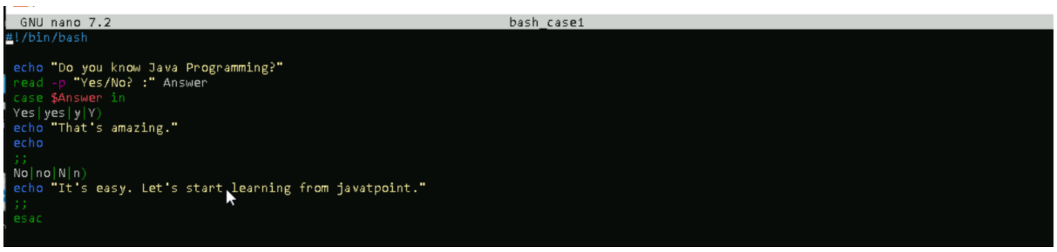
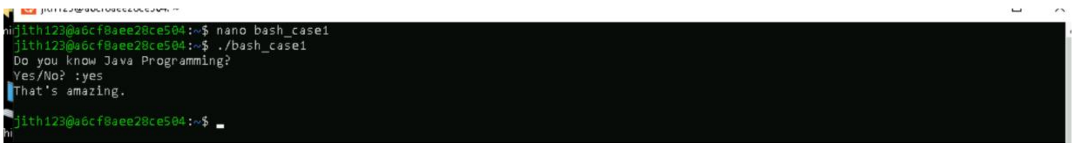
we have defined a simple scenario to demonstrate the use of the CASE statement:



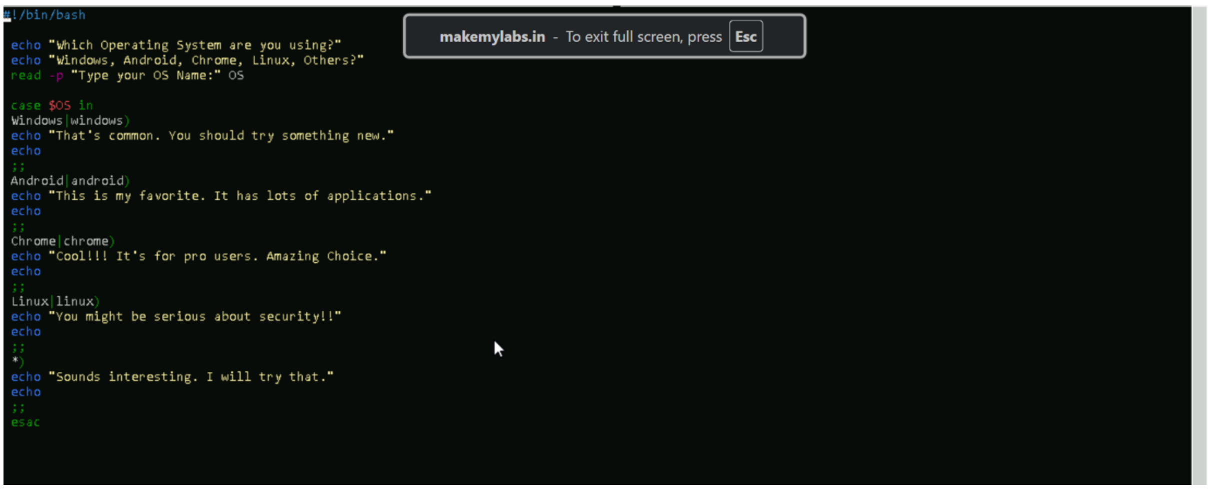
```
#!/bin/bash

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
```

OUTPUT:



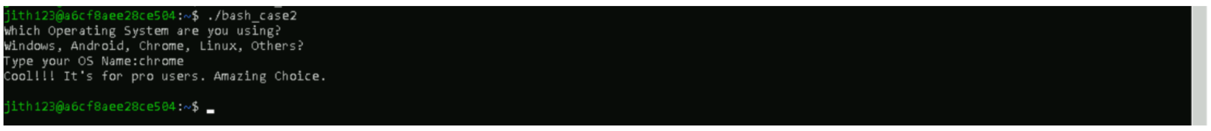
```
jith123@a6cf8aee28ce504:~$ nano bash_case1
jith123@a6cf8aee28ce504:~$ ./bash_case1
Do you know Java Programming?
Yes/No? :yes
That's amazing.

jith123@a6cf8aee28ce504:~$ _
```

In this example, we have defined a combined scenario where there is also a default case when no previous matched case is found.



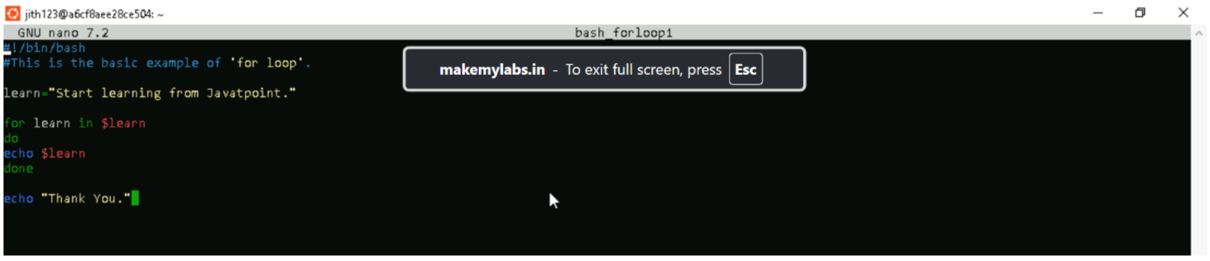
```
#!/bin/bash

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
```

OUTPUT:



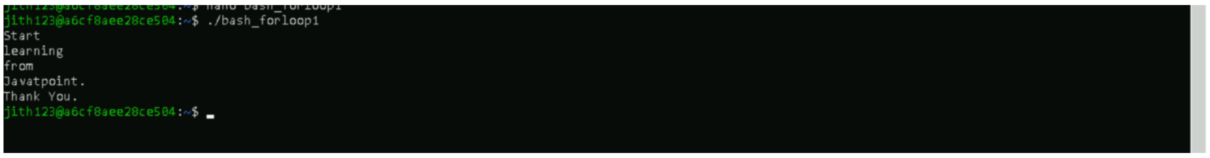
```
jith123@a6cf8aee28ce504:~$ ./bash_case2
Which Operating System are you using?
Windows, Android, Chrome, Linux, Others?
Type your OS Name:chrome
Cool!!! It's for pro users. Amazing Choice.

jith123@a6cf8aee28ce504:~$ _
```

BASIC FOR LOOP:


jith123@a6cf8aee28ce504: ~
makemylabs.in - To exit full screen, press Esc

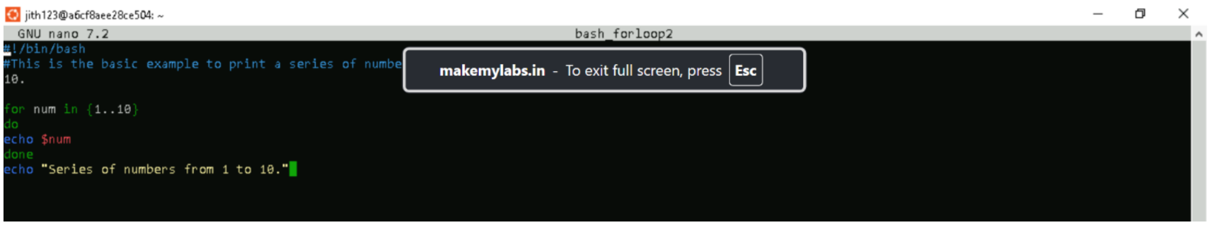
```
GNU nano 7.2                                   bash_forloop1
#!/bin/bash
#This is the basic example of 'for loop'.

learn="Start learning from Javatpoint."

for learn in $learn
do
echo $learn
done

echo "Thank You."
```

OUTPUT:



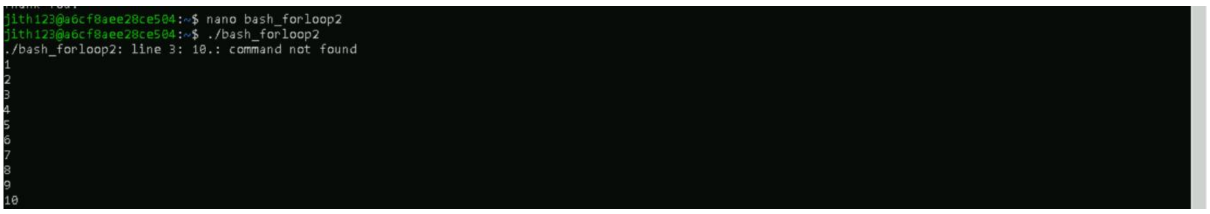
```
jith123@a6cf8aee28ce504:~$ ./bash_forloop1
Start
learning
from
Javatpoint.
Thank You.
jith123@a6cf8aee28ce504:~$ _
```

PRINT SERIES OF NUMBERS FROM A RANGE GIVEN:


jith123@a6cf8aee28ce504: ~
makemylabs.in - To exit full screen, press Esc

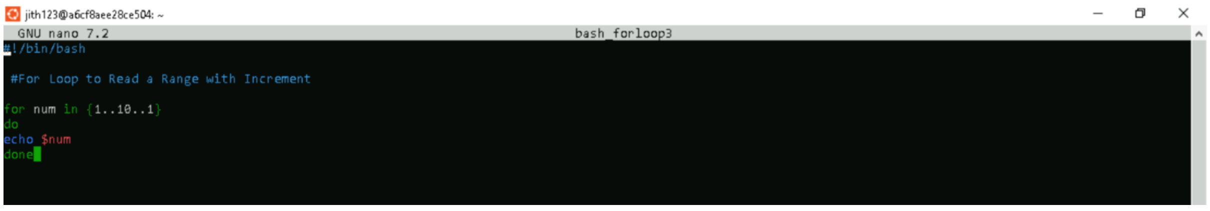
```
GNU nano 7.2                                   bash_forloop2
#!/bin/bash
#This is the basic example to print a series of numbe
10.

for num in {1..10}
do
echo $num
done
echo "Series of numbers from 1 to 10."
```

OUTPUT:



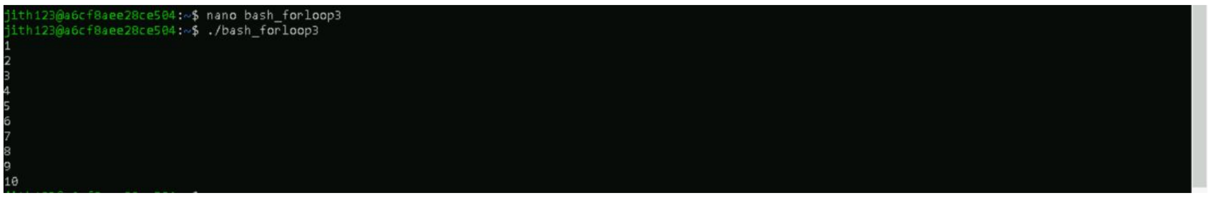
```
jith123@a6cf8aee28ce504:~$ nano bash_forloop2
jith123@a6cf8aee28ce504:~$ ./bash_forloop2
./bash_forloop2: line 3: 10.: command not found
1
2
3
4
5
6
7
8
9
10
```

HOW TO USE INCREMENT :



