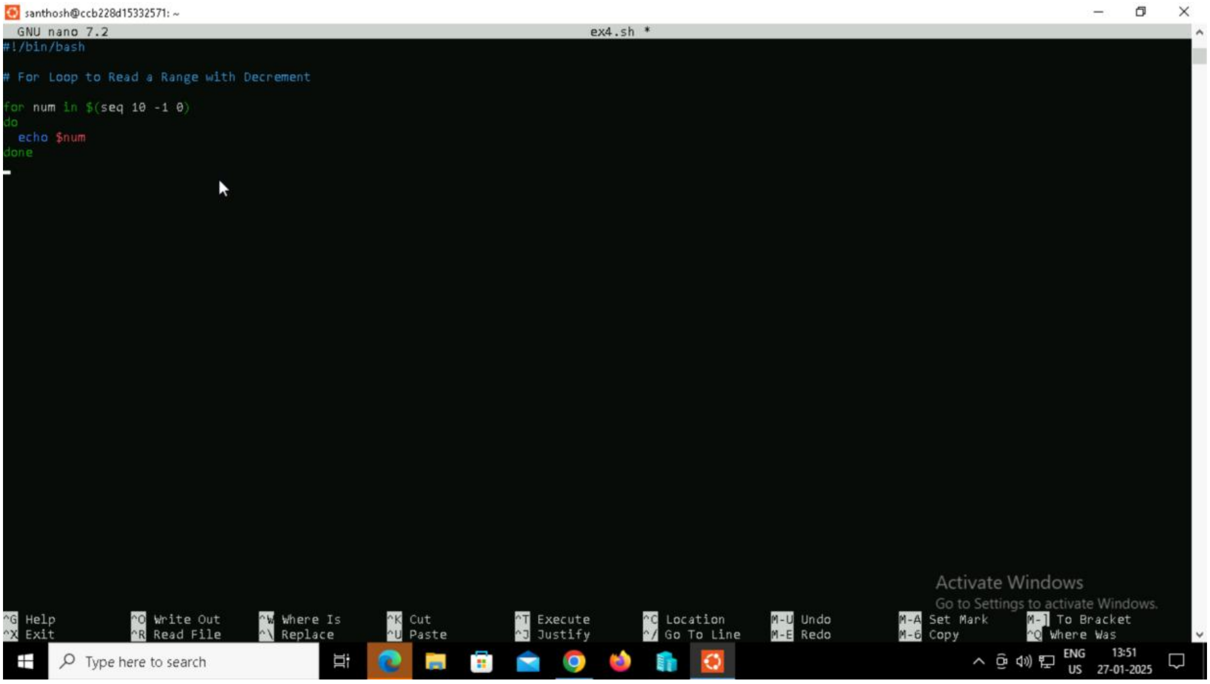
For Decrement

Step 1: Creating a bash script using touch command and adding the script bby editing the file using nano command.



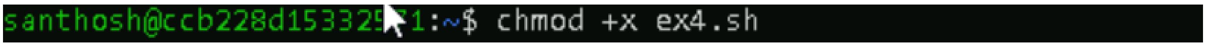
```
santhosh@ccb228d15332571:~$ touch ex4.sh
santhosh@ccb228d15332571:~$ nano ex4.sh
```

Step 2: Creating the script for a simple scenario to demonstrate the use of the case statement.



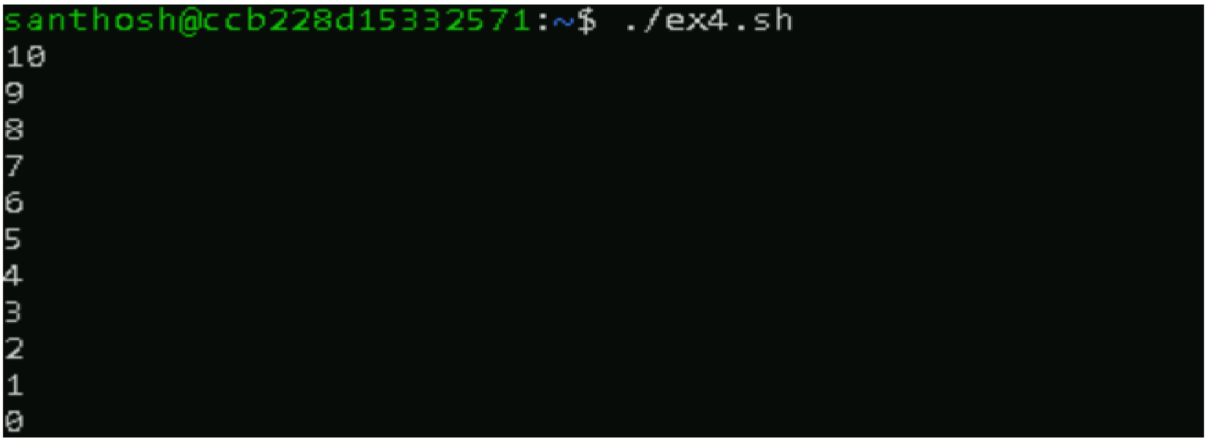
```
  GNU nano 7.2                                ex4.sh *
#!/bin/bash

# For Loop to Read a Range with Decrement

for num in $(seq 10 -1 0)
do
  echo $num
done
```

Step 3: Providing the necessary permissions for the ex4.sh script.



```
santhosh@ccb228d15332571:~$ chmod +x ex4.sh
```

Step 4: Executing the output.

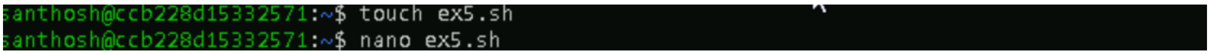


```
santhosh@ccb228d15332571:~$ ./ex4.sh
10
9
8
7
6
5
4
3
2
1
0
```

Example 5

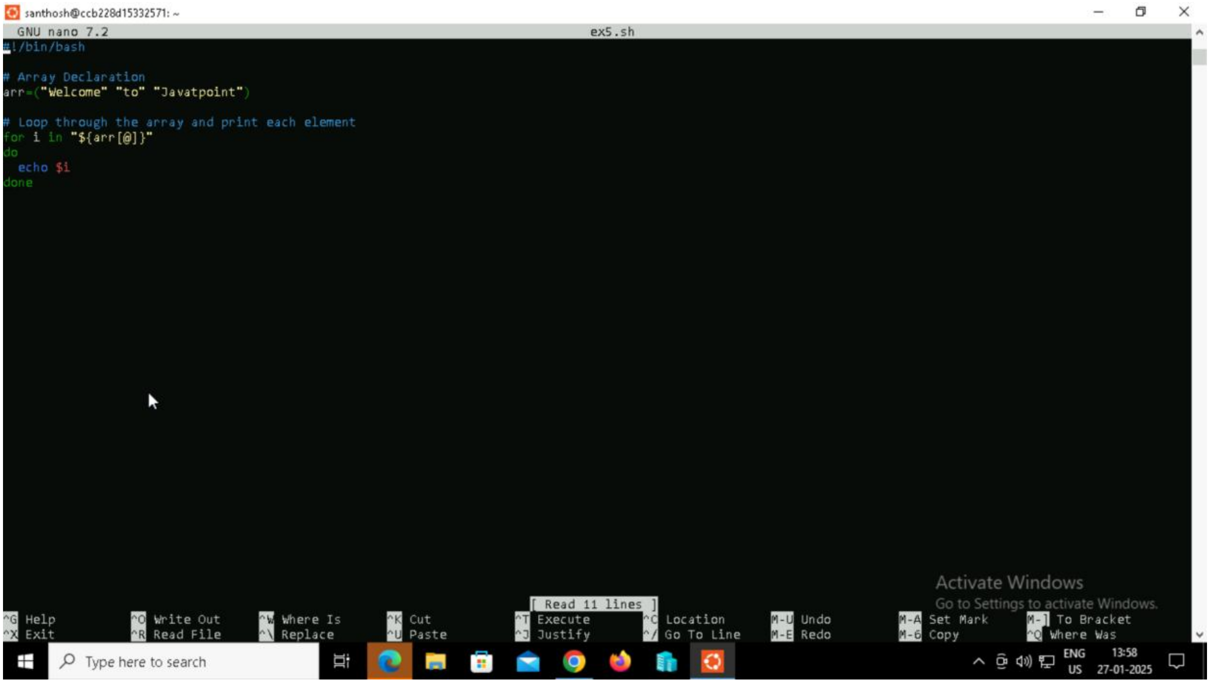
For Loop to Read Array Variables

Step 1: Creating a bash script using touch command and adding the script bby editing the file using nano command.



```
santhosh@ccb228d15332571:~$ touch ex5.sh
santhosh@ccb228d15332571:~$ nano ex5.sh
```

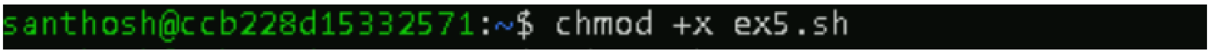
Step 2: Creating the script for a simple scenario to demonstrate the use of the case statement**.**



```
santhosh@ccb228d15332571: ~
  GNU nano 7.2                                  ex5.sh
#!/bin/bash

# Array Declaration
arr=("Welcome" "to" "Javatpoint")

# Loop through the array and print each element
for i in "${arr[@]}"
do
  echo $i
done
```

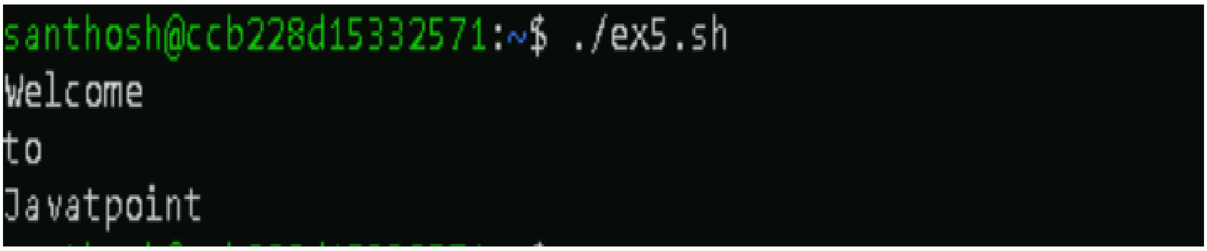
```
[ Read 11 lines ]
```

Step 3: Providing the necessary permissions for the ex5.sh script.



```
santhosh@ccb228d15332571:~$ chmod +x ex5.sh
```

Step 4: Executing the output.

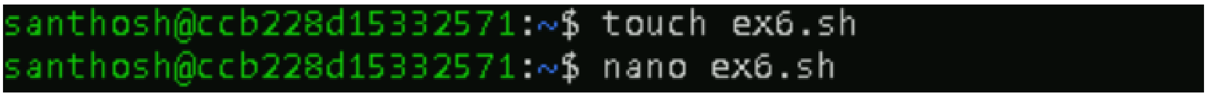


```
santhosh@ccb228d15332571:~$ ./ex5.sh
Welcome
to
Javatpoint
```

Example 6

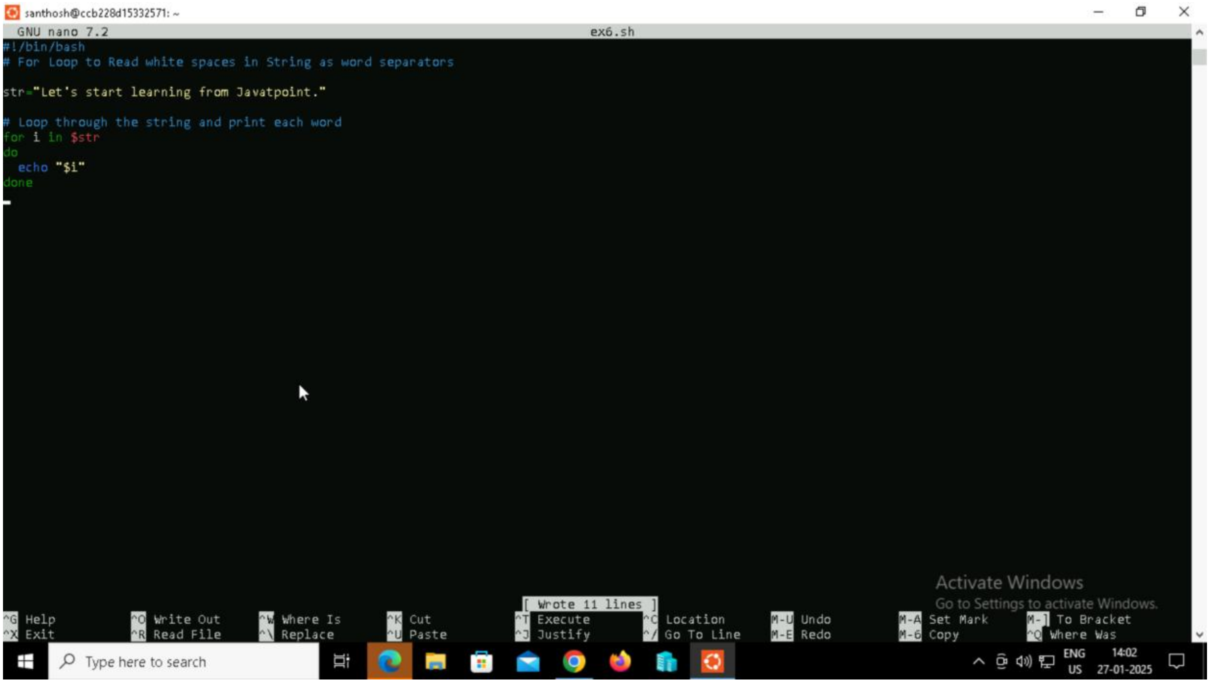
For Loop to Read white spaces in String as word separators.

Step 1: Creating a bash script using touch command and adding the script by editing the file using nano command.



```
santhosh@ccb228d15332571:~$ touch ex6.sh
santhosh@ccb228d15332571:~$ nano ex6.sh
```

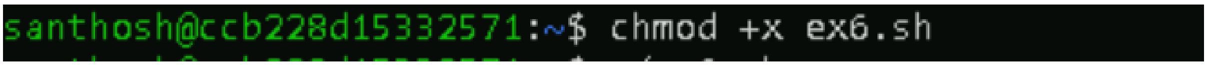
Step 2: Creating the script for a simple scenario to demonstrate the use of the case statement.



```
#!/bin/bash
# For Loop to Read white spaces in String as word separators

str="Let's start learning from Javatpoint."

# Loop through the string and print each word
for i in $str
do
  echo "$i"
done
```

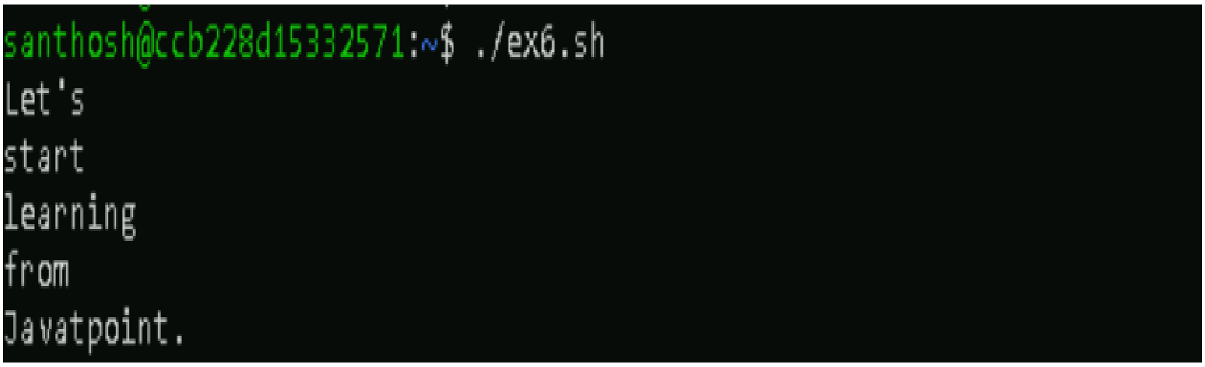
[Wrote 11 lines]

Step 3: Providing the necessary permissions for the ex6.sh script.



```
santhosh@ccb228d15332571:~$ chmod +x ex6.sh
```

Step 4: Executing the output.

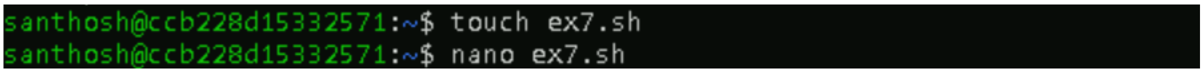


```
santhosh@ccb228d15332571:~$ ./ex6.sh
Let's
start
learning
from
Javatpoint.
```

Example 7

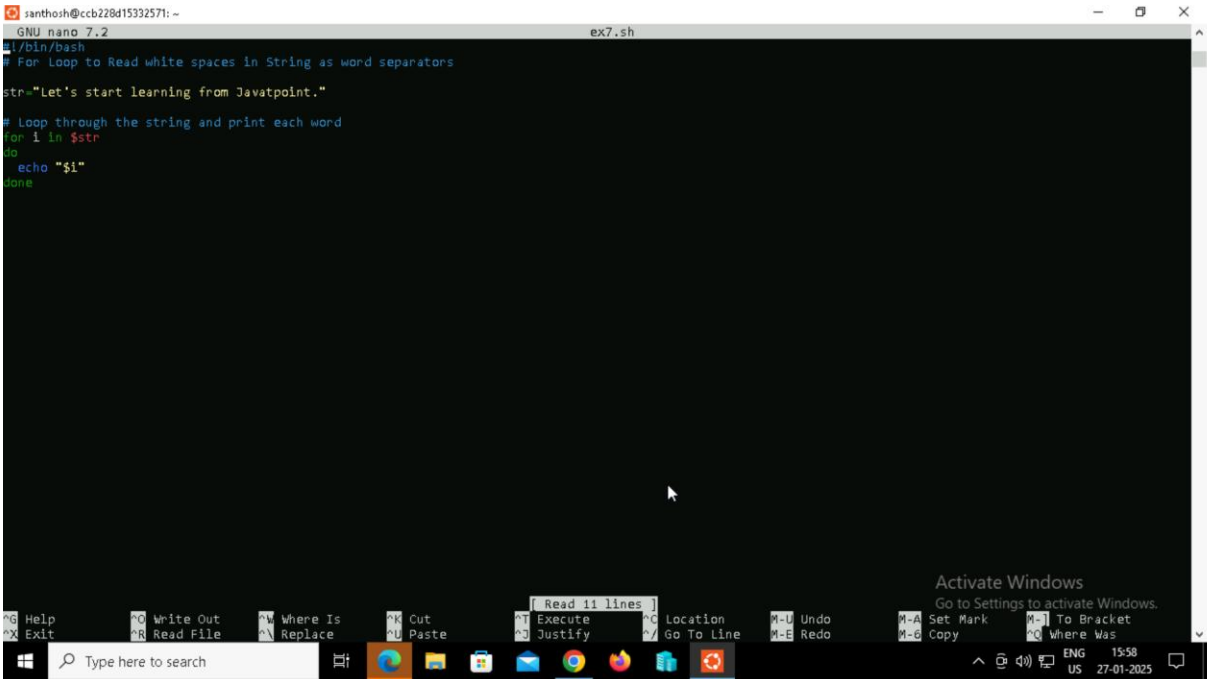
For Loop to Read each line in String as a word.