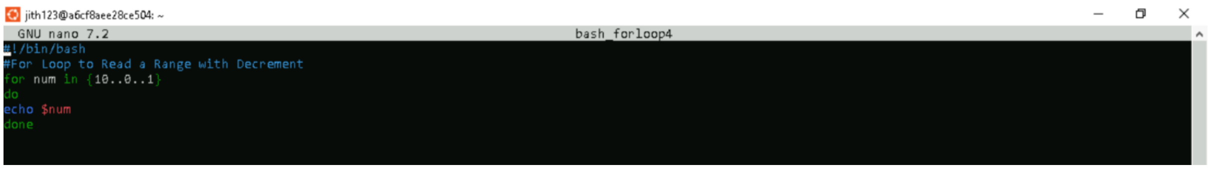
```
GNU nano 7.2                                   bash_forloop3
#!/bin/bash

 #For Loop to Read a Range with Increment

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

OUTPUT:



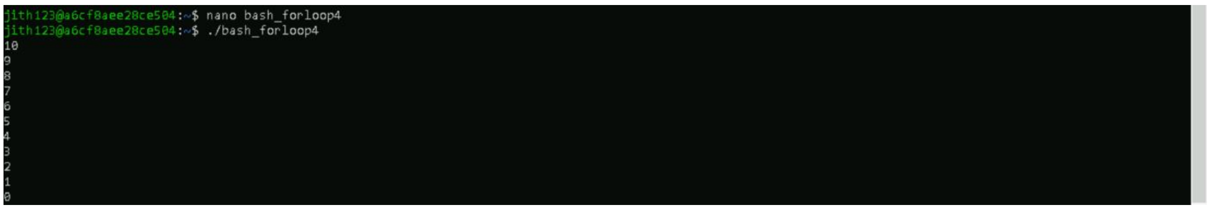
```
jith123@a6cf8aee28ce504:~$ nano bash_forloop3
jith123@a6cf8aee28ce504:~$ ./bash_forloop3
1
2
3
4
5
6
7
8
9
10
```

HOW TO USE DECREMENT:



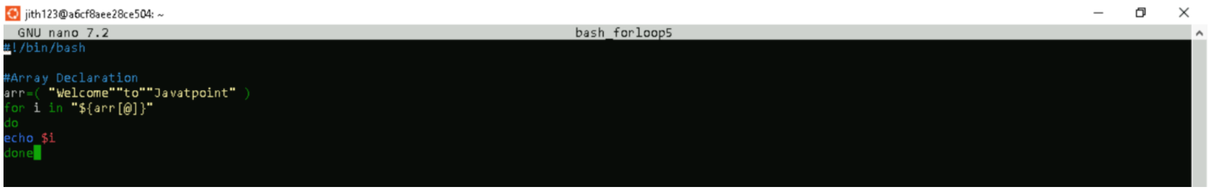
```
jith123@a6cf8aee28ce504: ~
  GNU nano 7.2                                    bash_forloop4
#!/bin/bash
#For Loop to Read a Range with Decrement
for num in {10..0..1}
do
echo $num
done
```

OUTPUT:



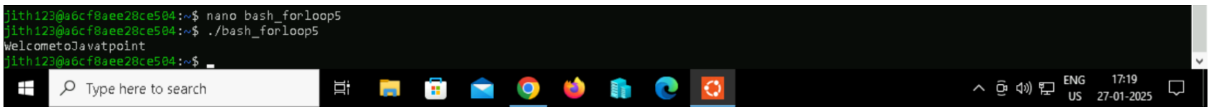
```
jith123@a6cf8aee28ce504:~$ nano bash_forloop4
jith123@a6cf8aee28ce504:~$ ./bash_forloop4
10
9
8
7
6
5
4
3
2
1
0
```

FOR LOOP TO READ RANGE OF ARRAY VARIABLES:



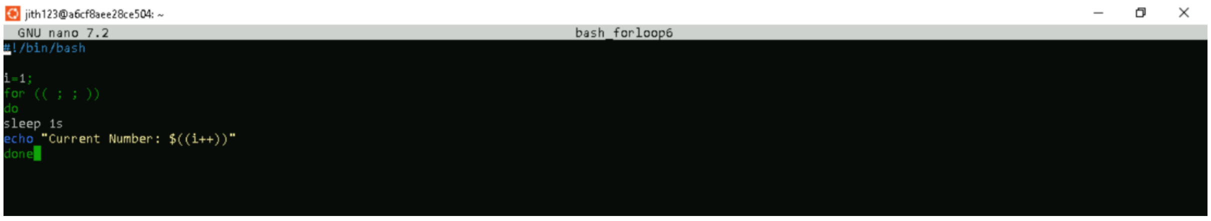
```
jith123@a6cf8aee28ce504: ~
  GNU nano 7.2                                    bash_forloop5
#!/bin/bash

#Array Declaration
arr=( "Welcome""to""Javatpoint" )
for i in "${arr[@]}"
do
echo $i
done
```

OUTPUT:



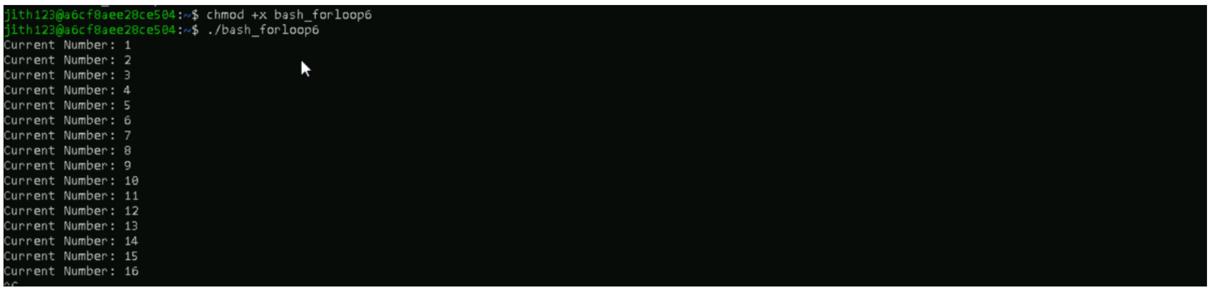
```
jith123@a6cf8aee28ce504:~$ nano bash_forloop5
jith123@a6cf8aee28ce504:~$ ./bash_forloop5
WelcometoJavatpoint
jith123@a6cf8aee28ce504:~$ _
```

How to use sleep:



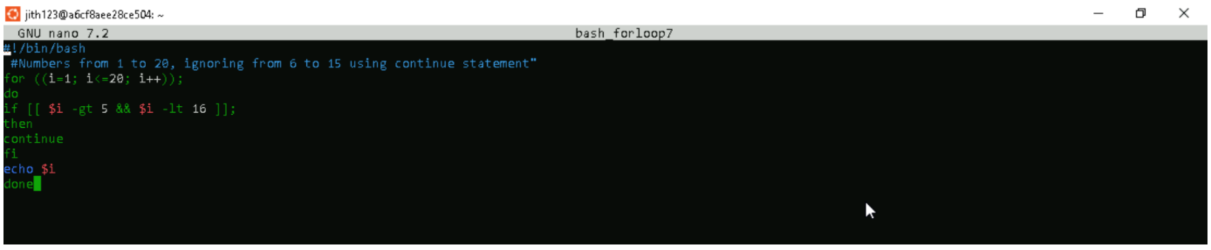
```
jith123@a6cf8aee28ce504: ~
  GNU nano 7.2                                          bash_forloop6
#!/bin/bash

i=1;
for (( ; ; ))
do
sleep 1s
echo "Current Number: $((i++))"
done
```

OUTPUT:



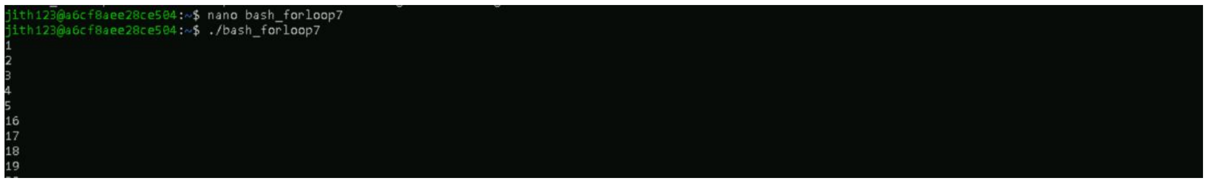
```
jith123@a6cf8aee28ce504:~$ chmod +x bash_forloop6
jith123@a6cf8aee28ce504:~$ ./bash_forloop6
Current Number: 1
Current Number: 2
Current Number: 3
Current Number: 4
Current Number: 5
Current Number: 6
Current Number: 7
Current Number: 8
Current Number: 9
Current Number: 10
Current Number: 11
Current Number: 12
Current Number: 13
Current Number: 14
Current Number: 15
Current Number: 16
```

HOW TO USE CONTINUE:



```
jith123@a6cf8aee28ce504: ~
  GNU nano 7.2                                          bash_forloop7
#!/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
```

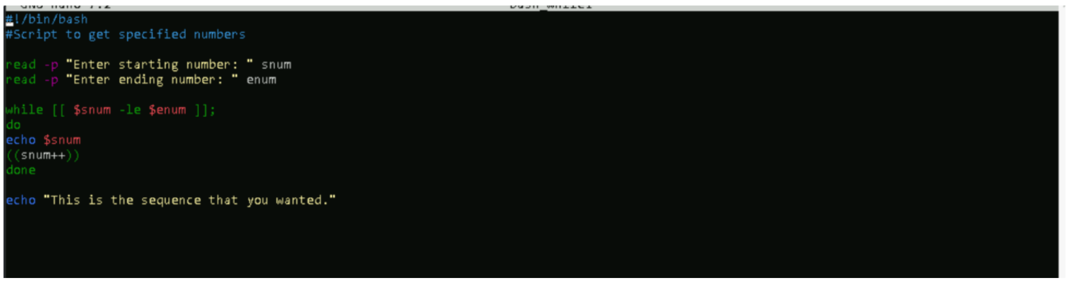
OUTPUT:



```
jith123@a6cf8aee28ce504:~$ nano bash_forloop7
jith123@a6cf8aee28ce504:~$ ./bash_forloop7
1
2
3
4
5
16
17
18
19
```

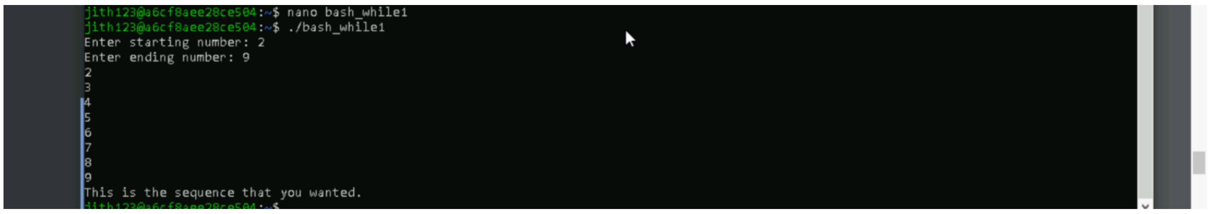
WHILE:

Simple while loop:



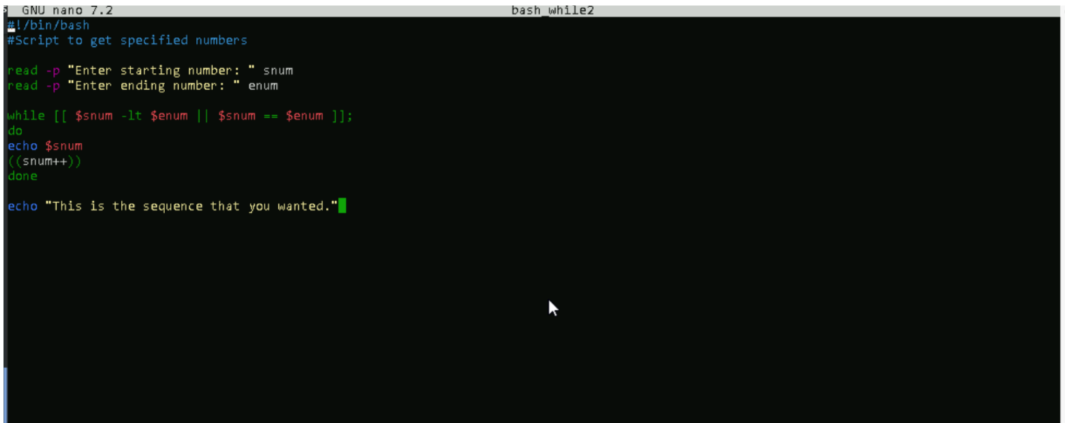
```
#!/bin/bash
#Script to get specified numbers