Step 1: Creating a bash script using touch command and adding the script bby editing the file using nano command.



```
santhosh@ccb228d15332571:~$ touch ex7.sh
santhosh@ccb228d15332571:~$ nano ex7.sh
```

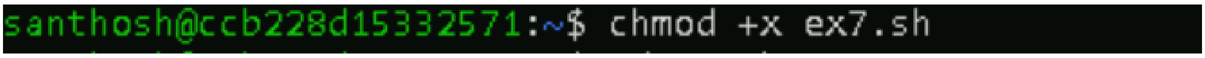
Step 2: Creating the script for a simple scenario to demonstrate the use of the case statement.



```
#!/bin/bash
# For Loop to Read white spaces in String as word separators

str="Let's start learning from Javatpoint."

# Loop through the string and print each word
for i in $str
do
  echo "$i"
done
```

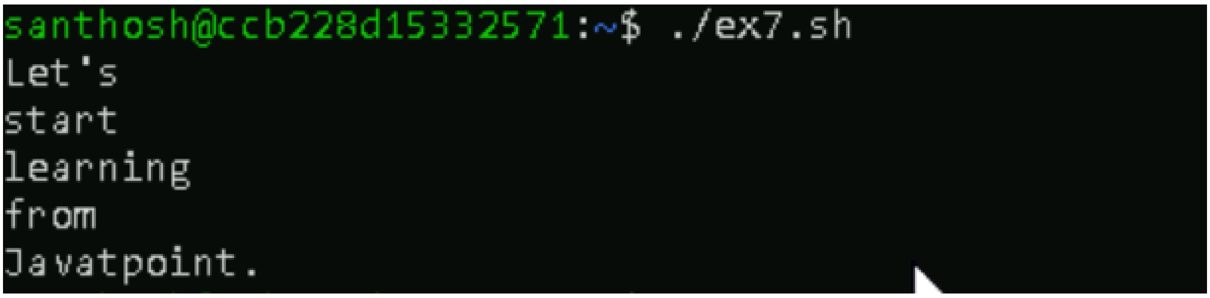
```
[ Read 11 lines ]
```

Step 3: Providing the necessary permissions for the ex7.sh script.



```
santhosh@ccb228d15332571:~$ chmod +x ex7.sh
```

Step 4: Executing the output.

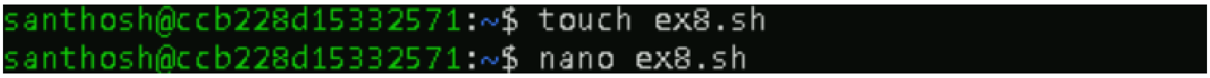


```
santhosh@ccb228d15332571:~$ ./ex7.sh
Let's
start
learning
from
Javatpoint.
```

Example 8

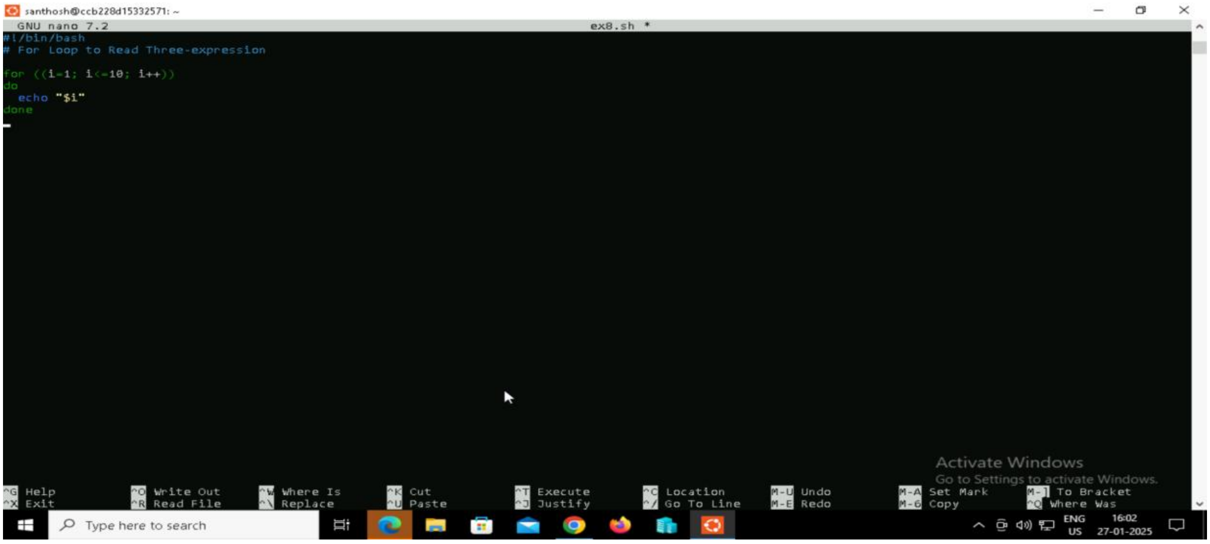
For Loop to Read Three-expression

Step 1: Creating a bash script using touch command and adding the script by editing the file using nano command.



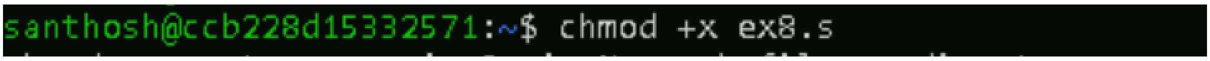
```
santhosh@ccb228d15332571:~$ touch ex8.sh
santhosh@ccb228d15332571:~$ nano ex8.sh
```

Step 2: Creating the script for a simple scenario to demonstrate the use of the case statement.



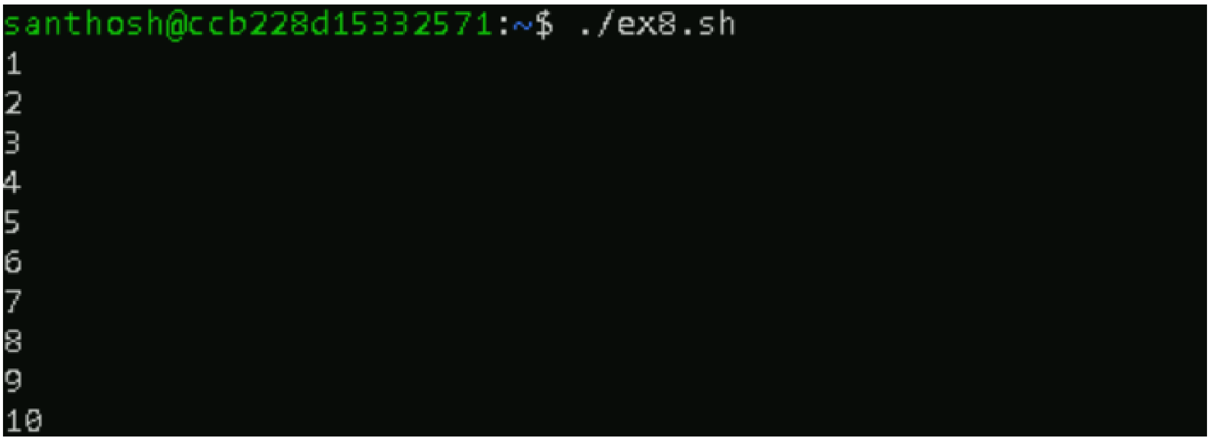
```
  GNU nano 7.2                                   ex8.sh *
#!/bin/bash
# For Loop to Read Three-expression

for ((i=1; i<=10; i++))
do
  echo "$i"
done
```

Step 3: Providing the necessary permissions for the ex8.sh script.



```
santhosh@ccb228d15332571:~$ chmod +x ex8.s
```

Step 4: Executing the output.

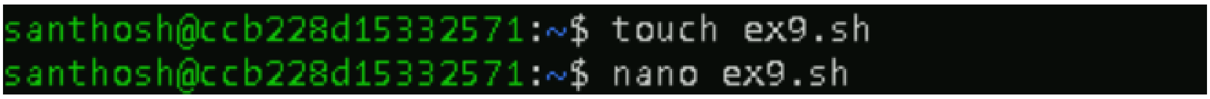


```
santhosh@ccb228d15332571:~$ ./ex8.sh
1
2
3
4
5
6
7
8
9
10
```

Example 9

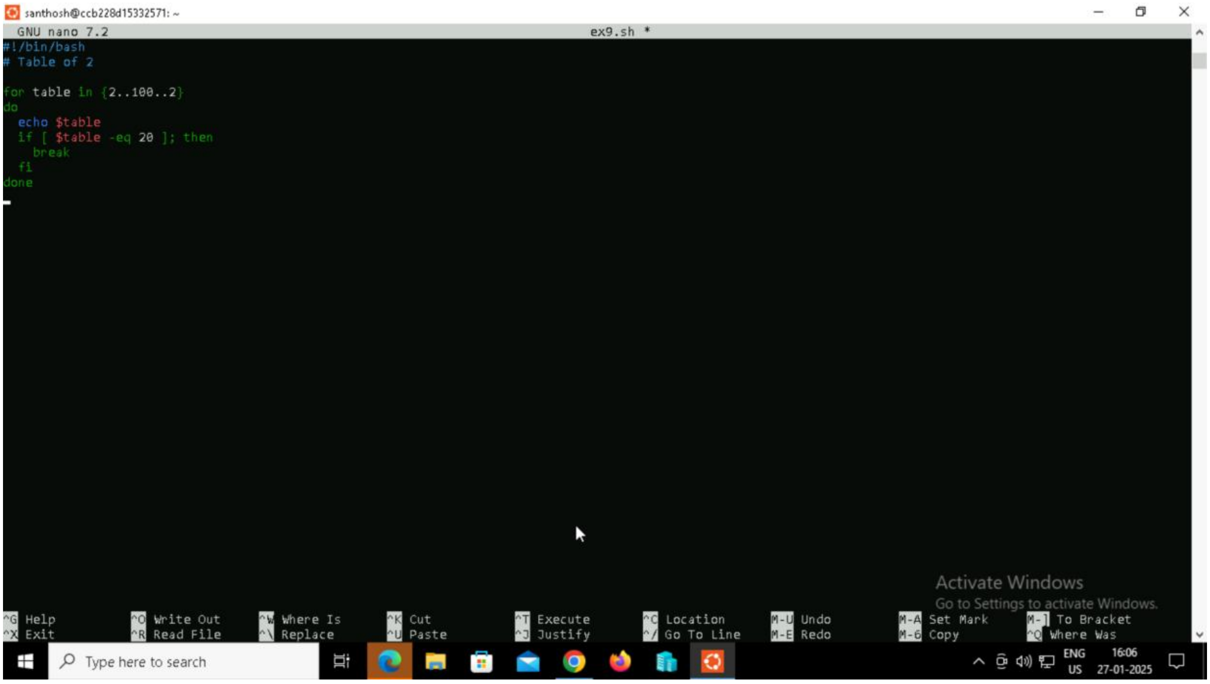
For Loop with a Break Statement

Step 1: Creating a bash script using touch command and adding the script bby editing the file using nano command.



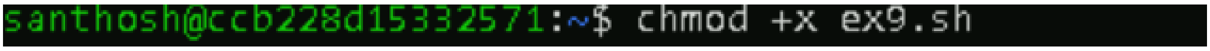
```
santhosh@ccb228d15332571:~$ touch ex9.sh
santhosh@ccb228d15332571:~$ nano ex9.sh
```

Step 2: Creating the script for a simple scenario to demonstrate the use of the case statement.



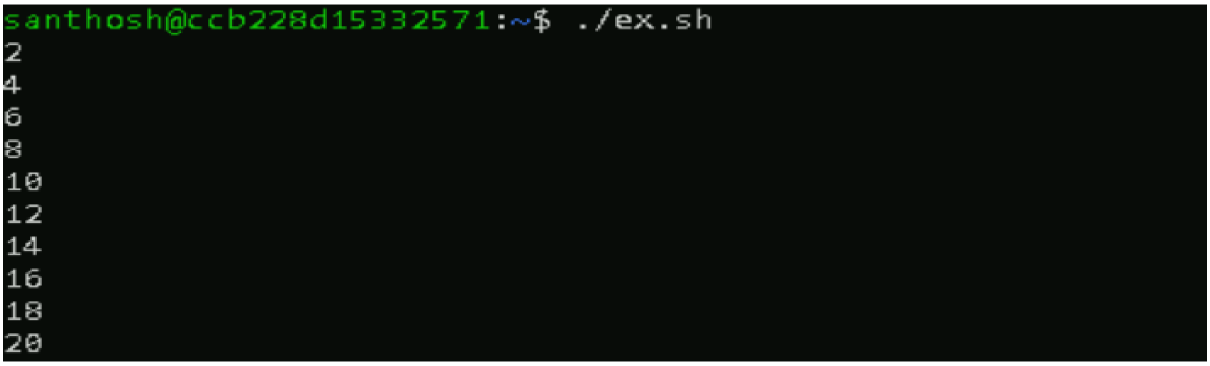
```
#!/bin/bash
# Table of 2

for table in {2..100..2}
do
  echo $table
  if [ $table -eq 20 ]; then
    break
  fi
done
```

Step 3: Providing the necessary permissions for the ex.sh script.



```
santhosh@ccb228d15332571:~$ chmod +x ex9.sh
```

Step 4: Executing the output.

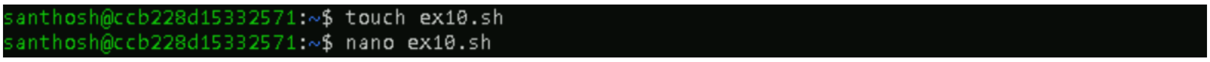


```
santhosh@ccb228d15332571:~$ ./ex.sh
2
4
6
8
10
12
14
16
18
20
```

Example 10

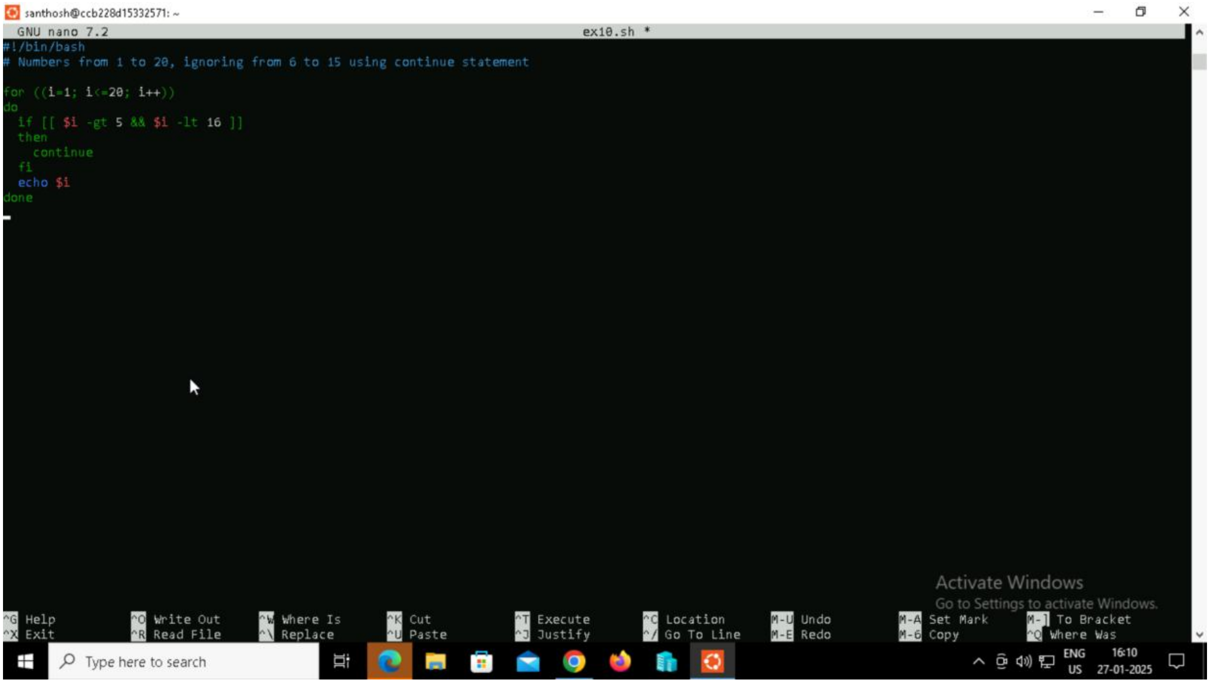
For Loop with a Continue Statement

Step 1: Creating a bash script using touch command and adding the script bby editing the file using nano command.



```
santhosh@ccb228d15332571:~$ touch ex10.sh
santhosh@ccb228d15332571:~$ nano ex10.sh
```

Step 2: Creating the script for a simple scenario to demonstrate the use of the case statement.

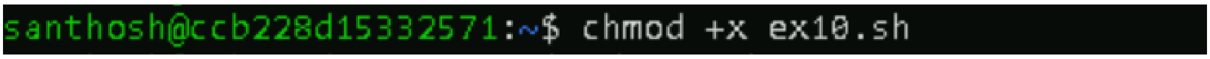


santhosh@ccb228d15332571: ~

GNU nano 7.2 ex10.sh *

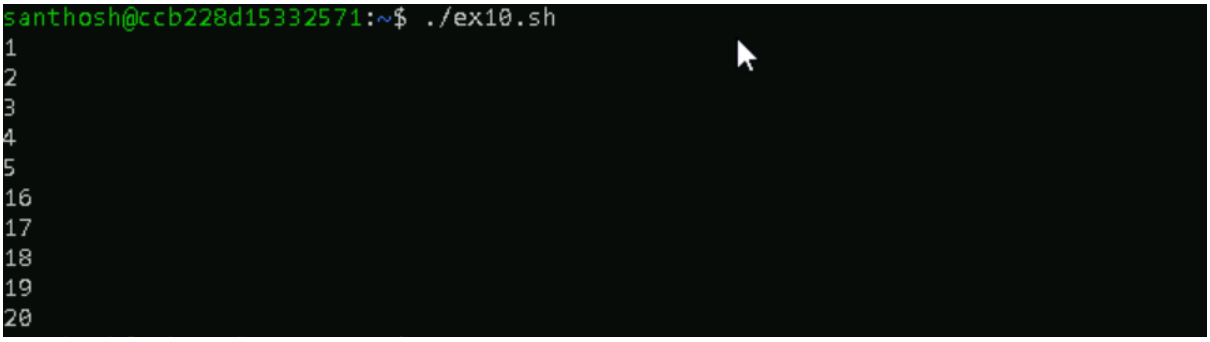
```
#!/bin/bash
# Numbers from 1 to 20, ignoring from 6 to 15 using continue statement

for ((i=1; i<=20; i++))
do
  if [[ $i -gt 5 && $i -lt 16 ]]
  then
    continue
  fi
  echo $i
done
```

Step 3: Providing the necessary permissions for the ex.sh script.



```
santhosh@ccb228d15332571:~$ chmod +x ex10.sh
```

Step 4: Executing the output.