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."
```

Output:



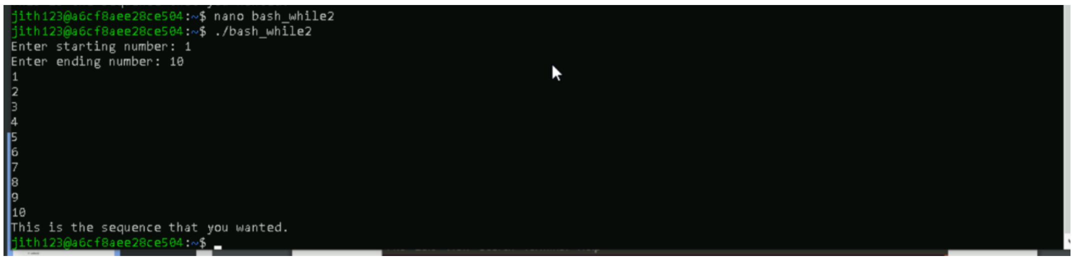
```
jith123@a6cf8aee28ce504:~$ nano bash_while1
jith123@a6cf8aee28ce504:~$ ./bash_while1
Enter starting number: 2
Enter ending number: 9
2
3
4
5
6
7
8
9
This is the sequence that you wanted.
jith123@a6cf8aee28ce504:~$
```

While loop to get specified numbers:



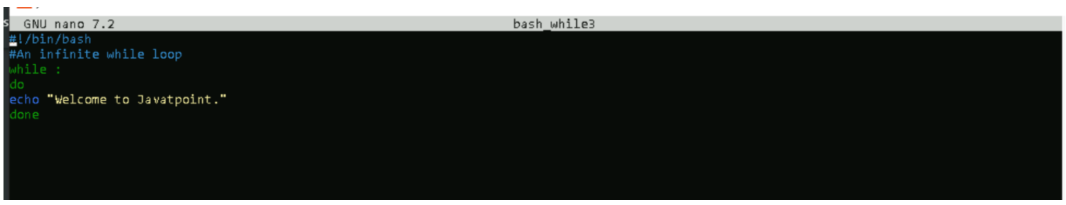
```
  GNU nano 7.2                                          bash_while2
#!/bin/bash
#Script to get specified numbers

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

while [[ $snum -lt $enum || $snum == $enum ]];
do
echo $snum
((snum++))
done

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

Output:



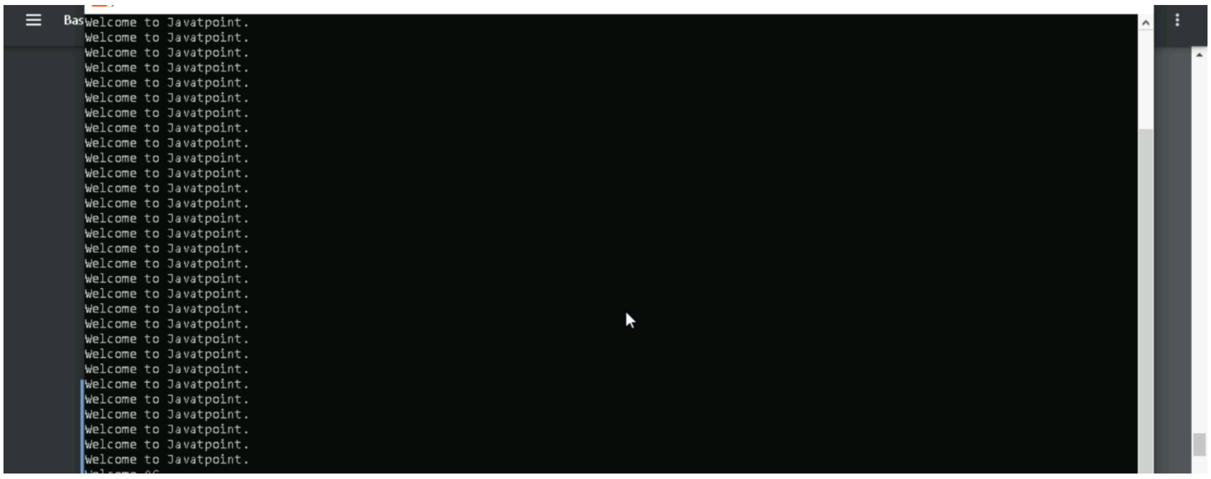
```
jith123@a6cf8aee28ce504:~$ nano bash_while2
jith123@a6cf8aee28ce504:~$ ./bash_while2
Enter starting number: 1
Enter ending number: 10
1
2
3
4
5
6
7
8
9
10
This is the sequence that you wanted.
jith123@a6cf8aee28ce504:~$ _
```

Infinite while loop:



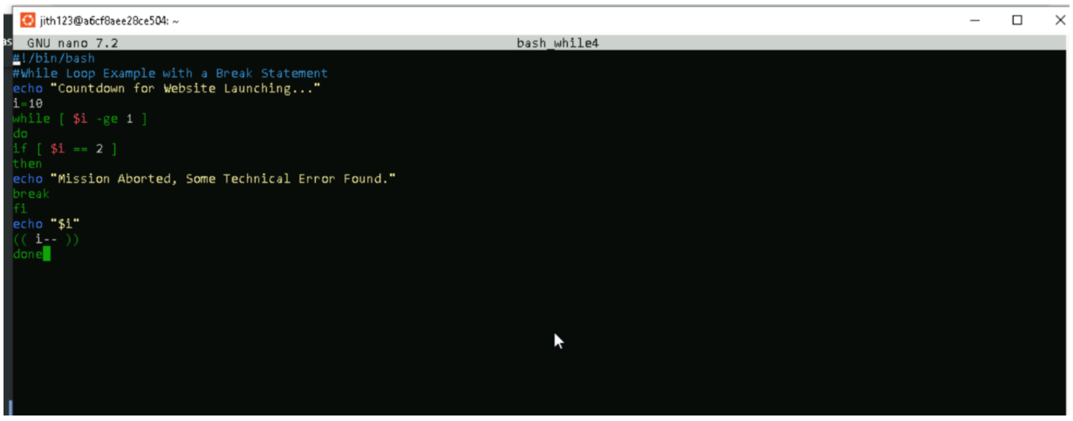
```
  GNU nano 7.2                              bash_while3
#!/bin/bash
#An infinite while loop
while :
do
echo "Welcome to Javatpoint."
done
```

Output:



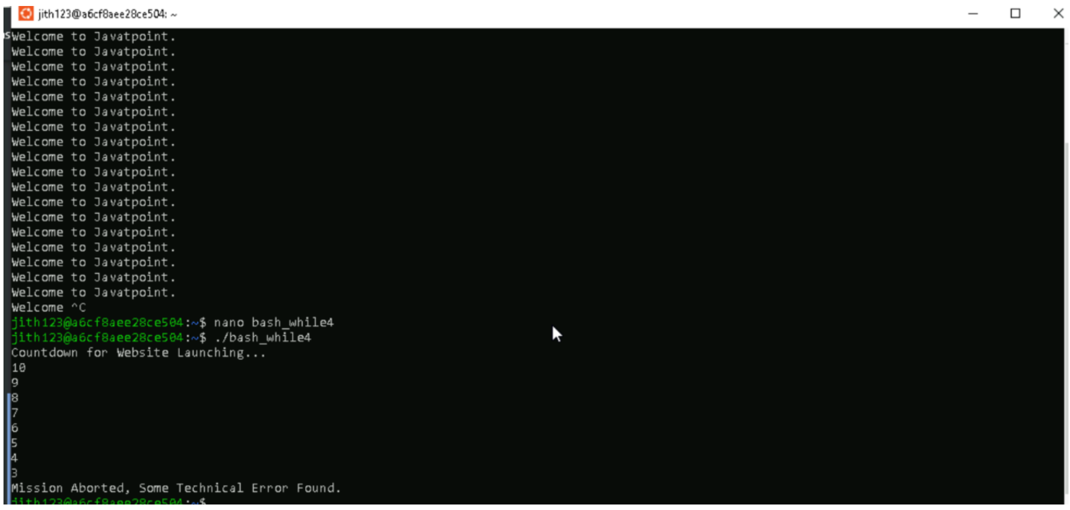
```
Welcome to Javatpoint.
Welcome to Javatpoint.
Welcome to Javatpoint.
Welcome to Javatpoint.
Welcome to Javatpoint.
Welcome to Javatpoint.
Welcome to Javatpoint.
Welcome to Javatpoint.
Welcome to Javatpoint.
Welcome to Javatpoint.
Welcome to Javatpoint.
Welcome to Javatpoint.
Welcome to Javatpoint.
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.
Welcome to Javatpoint.
Welcome to Javatpoint.
Welcome to Javatpoint.
Welcome to Javatpoint.
Welcome to Javatpoint.
Welcome to Javatpoint.
Welcome to Javatpoint.
Welcome to Javatpoint.
```

Example with break statement:



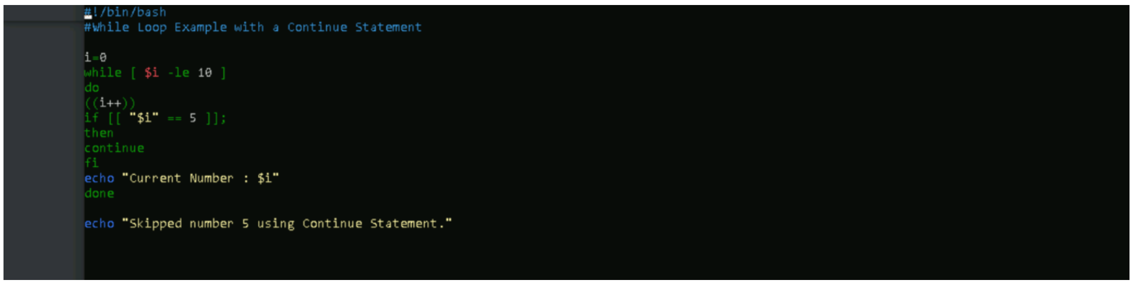
```
jith123@a6cf8aee28ce504: ~
  GNU nano 7.2                              bash_while4
#!/bin/bash
#While Loop Example with a Break Statement
echo "Countdown for Website Launching..."
i=10
while [ $i -ge 1 ]
do
if [ $i == 2 ]
then
echo "Mission Aborted, Some Technical Error Found."
break
fi
echo "$i"
(( i-- ))
done
```

Output:



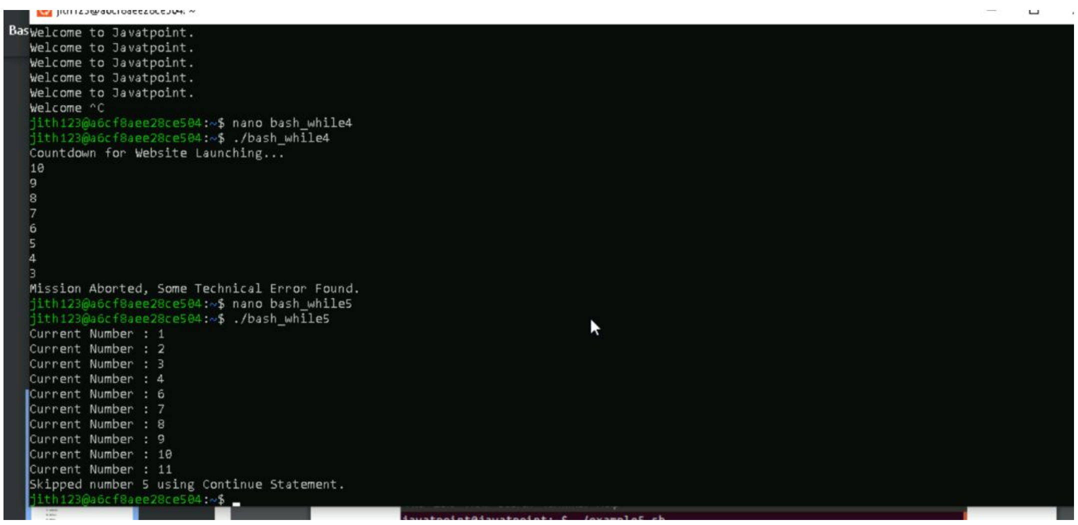
```
jith123@a6cf8aee28ce504: ~
Welcome to Javatpoint.
Welcome to Javatpoint.
Welcome to Javatpoint.
Welcome to Javatpoint.
Welcome to Javatpoint.
Welcome to Javatpoint.
Welcome to Javatpoint.
Welcome to Javatpoint.
Welcome to Javatpoint.
Welcome to Javatpoint.
Welcome to Javatpoint.
Welcome to Javatpoint.
Welcome to Javatpoint.
Welcome to Javatpoint.
Welcome to Javatpoint.
Welcome to Javatpoint.
Welcome to Javatpoint.
Welcome to Javatpoint.
Welcome ^C
jith123@a6cf8aee28ce504:~$ nano bash_while4
jith123@a6cf8aee28ce504:~$ ./bash_while4
Countdown for Website Launching...
10
9
8
7
6
5
4
3
Mission Aborted, Some Technical Error Found.
jith123@a6cf8aee28ce504:~$
```

Example with continue 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."
```

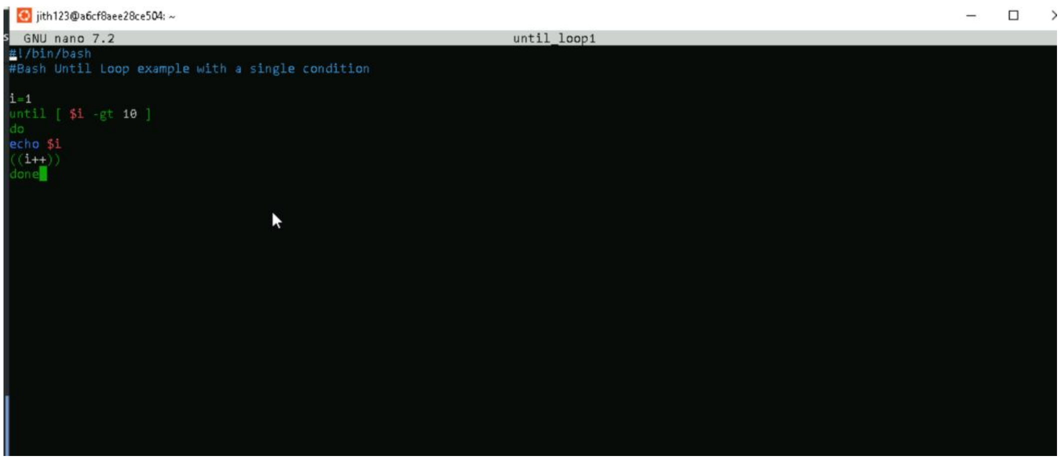
Output:



```
Welcome to Javatpoint.
Welcome to Javatpoint.
Welcome to Javatpoint.
Welcome to Javatpoint.
Welcome to Javatpoint.
Welcome ^C
jith123@a6cf8aee28ce504:~$ nano bash_while4
jith123@a6cf8aee28ce504:~$ ./bash_while4
Countdown for Website Launching...
10
9
8
7
6
5
4
3
Mission Aborted, Some Technical Error Found.
jith123@a6cf8aee28ce504:~$ nano bash_while5
jith123@a6cf8aee28ce504:~$ ./bash_while5
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.
jith123@a6cf8aee28ce504:~$
```

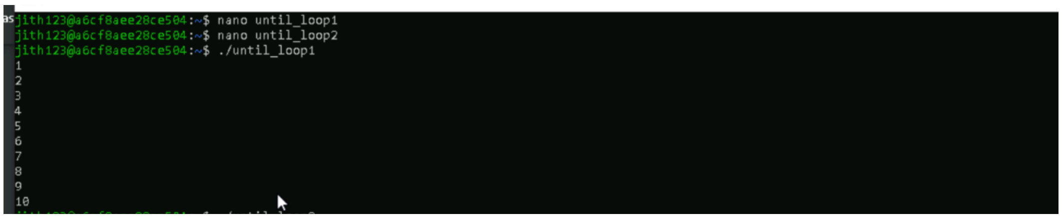
Untill:

Untill loop with single condition:



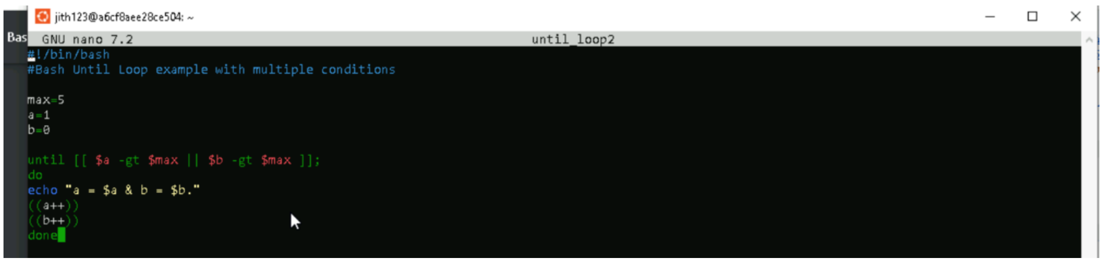
```
jith123@a6cf8aee28ce504: ~
  GNU nano 7.2                          until_loop1
#!/bin/bash
#Bash Until Loop example with a single condition

i=1
until [ $i -gt 10 ]
do
echo $i
((i++))
done
```

Output:



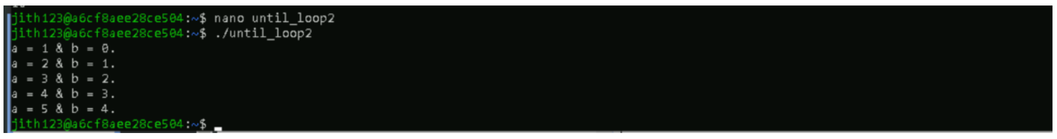
```
jith123@a6cf8aee28ce504:~$ nano until_loop1
jith123@a6cf8aee28ce504:~$ nano until_loop2
jith123@a6cf8aee28ce504:~$ ./until_loop1
1
2
3
4
5
6
7
8
9
10
```

Untill loop with multiple conditions:



```
jith123@a6cf8aee28ce504: ~
  GNU nano 7.2                          until_loop2
#!/bin/bash
#Bash Until Loop example with multiple conditions

max=5
a=1
b=0

until [[ $a -gt $max || $b -gt $max ]];
do
echo "a = $a & b = $b."
((a++))
((b++))
done
```

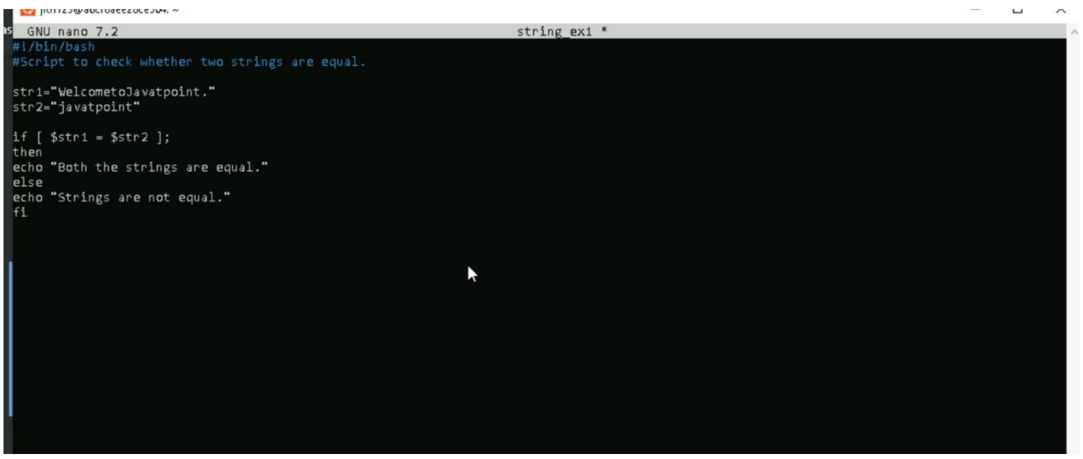
Output:



```
jith123@a6cf8aee28ce504:~$ nano until_loop2
jith123@a6cf8aee28ce504:~$ ./until_loop2
a = 1 & b = 0.
a = 2 & b = 1.
a = 3 & b = 2.
a = 4 & b = 3.
a = 5 & b = 4.
jith123@a6cf8aee28ce504:~$
```

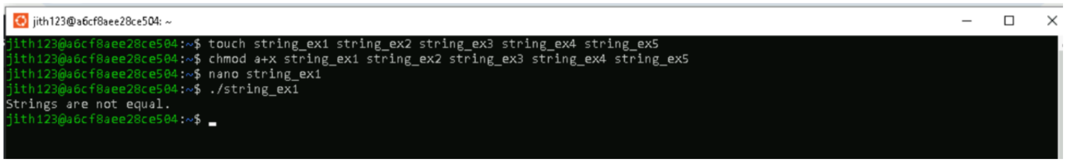
String:

Equal operator:



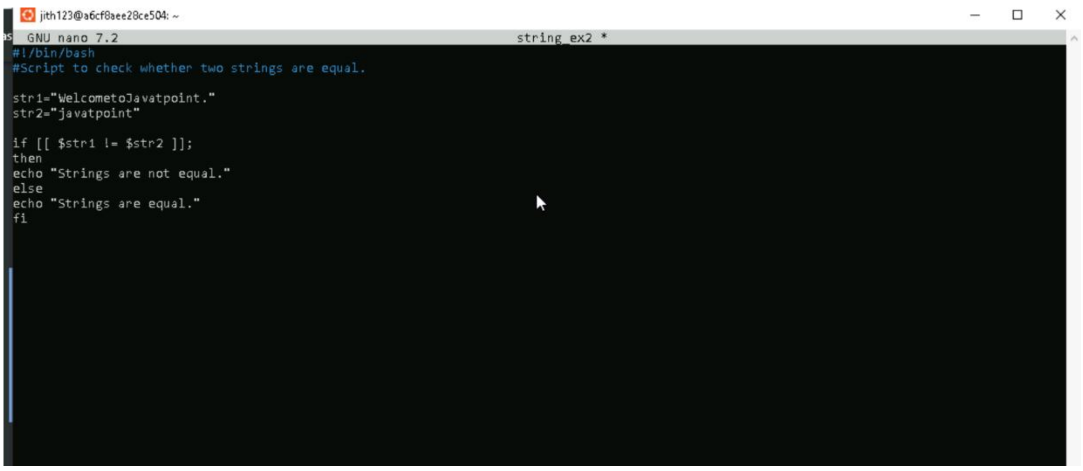
```
GNU nano 7.2                                    string_ex1 *
#!/bin/bash
#Script to check whether two strings are equal.

str1="WelcometoJavatpoint."
str2="javatpoint"

if [ $str1 = $str2 ];
then
echo "Both the strings are equal."
else
echo "Strings are not equal."
fi
```

Output:



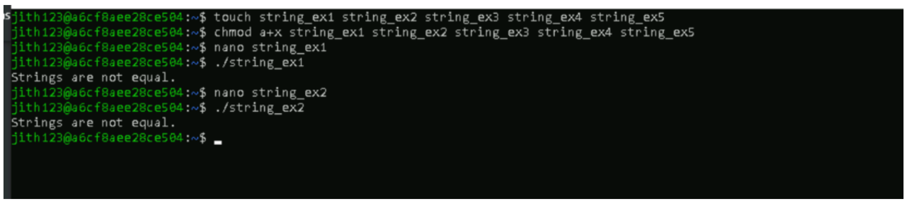
```
jith123@a6cf8aee28ce504: ~
jith123@a6cf8aee28ce504:~$ touch string_ex1 string_ex2 string_ex3 string_ex4 string_ex5
jith123@a6cf8aee28ce504:~$ chmod a+x string_ex1 string_ex2 string_ex3 string_ex4 string_ex5
jith123@a6cf8aee28ce504:~$ nano string_ex1
jith123@a6cf8aee28ce504:~$ ./string_ex1
Strings are not equal.
jith123@a6cf8aee28ce504:~$ _
```

Not equal operator:



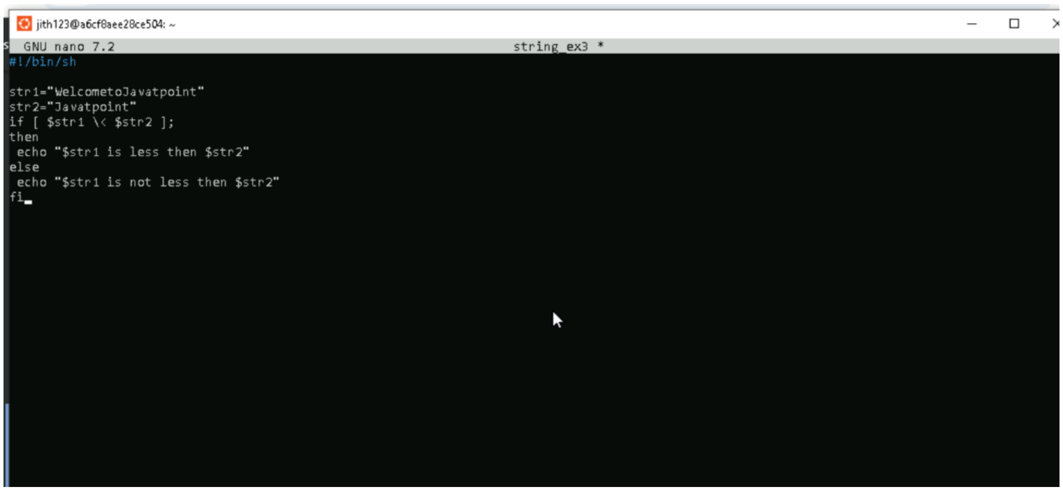
```
jith123@a6cf8aee28ce504: ~
GNU nano 7.2                                    string_ex2 *
#!/bin/bash
#Script to check whether two strings are equal.