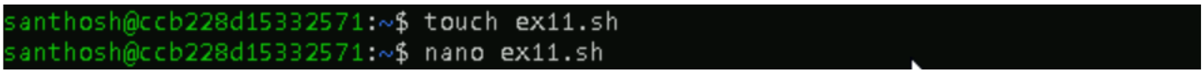


```
santhosh@ccb228d15332571:~$ ./ex10.sh
1
2
3
4
5
16
17
18
19
20
```

Example 11

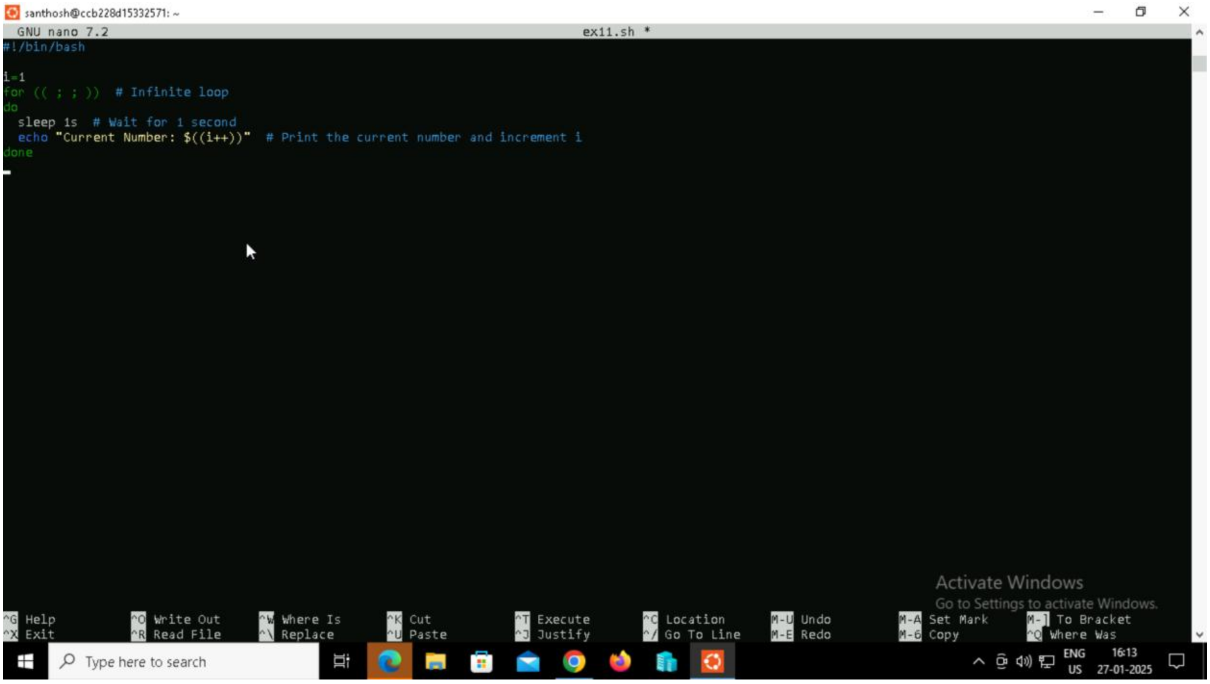
Infinite Bash for Loop

Step 1: Creating a bash script using touch command and adding the script bby editing the file using nano command.



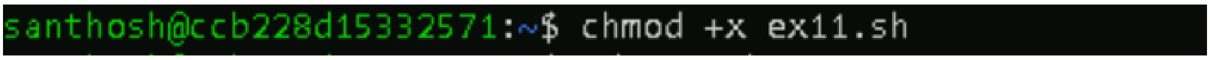
```
santhosh@ccb228d15332571:~$ touch ex11.sh
santhosh@ccb228d15332571:~$ nano ex11.sh
```

Step 2: Creating the script for a simple scenario to demonstrate the use of the case statement.



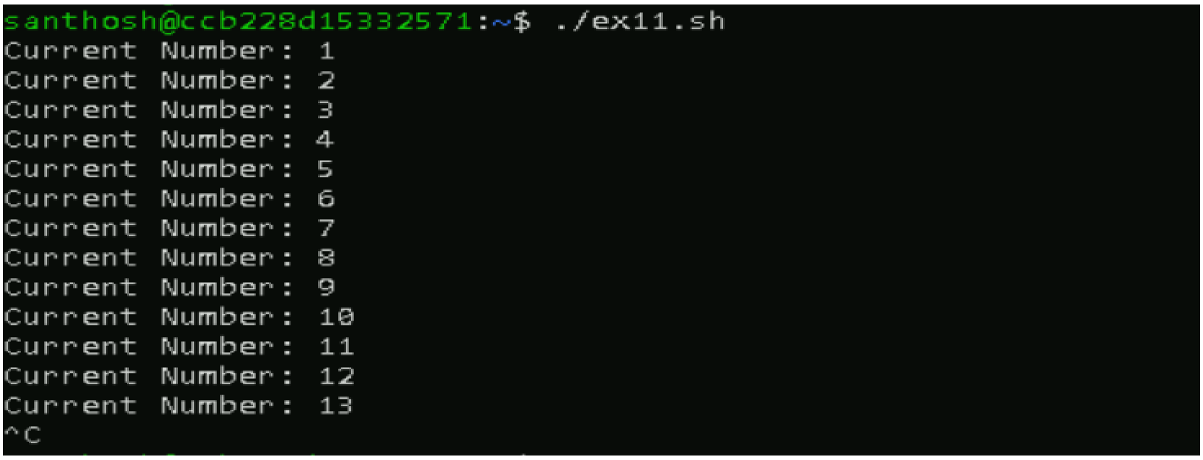
```
#!/bin/bash

i=1
for (( ; ; ))  # Infinite loop
do
  sleep 1s  # Wait for 1 second
  echo "Current Number: $((i++))"  # Print the current number and increment i
done
```

Step 3: Providing the necessary permissions for the ex.sh script.



```
santhosh@ccb228d15332571:~$ chmod +x ex11.sh
```

Step 4: Executing the output.



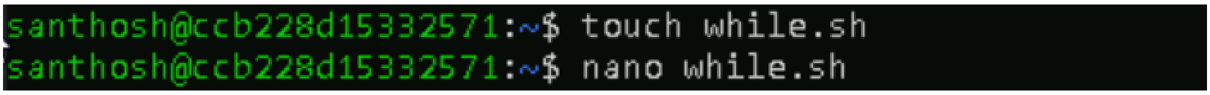
```
santhosh@ccb228d15332571:~$ ./ex11.sh
Current Number: 1
Current Number: 2
Current Number: 3
Current Number: 4
Current Number: 5
Current Number: 6
Current Number: 7
Current Number: 8
Current Number: 9
Current Number: 10
Current Number: 11
Current Number: 12
Current Number: 13
^C
```

BASH While Loop

The bash while loop can be defined as a control flow statement which allows executing the given set of commands repeatedly as long as the applied condition evaluates to true. For example, we can either run echo command many times or just read a text file line by line and process the result by using while loop in Bash.

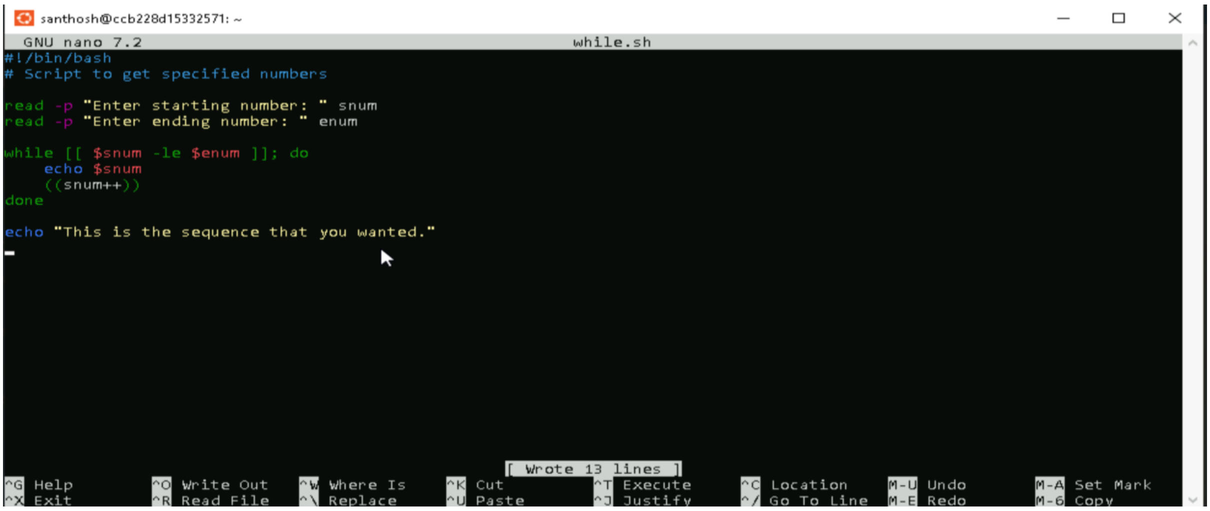
While Loop with Single Condition

Step 1: Creating a bash script using touch command and adding the script bby editing the file using nano command.



```
santhosh@ccb228d15332571:~$ touch while.sh
santhosh@ccb228d15332571:~$ nano while.sh
```

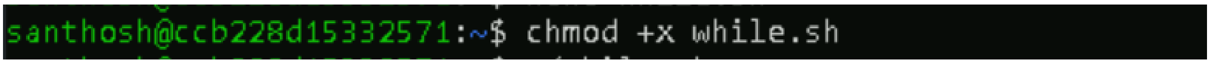
Step 2: Creating the script for a simple scenario to demonstrate the use of the case statement.



```
#!/bin/bash
# Script to get specified numbers

read -p "Enter starting number: " snum
read -p "Enter ending number: " enum

while [[ $snum -le $enum ]]; do
    echo $snum
    ((snum++))
done

echo "This is the sequence that you wanted."
```

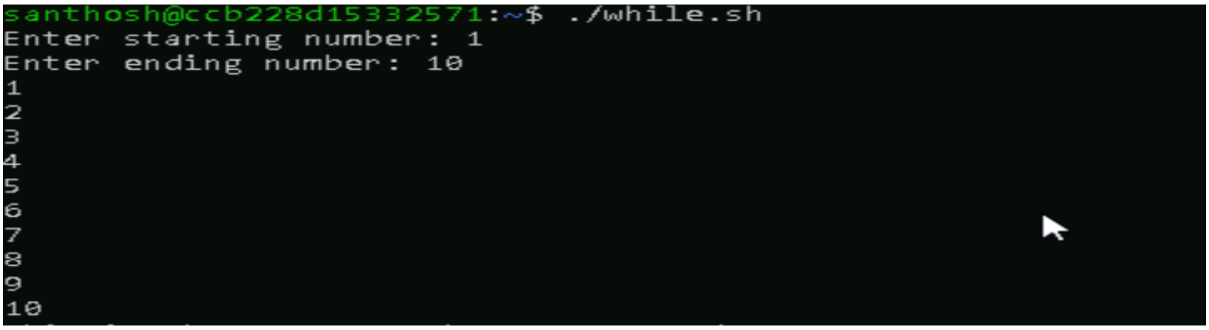
```
[ Wrote 13 lines ]
```

Step 3: Providing the necessary permissions for the ex.sh script.



```
santhosh@ccb228d15332571:~$ chmod +x while.sh
```