str1="WelcometoJavatpoint."
str2="javatpoint"

if [[ $str1 != $str2 ]];
then
echo "Strings are not equal."
else
echo "Strings are equal."
fi
```

Ouput:



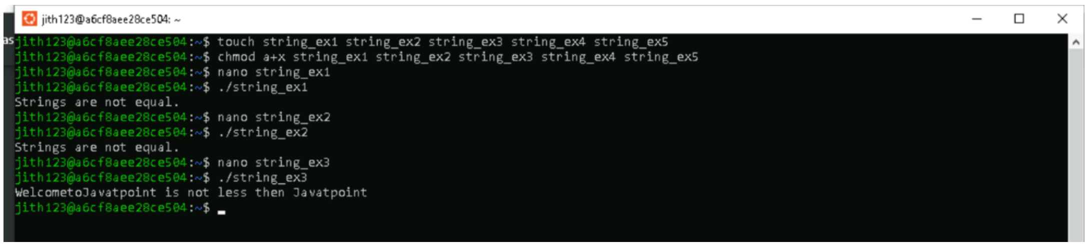
```
jith123@a6cf8aee28ce504:~$ touch string_ex1 string_ex2 string_ex3 string_ex4 string_ex5
jith123@a6cf8aee28ce504:~$ chmod a+x string_ex1 string_ex2 string_ex3 string_ex4 string_ex5
jith123@a6cf8aee28ce504:~$ nano string_ex1
jith123@a6cf8aee28ce504:~$ ./string_ex1
Strings are not equal.
jith123@a6cf8aee28ce504:~$ nano string_ex2
jith123@a6cf8aee28ce504:~$ ./string_ex2
Strings are not equal.
jith123@a6cf8aee28ce504:~$ _
```

Less than operator:



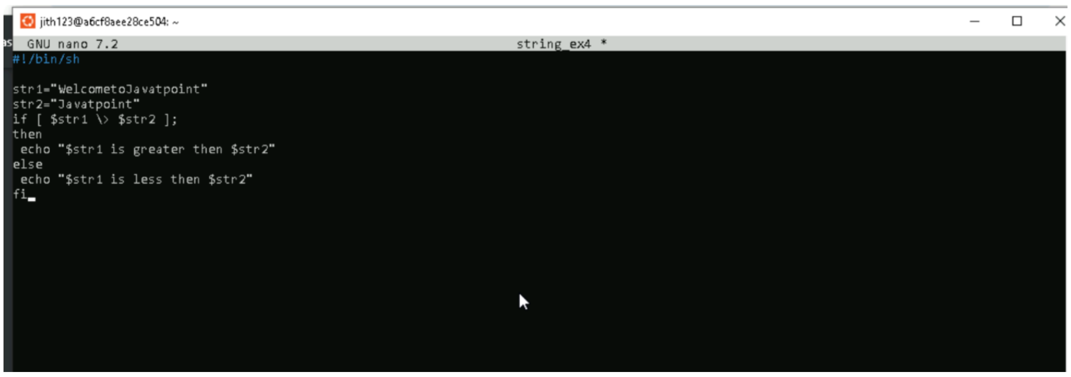
```
jith123@a6cf8aee28ce504: ~
  GNU nano 7.2                                    string_ex3 *
#!/bin/sh

str1="WelcometoJavatpoint"
str2="Javatpoint"
if [ $str1 \< $str2 ];
then
 echo "$str1 is less then $str2"
else
 echo "$str1 is not less then $str2"
fi
```

Output:



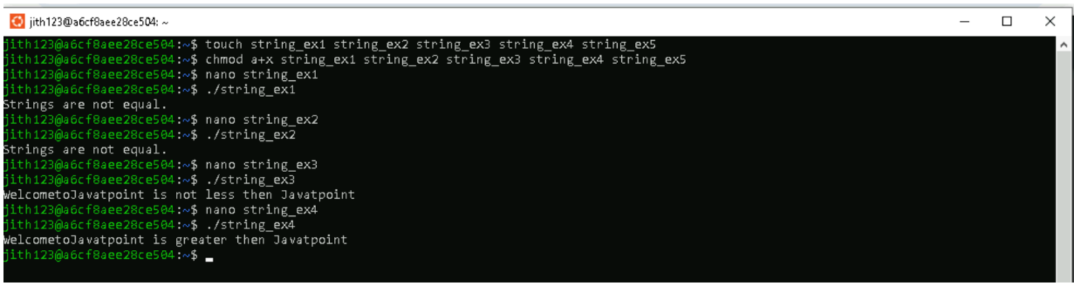
```
jith123@a6cf8aee28ce504: ~
jith123@a6cf8aee28ce504:~$ touch string_ex1 string_ex2 string_ex3 string_ex4 string_ex5
jith123@a6cf8aee28ce504:~$ chmod a+x string_ex1 string_ex2 string_ex3 string_ex4 string_ex5
jith123@a6cf8aee28ce504:~$ nano string_ex1
jith123@a6cf8aee28ce504:~$ ./string_ex1
Strings are not equal.
jith123@a6cf8aee28ce504:~$ nano string_ex2
jith123@a6cf8aee28ce504:~$ ./string_ex2
Strings are not equal.
jith123@a6cf8aee28ce504:~$ nano string_ex3
jith123@a6cf8aee28ce504:~$ ./string_ex3
WelcometoJavatpoint is not less then Javatpoint
jith123@a6cf8aee28ce504:~$ _
```

Greater than operator:



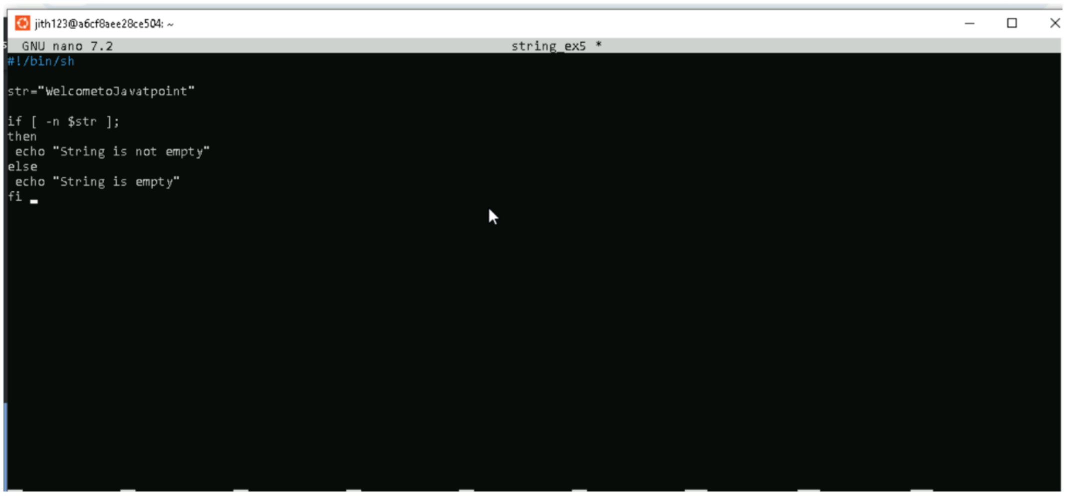
```
jith123@a6cf8aee28ce504: ~
  GNU nano 7.2                                    string_ex4 *
#!/bin/sh

str1="WelcometoJavatpoint"
str2="Javatpoint"
if [ $str1 \> $str2 ];
then
 echo "$str1 is greater then $str2"
else
 echo "$str1 is less then $str2"
fi
```

Output:



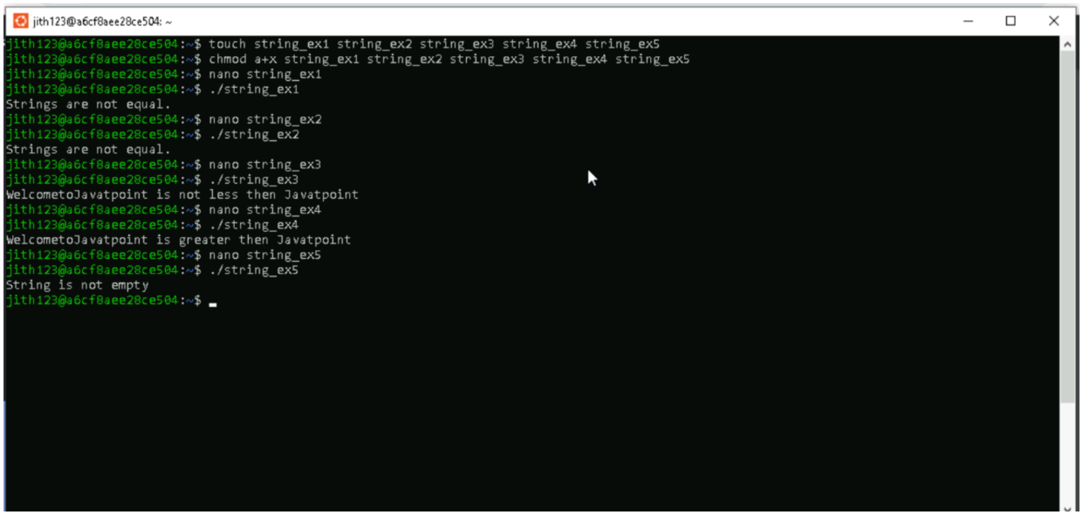
```
jith123@a6cf8aee28ce504: ~
jith123@a6cf8aee28ce504:~$ touch string_ex1 string_ex2 string_ex3 string_ex4 string_ex5
jith123@a6cf8aee28ce504:~$ chmod a+x string_ex1 string_ex2 string_ex3 string_ex4 string_ex5
jith123@a6cf8aee28ce504:~$ nano string_ex1
jith123@a6cf8aee28ce504:~$ ./string_ex1
Strings are not equal.
jith123@a6cf8aee28ce504:~$ nano string_ex2
jith123@a6cf8aee28ce504:~$ ./string_ex2
Strings are not equal.
jith123@a6cf8aee28ce504:~$ nano string_ex3
jith123@a6cf8aee28ce504:~$ ./string_ex3
WelcometoJavatpoint is not less then Javatpoint
jith123@a6cf8aee28ce504:~$ nano string_ex4
jith123@a6cf8aee28ce504:~$ ./string_ex4
WelcometoJavatpoint is greater then Javatpoint
jith123@a6cf8aee28ce504:~$
```

String length is greator than zero:



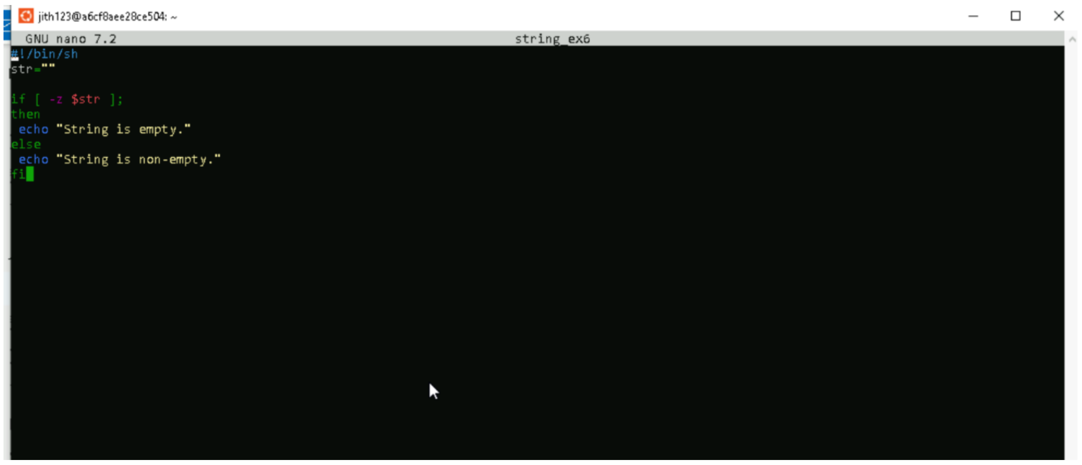
```
jith123@a6cf8aee28ce504: ~
  GNU nano 7.2                                    string_ex5 *
#!/bin/sh

str="WelcometoJavatpoint"

if [ -n $str ];
then
 echo "String is not empty"
else
 echo "String is empty"
fi
```

Output:



```
jith123@a6cf8aee28ce504: ~
jith123@a6cf8aee28ce504:~$ touch string_ex1 string_ex2 string_ex3 string_ex4 string_ex5
jith123@a6cf8aee28ce504:~$ chmod a+x string_ex1 string_ex2 string_ex3 string_ex4 string_ex5
jith123@a6cf8aee28ce504:~$ nano string_ex1
jith123@a6cf8aee28ce504:~$ ./string_ex1
Strings are not equal.
jith123@a6cf8aee28ce504:~$ nano string_ex2
jith123@a6cf8aee28ce504:~$ ./string_ex2
Strings are not equal.
jith123@a6cf8aee28ce504:~$ nano string_ex3
jith123@a6cf8aee28ce504:~$ ./string_ex3
WelcometoJavatpoint is not less then Javatpoint
jith123@a6cf8aee28ce504:~$ nano string_ex4
jith123@a6cf8aee28ce504:~$ ./string_ex4
WelcometoJavatpoint is greater then Javatpoint
jith123@a6cf8aee28ce504:~$ nano string_ex5
jith123@a6cf8aee28ce504:~$ ./string_ex5
String is not empty
jith123@a6cf8aee28ce504:~$
```

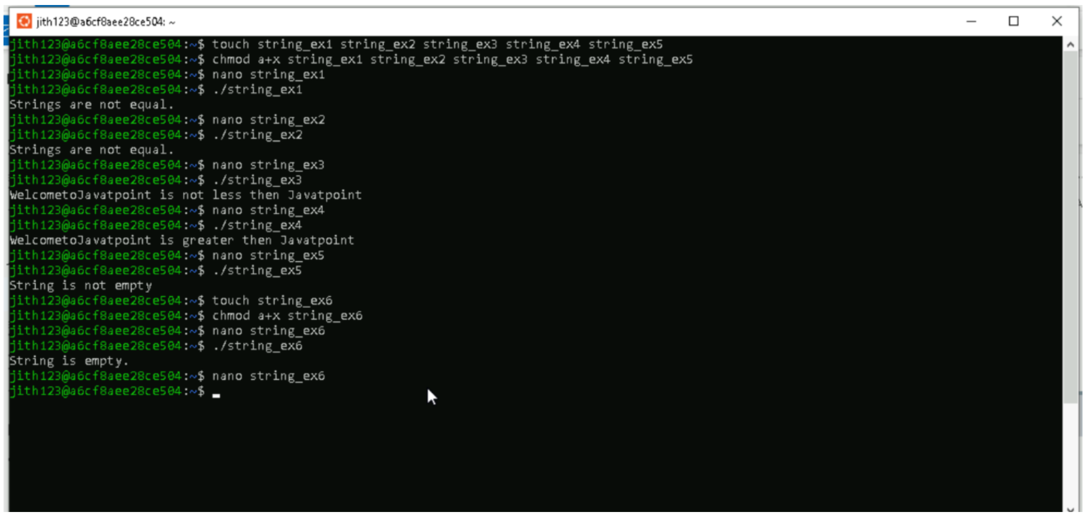
String length is equal to zero:



jith123@a6cf8aee28ce504: ~

```
  GNU nano 7.2                          string_ex6
#!/bin/sh
str=""

if [ -z $str ];
then
 echo "String is empty."
else
 echo "String is non-empty."
fi
```

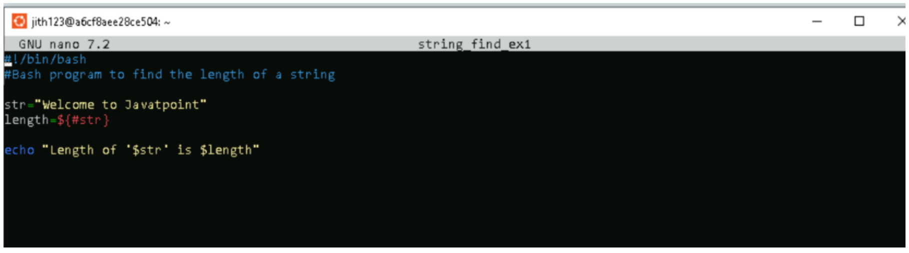
Output:



jith123@a6cf8aee28ce504: ~

```
jith123@a6cf8aee28ce504:~$ touch string_ex1 string_ex2 string_ex3 string_ex4 string_ex5
jith123@a6cf8aee28ce504:~$ chmod a+x string_ex1 string_ex2 string_ex3 string_ex4 string_ex5
jith123@a6cf8aee28ce504:~$ nano string_ex1
jith123@a6cf8aee28ce504:~$ ./string_ex1
Strings are not equal.
jith123@a6cf8aee28ce504:~$ nano string_ex2
jith123@a6cf8aee28ce504:~$ ./string_ex2
Strings are not equal.
jith123@a6cf8aee28ce504:~$ nano string_ex3
jith123@a6cf8aee28ce504:~$ ./string_ex3
WelcometoJavatpoint is not less then Javatpoint
jith123@a6cf8aee28ce504:~$ nano string_ex4
jith123@a6cf8aee28ce504:~$ ./string_ex4
WelcometoJavatpoint is greater then Javatpoint
jith123@a6cf8aee28ce504:~$ nano string_ex5
jith123@a6cf8aee28ce504:~$ ./string_ex5
String is not empty
jith123@a6cf8aee28ce504:~$ touch string_ex6
jith123@a6cf8aee28ce504:~$ chmod a+x string_ex6
jith123@a6cf8aee28ce504:~$ nano string_ex6
jith123@a6cf8aee28ce504:~$ ./string_ex6
String is empty.
jith123@a6cf8aee28ce504:~$ nano string_ex6
jith123@a6cf8aee28ce504:~$ _
```

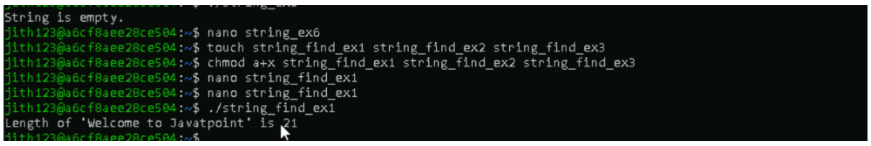
To find the length of the string:



jith123@a6cf8aee28ce504: ~

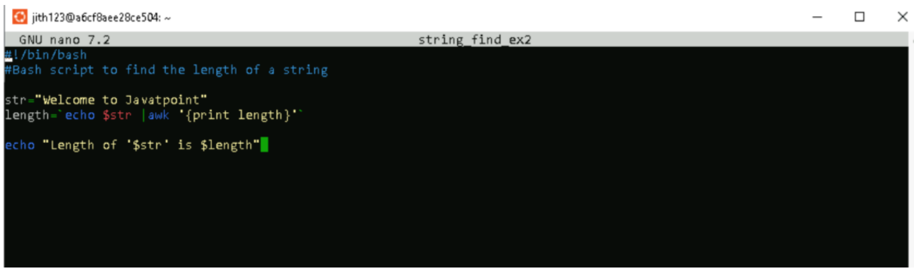
```
  GNU nano 7.2                          string_find_ex1
#!/bin/bash
#Bash program to find the length of a string

str="Welcome to Javatpoint"
length=${#str}

echo "Length of '$str' is $length"
```

Output:



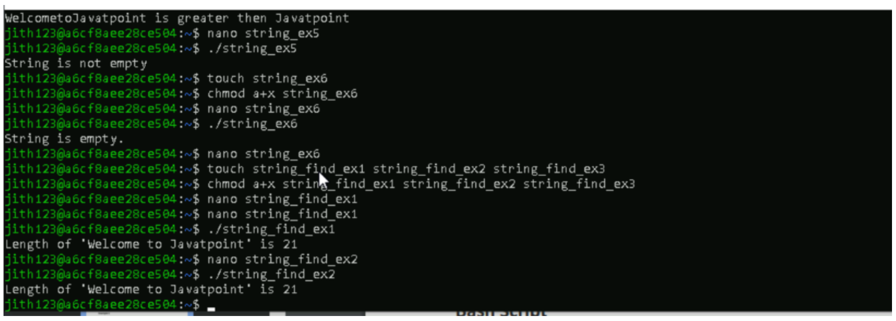
```
String is empty.
jith123@a6cf8aee28ce504:~$ nano string_ex6
jith123@a6cf8aee28ce504:~$ touch string_find_ex1 string_find_ex2 string_find_ex3
jith123@a6cf8aee28ce504:~$ chmod a+x string_find_ex1 string_find_ex2 string_find_ex3
jith123@a6cf8aee28ce504:~$ nano string_find_ex1
jith123@a6cf8aee28ce504:~$ nano string_find_ex1
jith123@a6cf8aee28ce504:~$ ./string_find_ex1
Length of 'Welcome to Javatpoint' is 21
jith123@a6cf8aee28ce504:~$
```

Find the length of the string using “awk” :



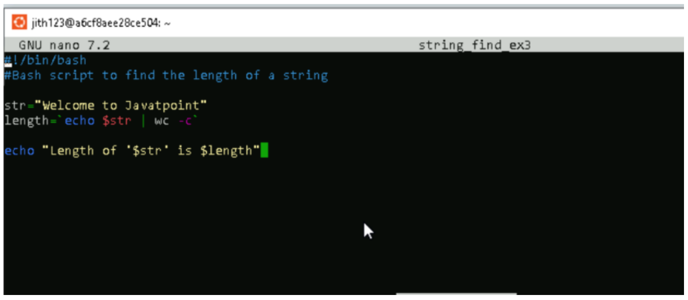
```
jith123@a6cf8aee28ce504: ~
  GNU nano 7.2                                              string_find_ex2
#!/bin/bash
#Bash script to find the length of a string

str="Welcome to Javatpoint"
length=`echo $str |awk '{print length}'`

echo "Length of '$str' is $length"
```

Output:



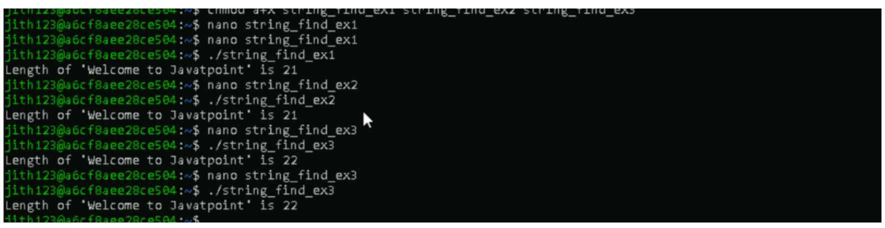
```
WelcometoJavatpoint is greater then Javatpoint
jith123@a6cf8aee28ce504:~$ nano string_ex5
jith123@a6cf8aee28ce504:~$ ./string_ex5
String is not empty
jith123@a6cf8aee28ce504:~$ touch string_ex6
jith123@a6cf8aee28ce504:~$ chmod a+x string_ex6
jith123@a6cf8aee28ce504:~$ nano string_ex6
jith123@a6cf8aee28ce504:~$ ./string_ex6
String is empty.
jith123@a6cf8aee28ce504:~$ nano string_ex6
jith123@a6cf8aee28ce504:~$ touch string_find_ex1 string_find_ex2 string_find_ex3
jith123@a6cf8aee28ce504:~$ chmod a+x string_find_ex1 string_find_ex2 string_find_ex3
jith123@a6cf8aee28ce504:~$ nano string_find_ex1
jith123@a6cf8aee28ce504:~$ nano string_find_ex1
jith123@a6cf8aee28ce504:~$ ./string_find_ex1
Length of 'Welcome to Javatpoint' is 21
jith123@a6cf8aee28ce504:~$ nano string_find_ex2
jith123@a6cf8aee28ce504:~$ ./string_find_ex2
Length of 'Welcome to Javatpoint' is 21
jith123@a6cf8aee28ce504:~$ _
```

To find the length of the string using “wc” :