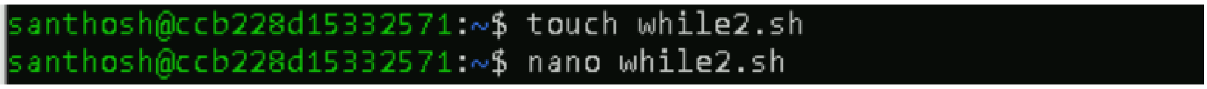
Step 4: Executing the output.



```
santhosh@ccb228d15332571:~$ ./while.sh
Enter starting number: 1
Enter ending number: 10
1
2
3
4
5
6
7
8
9
10
```

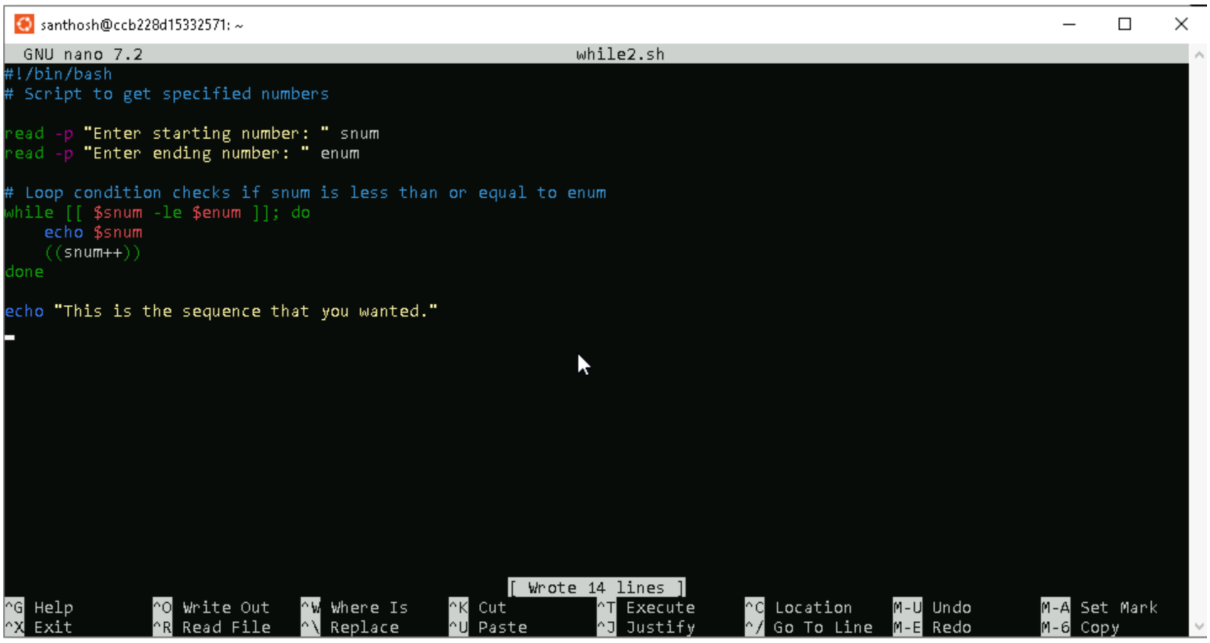
While Loop with Multiple Conditions

Step 1: Creating a bash script using touch command and adding the script bby editing the file using nano command.



```
santhosh@ccb228d15332571:~$ touch while2.sh
santhosh@ccb228d15332571:~$ nano while2.sh
```

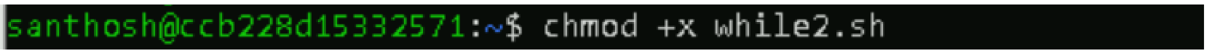
Step 2: Creating the script for a simple scenario to demonstrate the use of the case statement.



```
#!/bin/bash
# Script to get specified numbers

read -p "Enter starting number: " snum
read -p "Enter ending number: " enum

# Loop condition checks if snum is less than or equal to enum
while [[ $snum -le $enum ]]; do
    echo $snum
    ((snum++))
done

echo "This is the sequence that you wanted."
```

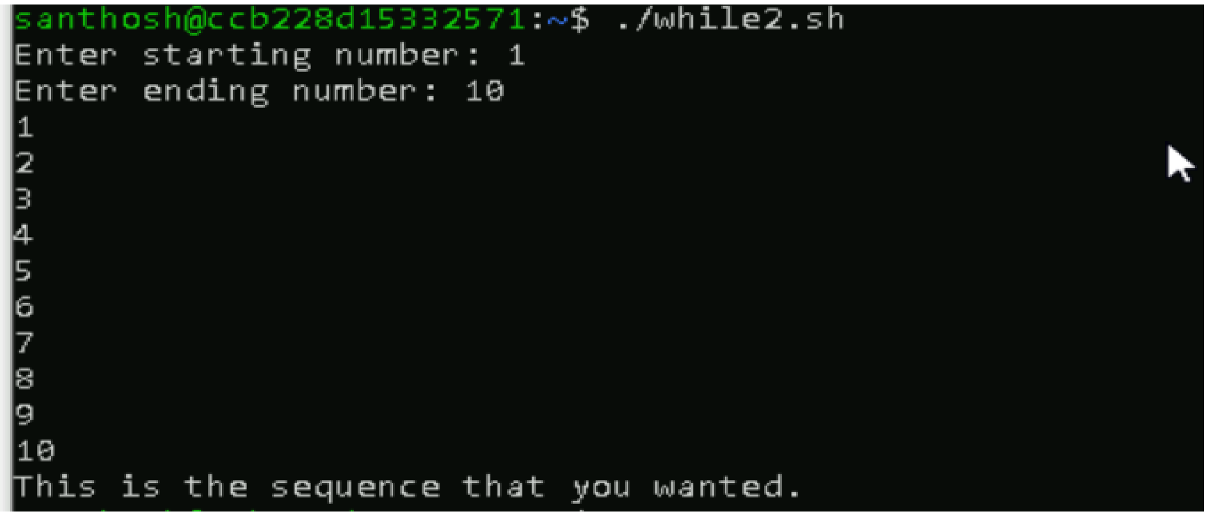
```
[ Wrote 14 lines ]
```

Step 3: Providing the necessary permissions for the ex.sh script.



```
santhosh@ccb228d15332571:~$ chmod +x while2.sh
```

Step 4: Executing the output.

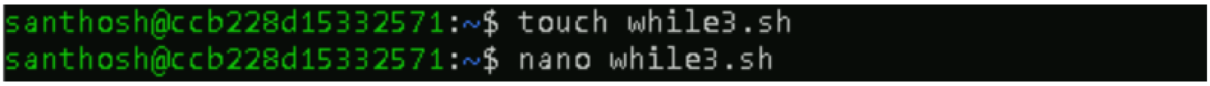


```
santhosh@ccb228d15332571:~$ ./while2.sh
Enter starting number: 1
Enter ending number: 10
1
2
3
4
5
6
7
8
9
10
This is the sequence that you wanted.
```

Infinite While Loop

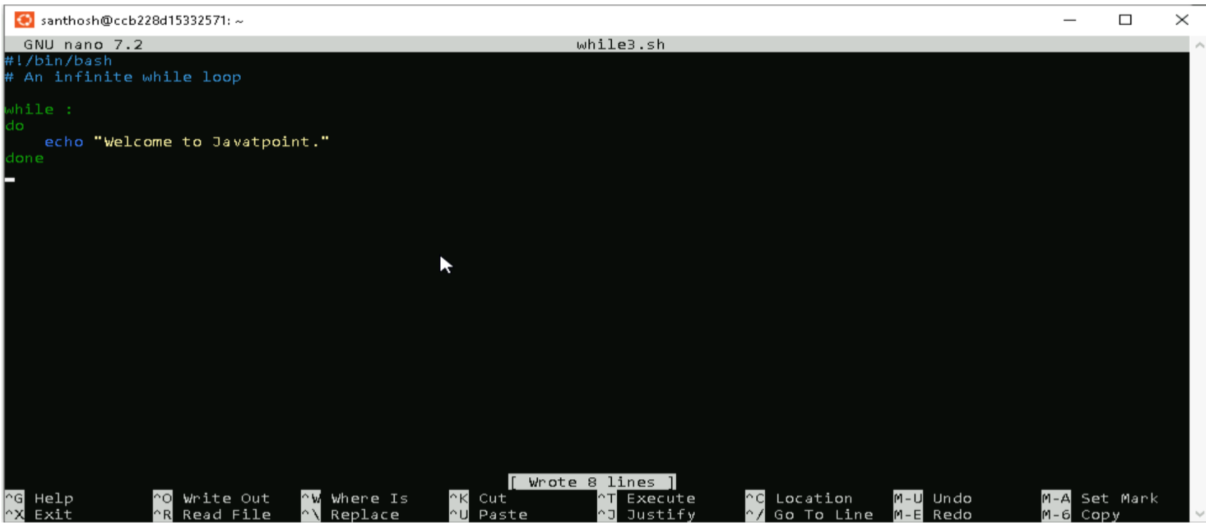
An infinite loop is a loop that has no ending or termination. If the condition always evaluates to true, it creates an infinite loop. The loop will execute continuously until it is forcefully stopped using CTRL+C :

Step 1: Creating a bash script using touch command and adding the script bby editing the file using nano command.



```
santhosh@ccb228d15332571:~$ touch while3.sh
santhosh@ccb228d15332571:~$ nano while3.sh
```

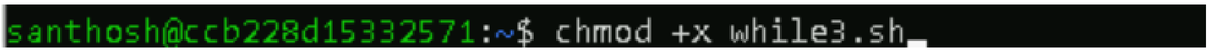
Step 2: Creating the script for a simple scenario to demonstrate the use of the case statement.



```
#!/bin/bash
# An infinite while loop

while :
do
    echo "Welcome to Javatpoint."
done
```

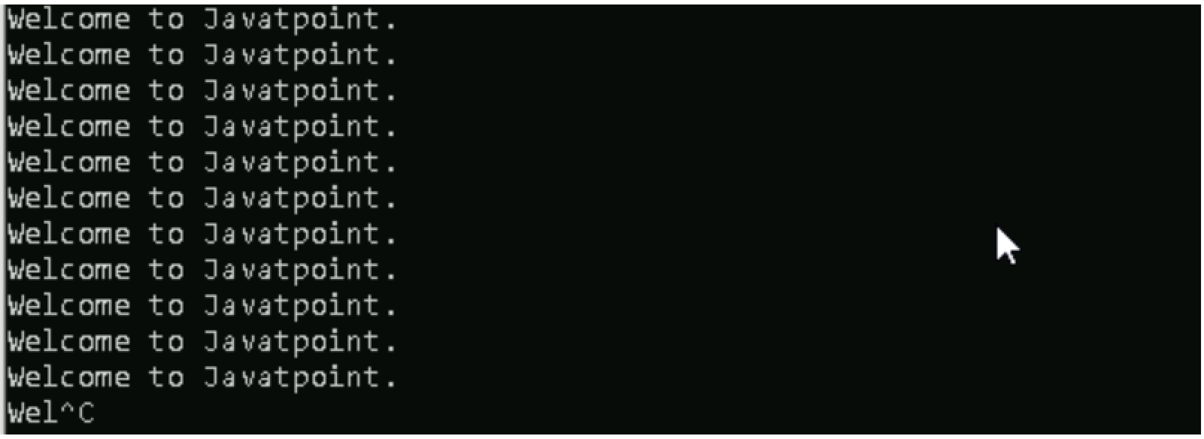
```
[ Wrote 8 lines ]
```

Step 3: Providing the necessary permissions for the ex.sh script.



```
santhosh@ccb228d15332571:~$ chmod +x while3.sh
```

Step 4: Executing the output.

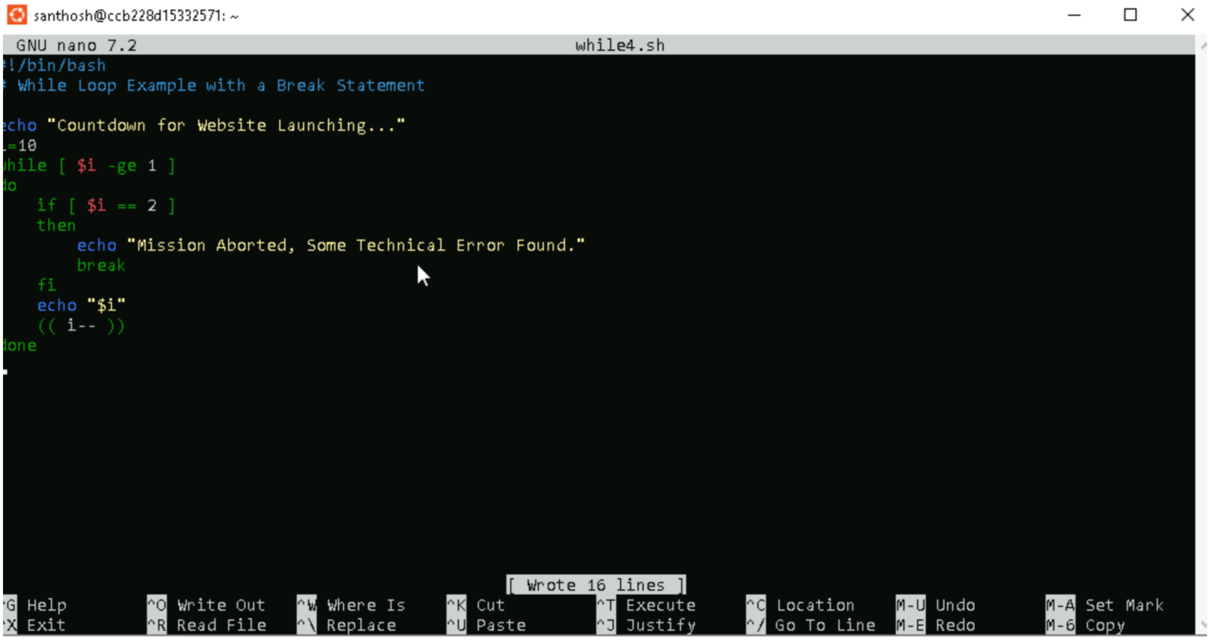


```
Welcome to Javatpoint.
Welcome to Javatpoint.
Welcome to Javatpoint.
Welcome to Javatpoint.
Welcome to Javatpoint.
Welcome to Javatpoint.
Welcome to Javatpoint.
Welcome to Javatpoint.
Welcome to Javatpoint.
Welcome to Javatpoint.
Welcome to Javatpoint.
Wel^C
```

While Loop with a Break Statement

Step 1: Creating a bash script using touch command and adding the script by editing the file using nano command.

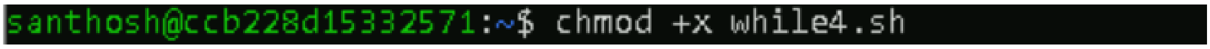
Step 2: Creating the script for a simple scenario to demonstrate the use of the case statement.



```
#!/bin/bash
# While Loop Example with a Break Statement

echo "Countdown for Website Launching..."
i=10
while [ $i -ge 1 ]
do
    if [ $i == 2 ]
    then
        echo "Mission Aborted, Some Technical Error Found."
        break
    fi
    echo "$i"
    (( i-- ))
done
```

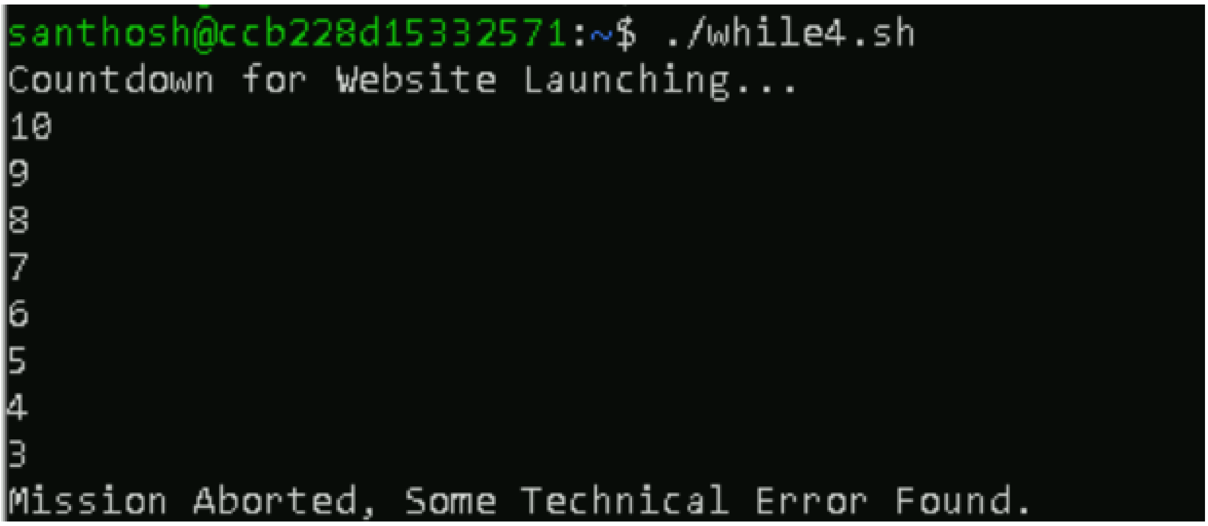
[Wrote 16 lines]

Step 3: Providing the necessary permissions for the ex.sh script.



```
santhosh@ccb228d15332571:~$ chmod +x while4.sh
```

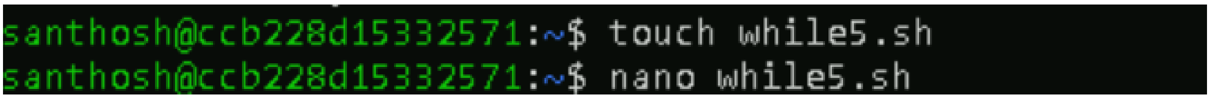
Step 4: Executing the output.



```
santhosh@ccb228d15332571:~$ ./while4.sh
Countdown for Website Launching...
10
9
8
7
6
5
4
3
Mission Aborted, Some Technical Error Found.
```

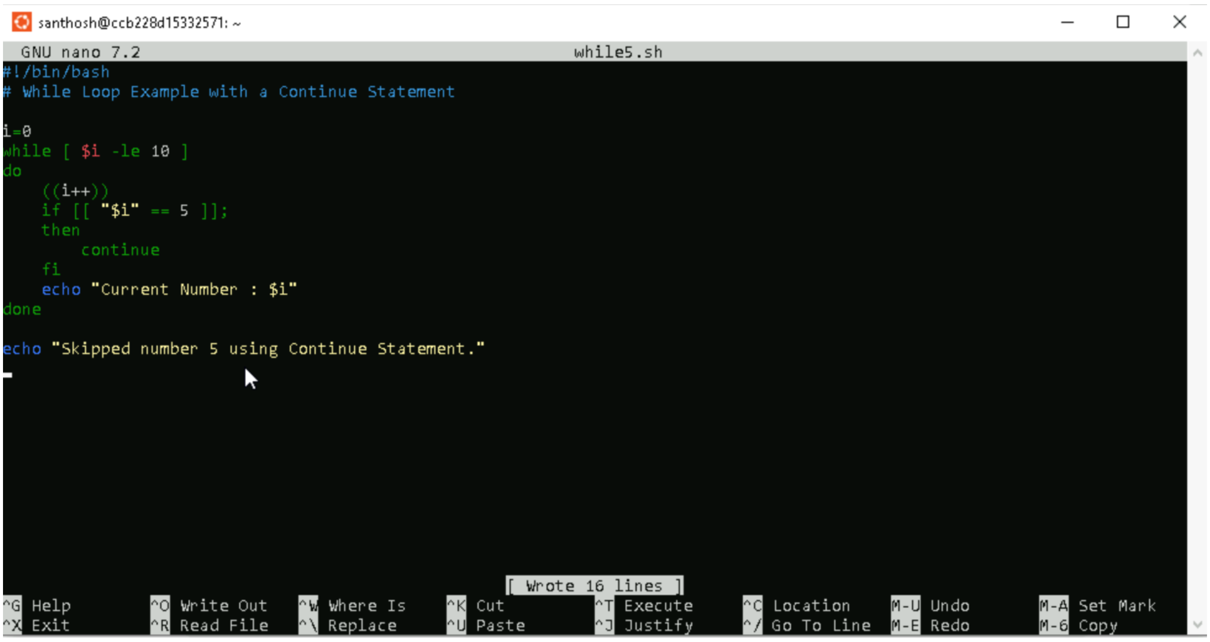
While Loop with a Continue Statement

Step 1: Creating a bash script using touch command and adding the script bby editing the file using nano command.



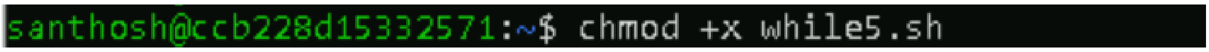
```
santhosh@ccb228d15332571:~$ touch while5.sh
santhosh@ccb228d15332571:~$ nano while5.sh
```

Step 2: Creating the script for a simple scenario to demonstrate the use of the case statement.



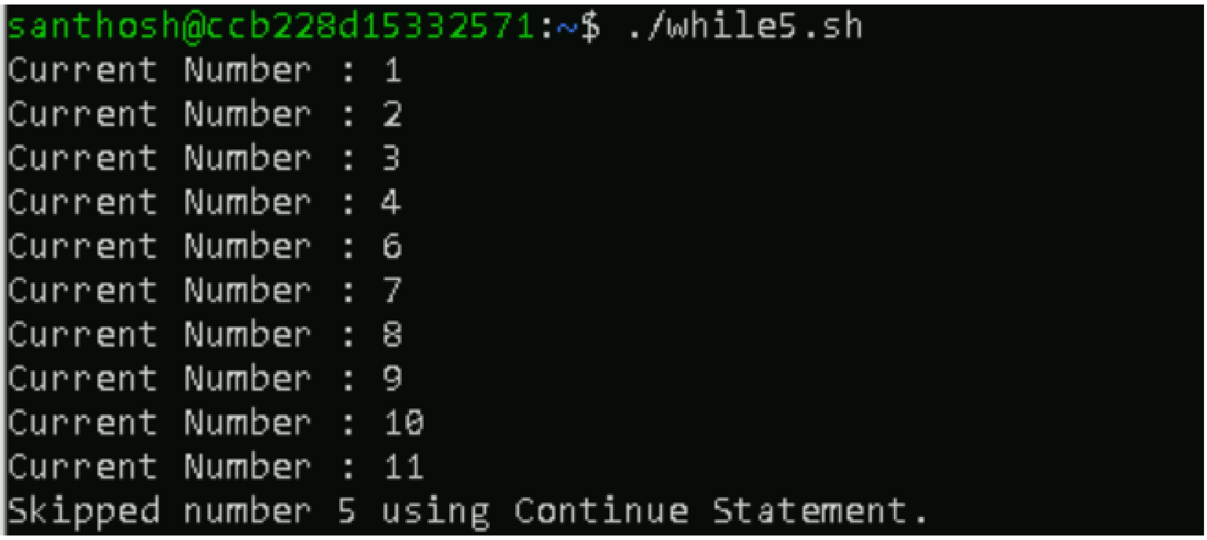
```
#!/bin/bash
# While Loop Example with a Continue Statement

i=0
while [ $i -le 10 ]
do
    ((i++))
    if [[ "$i" == 5 ]];
    then
        continue
    fi
    echo "Current Number : $i"
done

echo "Skipped number 5 using Continue Statement."
```

Step 3: Providing the necessary permissions for the ex.sh script.



```
santhosh@ccb228d15332571:~$ chmod +x while5.sh
```

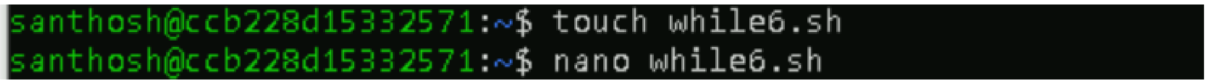
Step 4: Executing the output.



```
santhosh@ccb228d15332571:~$ ./while5.sh
Current Number : 1
Current Number : 2
Current Number : 3
Current Number : 4
Current Number : 6
Current Number : 7
Current Number : 8
Current Number : 9
Current Number : 10
Current Number : 11
Skipped number 5 using Continue Statement.
```

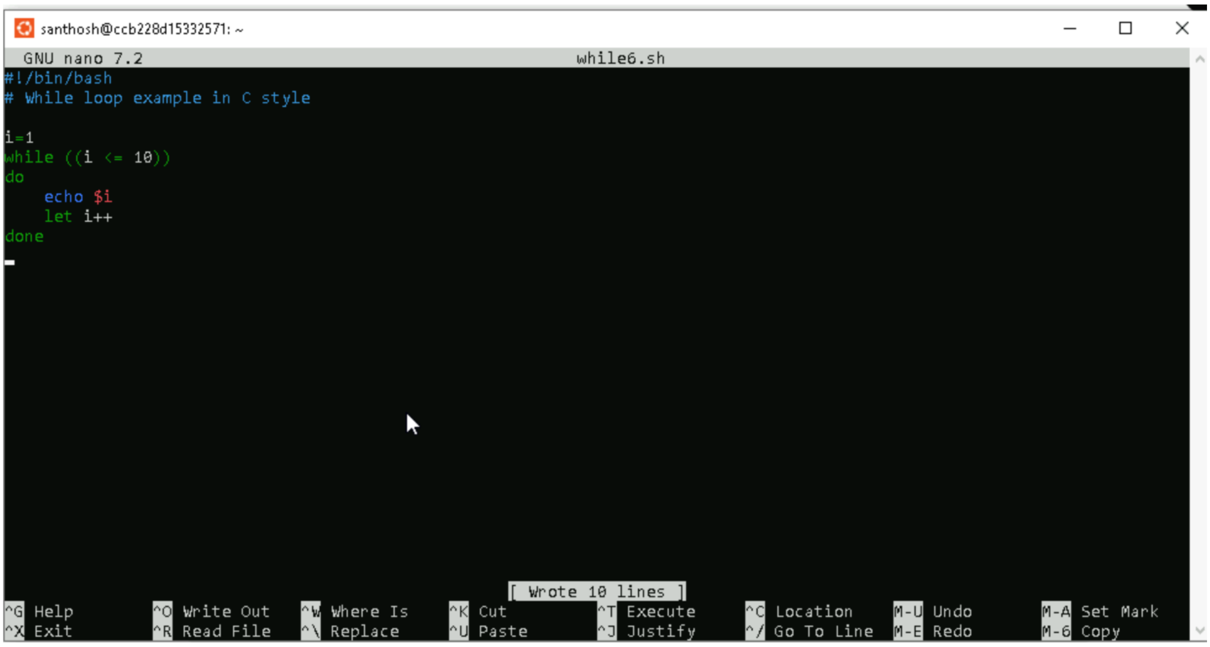
While Loop with C-Style

Step 1: Creating a bash script using touch command and adding the script bby editing the file using nano command.



```
santhosh@ccb228d15332571:~$ touch while6.sh
santhosh@ccb228d15332571:~$ nano while6.sh
```

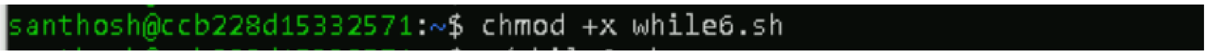
Step 2: Creating the script for a simple scenario to demonstrate the use of the case statement.



```
#!/bin/bash
# While loop example in C style

i=1
while ((i <= 10))
do
    echo $i
    let i++
done
```

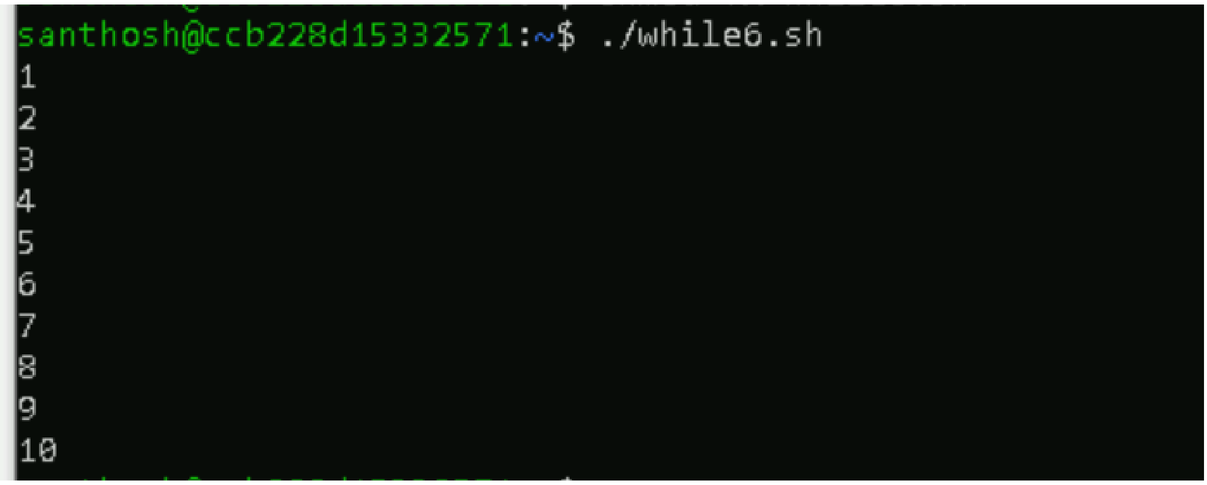
```
[ Wrote 10 lines ]
```

Step 3: Providing the necessary permissions for the ex.sh script.



```
santhosh@ccb228d15332571:~$ chmod +x while6.sh
```

Step 4: Executing the output.



```
santhosh@ccb228d15332571:~$ ./while6.sh
1
2
3
4
5
6
7
8
9
10
```