```
jith123@a6cf8aee28ce504: ~
  GNU nano 7.2                                              string_find_ex3
#!/bin/bash
#Bash script to find the length of a string

str="Welcome to Javatpoint"
length=`echo $str | wc -c`

echo "Length of '$str' is $length"
```

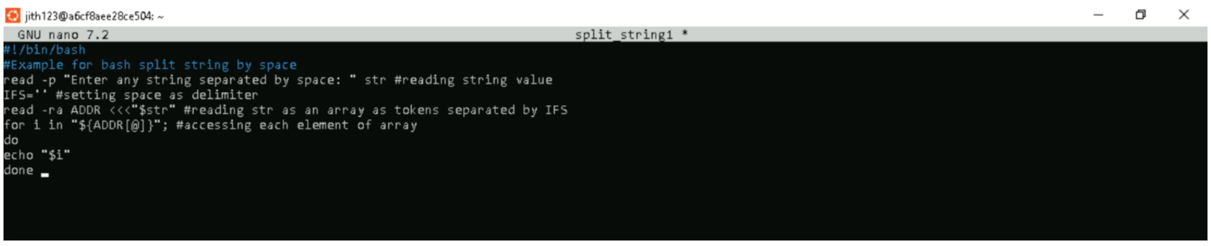
Output:



```
jith123@a6cf8aee28ce504:~$ chmod a+x string_find_ex1 string_find_ex2 string_find_ex3
jith123@a6cf8aee28ce504:~$ nano string_find_ex1
jith123@a6cf8aee28ce504:~$ nano string_find_ex1
jith123@a6cf8aee28ce504:~$ ./string_find_ex1
Length of 'Welcome to Javatpoint' is 21
jith123@a6cf8aee28ce504:~$ nano string_find_ex2
jith123@a6cf8aee28ce504:~$ ./string_find_ex2
Length of 'Welcome to Javatpoint' is 21
jith123@a6cf8aee28ce504:~$ nano string_find_ex3
jith123@a6cf8aee28ce504:~$ ./string_find_ex3
Length of 'Welcome to Javatpoint' is 22
jith123@a6cf8aee28ce504:~$ nano string_find_ex3
jith123@a6cf8aee28ce504:~$ ./string_find_ex3
Length of 'Welcome to Javatpoint' is 22
jith123@a6cf8aee28ce504:~$
```

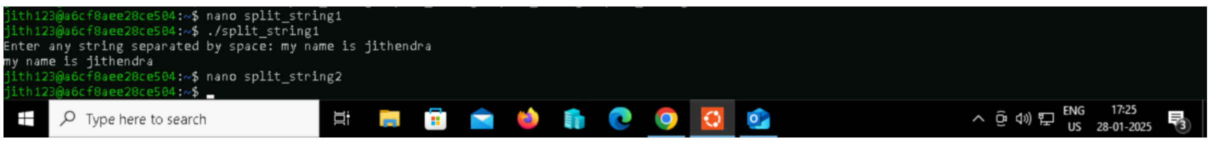
BASH Split string:

Split string by space:



```
jith123@a6cf8aee28ce504: ~
  GNU nano 7.2                                           split_string1 *
#!/bin/bash
#Example for bash split string by space
read -p "Enter any string separated by space: " str #reading string value
IFS='' #setting space as delimiter
read -ra ADDR <<<"$str" #reading str as an array as tokens separated by IFS
for i in "${ADDR[@]}"; #accessing each element of array
do
echo "$i"
done
```

Output:

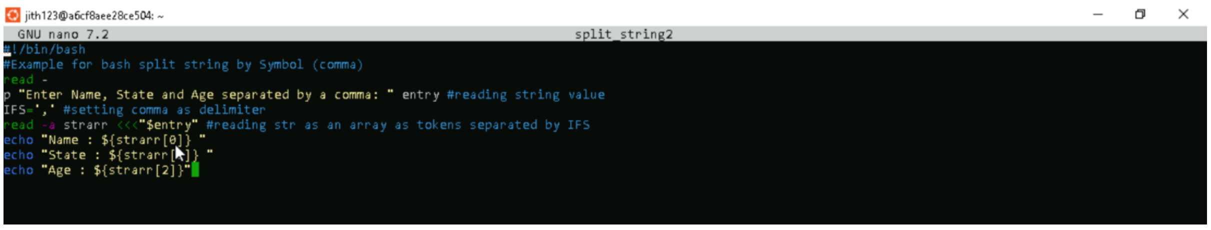


```
jith123@a6cf8aee28ce504:~$ nano split_string1
jith123@a6cf8aee28ce504:~$ ./split_string1
Enter any string separated by space: my name is jithendra
my name is jithendra
jith123@a6cf8aee28ce504:~$ nano split_string2
jith123@a6cf8aee28ce504:~$ 
```

Type here to search

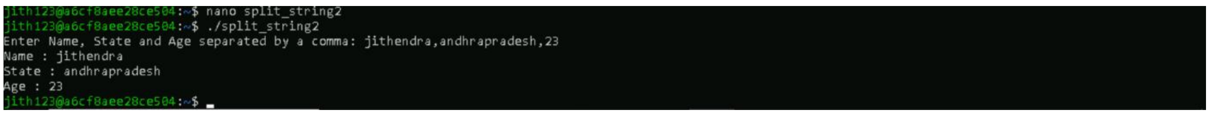
ENG US 17:25 28-01-2025

Split_string by symbol:



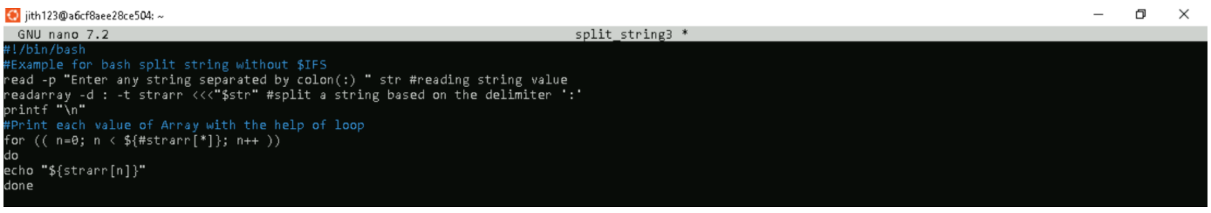
```
jith123@a6cf8aee28ce504: ~
  GNU nano 7.2                                           split_string2
#!/bin/bash
#Example for bash split string by Symbol (comma)
read -
p "Enter Name, State and Age separated by a comma: " entry #reading string value
IFS=',' #setting comma as delimiter
read -a strarr <<<"$entry" #reading str as an array as tokens separated by IFS
echo "Name : ${strarr[0]} "
echo "State : ${strarr[1]} "
echo "Age : ${strarr[2]}"
```

Output:



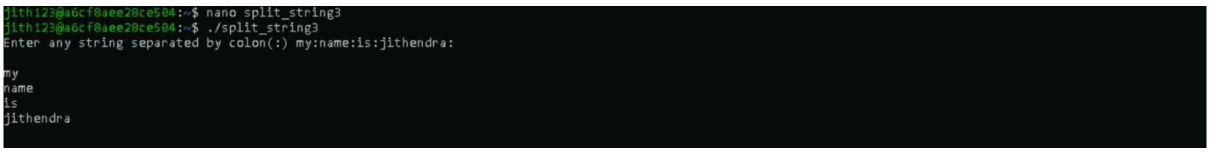
```
jith123@a6cf8aee28ce504:~$ nano split_string2
jith123@a6cf8aee28ce504:~$ ./split_string2
Enter Name, State and Age separated by a comma: jithendra,andhrapradesh,23
Name : jithendra
State : andhrapradesh
Age : 23
jith123@a6cf8aee28ce504:~$ 
```

Split string without IFS:



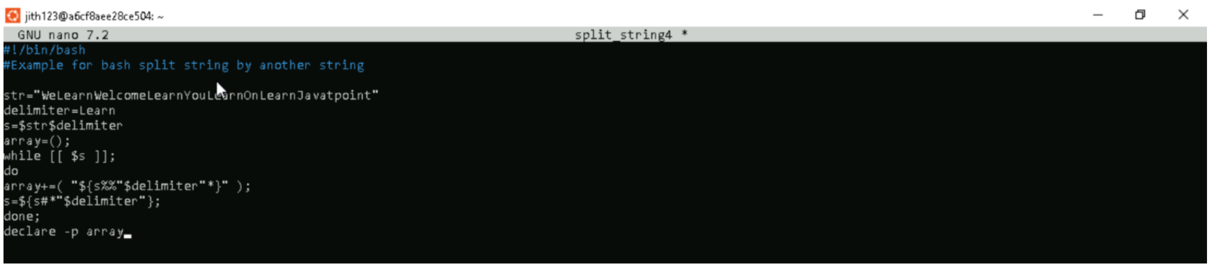
```
jith123@a6cf8aee28ce504: ~
  GNU nano 7.2                                           split_string3 *
#!/bin/bash
#Example for bash split string without $IFS
read -p "Enter any string separated by colon(:) " str #reading string value
readarray -d : -t strarr <<<"$str" #split a string based on the delimiter ':'
printf "\n"
#Print each value of Array with the help of loop
for (( n=0; n < ${#strarr[*]}; n++ ))
do
echo "${strarr[n]}"
done
```

Output:



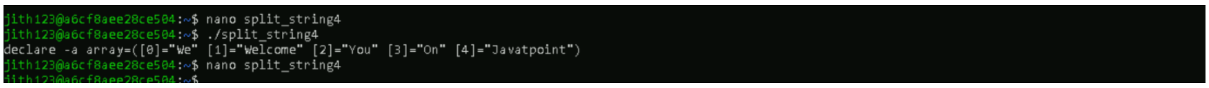
```
jith123@a6cf8aee28ce504:~$ nano split_string3
jith123@a6cf8aee28ce504:~$ ./split_string3
Enter any string separated by colon(:) my:name:is:jithendra:

my
name
is
jithendra
```

Split string by another string:



```
jith123@a6cf8aee28ce504: ~
  GNU nano 7.2                                                           split_string4 *
#!/bin/bash
#Example for bash split string by another string

str="WeLearnWelcomeLearnYouLearnOnLearnJavatpoint"
delimiter=Learn
s=$str$delimiter
array=();
while [[ $s ]];
do
array+=( "${s%%"$delimiter"*}" );
s=${s#*"$delimiter"};
done;
declare -p array
```

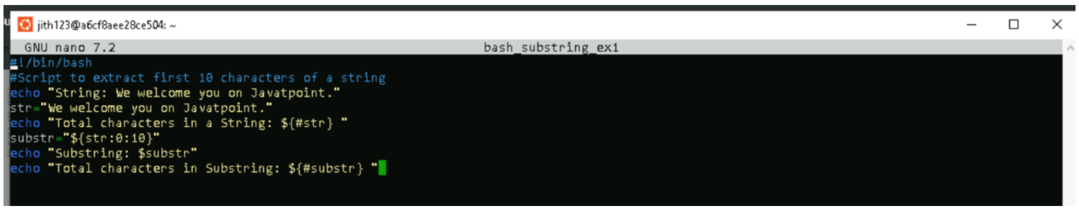
Output:



```
jith123@a6cf8aee28ce504:~$ nano split_string4
jith123@a6cf8aee28ce504:~$ ./split_string4
declare -a array=([0]="We" [1]="Welcome" [2]="You" [3]="On" [4]="Javatpoint")
jith123@a6cf8aee28ce504:~$ nano split_string4
jith123@a6cf8aee28ce504:~$
```

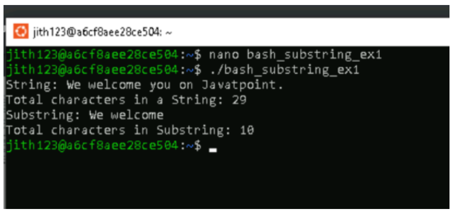
<u>Substring:</u>

Extract first 10 characters of a string:



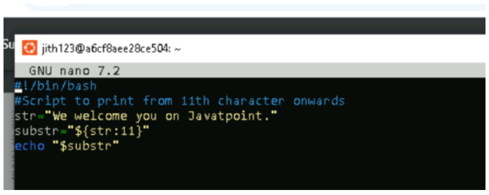
```
jith123@a6cf8aee28ce504: ~
  GNU nano 7.2                                             bash_substring_ex1
#!/bin/bash
#Script to extract first 10 characters of a string
echo "String: We welcome you on Javatpoint."
str="We welcome you on Javatpoint."
echo "Total characters in a String: ${#str} "
substr="${str:0:10}"
echo "Substring: $substr"
echo "Total characters in Substring: ${#substr} "
```

Output:



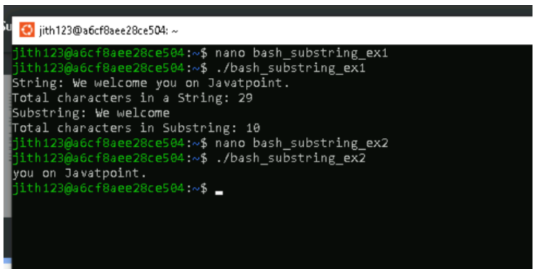
```
jith123@a6cf8aee28ce504: ~
jith123@a6cf8aee28ce504:~$ nano bash_substring_ex1
jith123@a6cf8aee28ce504:~$ ./bash_substring_ex1
String: We welcome you on Javatpoint.
Total characters in a String: 29
Substring: We welcome
Total characters in Substring: 10
jith123@a6cf8aee28ce504:~$
```

Printing the 11th character in a string:



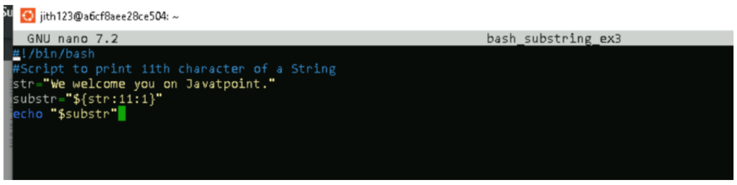
```
jith123@a6cf8aee28ce504: ~
  GNU nano 7.2
#!/bin/bash
#Script to print from 11th character onwards
str="We welcome you on Javatpoint."
substr="${str:11}"
echo "$substr"
```

Output:



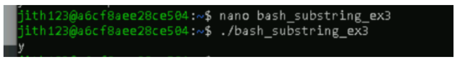
```
jith123@a6cf8aee28ce504:~$ nano bash_substring_ex1
jith123@a6cf8aee28ce504:~$ ./bash_substring_ex1
String: We welcome you on Javatpoint.
Total characters in a String: 29
Substring: We welcome
Total characters in Substring: 10
jith123@a6cf8aee28ce504:~$ nano bash_substring_ex2
jith123@a6cf8aee28ce504:~$ ./bash_substring_ex2
you on Javatpoint.
jith123@a6cf8aee28ce504:~$ _
```

Print the 11th character in a string:



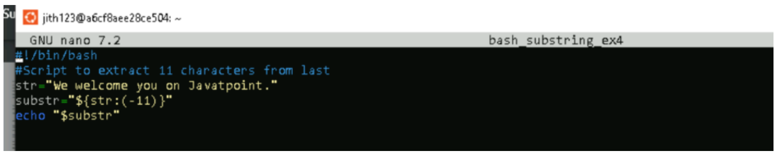
```
  GNU nano 7.2                                              bash_substring_ex3
#!/bin/bash
#Script to print 11th character of a String
str="We welcome you on Javatpoint."
substr="${str:11:1}"
echo "$substr"
```

Output:



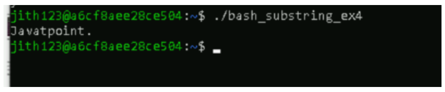
```
jith123@a6cf8aee28ce504:~$ nano bash_substring_ex3
jith123@a6cf8aee28ce504:~$ ./bash_substring_ex3
y
```

To exctract 11th character from Last:



```
  GNU nano 7.2                                              bash_substring_ex4
#!/bin/bash
#Script to extract 11 characters from last
str="We welcome you on Javatpoint."
substr="${str:(-11)}"
echo "$substr"
```

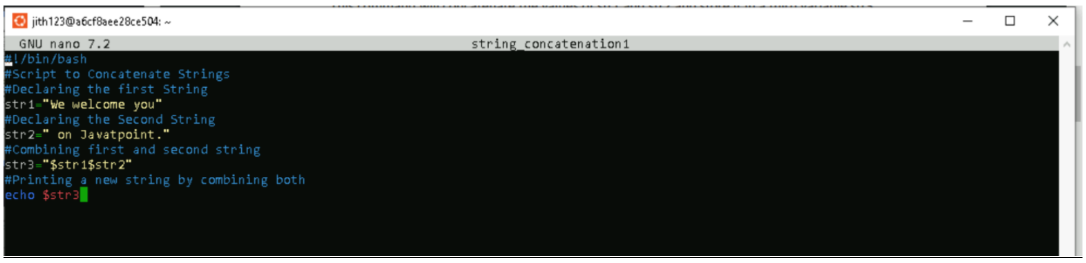
Output:



```
jith123@a6cf8aee28ce504:~$ ./bash_substring_ex4
Javatpoint.
jith123@a6cf8aee28ce504:~$ _
```

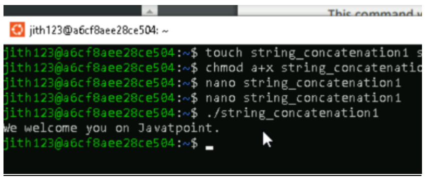
<u>String concatenation:</u>

Example 1: Write Variables Side by Side:



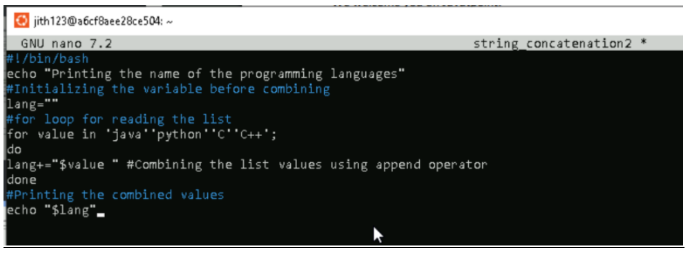
```
  GNU nano 7.2                                           string_concatenation1
#!/bin/bash
#Script to Concatenate Strings
#Declaring the first String
str1="We welcome you"
#Declaring the Second String
str2=" on Javatpoint."
#Combining first and second string
str3="$str1$str2"
#Printing a new string by combining both
echo $str3
```

Output:



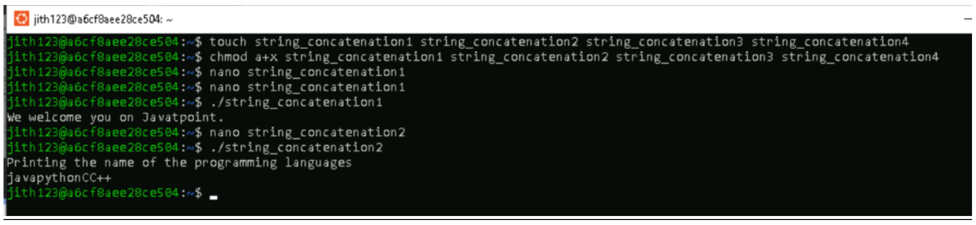
```
jith123@a6cf8aee28ce504:~$ touch string_concatenation1 s
jith123@a6cf8aee28ce504:~$ chmod a+x string_concatenatio
jith123@a6cf8aee28ce504:~$ nano string_concatenation1
jith123@a6cf8aee28ce504:~$ nano string_concatenation1
jith123@a6cf8aee28ce504:~$ ./string_concatenation1
We welcome you on Javatpoint.
jith123@a6cf8aee28ce504:~$ _
```

Example 2: Using Append Operator with Loop:



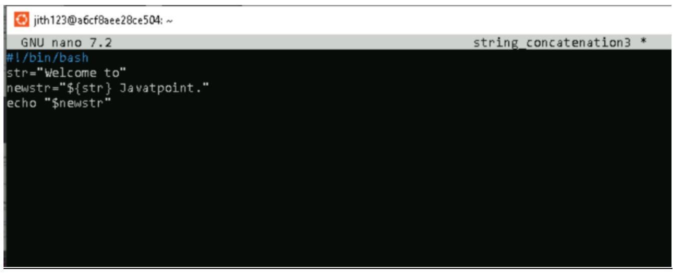
```
GNU nano 7.2                                              string_concatenation2 *
#!/bin/bash
echo "Printing the name of the programming languages"
#Initializing the variable before combining
lang=""
#for loop for reading the list
for value in 'java''python''C''C++';
do
lang+="$value " #Combining the list values using append operator
done
#Printing the combined values
echo "$lang"
```

Output:



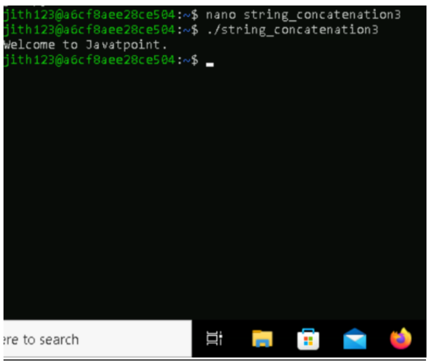
```
jith123@a6cf8aee28ce504:~$ touch string_concatenation1 string_concatenation2 string_concatenation3 string_concatenation4
jith123@a6cf8aee28ce504:~$ chmod a+x string_concatenation1 string_concatenation2 string_concatenation3 string_concatenation4
jith123@a6cf8aee28ce504:~$ nano string_concatenation1
jith123@a6cf8aee28ce504:~$ nano string_concatenation1
jith123@a6cf8aee28ce504:~$ ./string_concatenation1
We welcome you on Javatpoint.
jith123@a6cf8aee28ce504:~$ nano string_concatenation2
jith123@a6cf8aee28ce504:~$ ./string_concatenation2
Printing the name of the programming languages
javapythonCC++
jith123@a6cf8aee28ce504:~$ _
```

Example 3: Using Literal Strings:



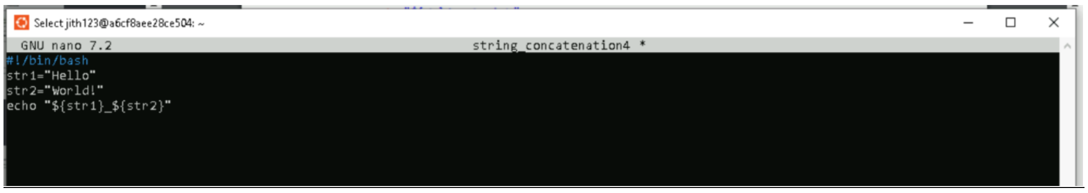
```
GNU nano 7.2                                              string_concatenation3 *
#!/bin/bash
str="Welcome to"
newstr="${str} Javatpoint."
echo "$newstr"
```

Output:



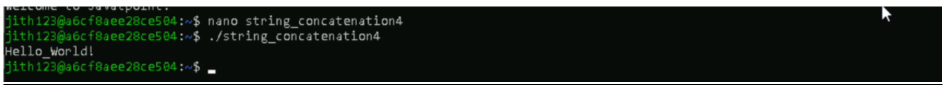
```
jith123@a6cf8aee28ce504:~$ nano string_concatenation3
jith123@a6cf8aee28ce504:~$ ./string_concatenation3
Welcome to Javatpoint.
jith123@a6cf8aee28ce504:~$ _
```

Example 4: Using Underscore:



```
Select jith123@a6cf8aee28ce504: ~
 GNU nano 7.2                         string_concatenation4 *
#!/bin/bash
str1="Hello"
str2="World!"
echo "${str1}_${str2}"
```

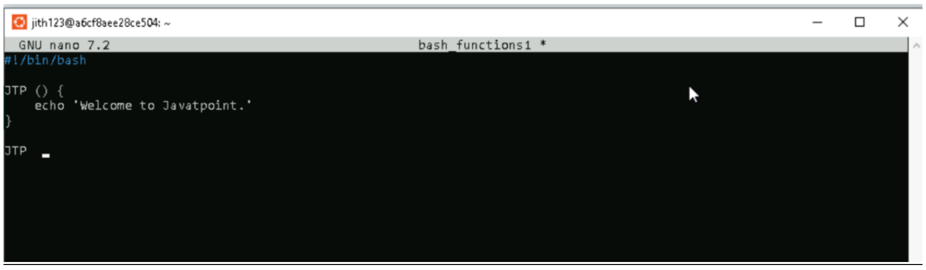
Ouput:



```
jith123@a6cf8aee28ce504:~$ nano string_concatenation4
jith123@a6cf8aee28ce504:~$ ./string_concatenation4
Hello_World!
jith123@a6cf8aee28ce504:~$
```

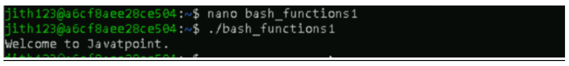
<u>FUNCTIONS:</u>

HOW TO USE FUNCTIONS:



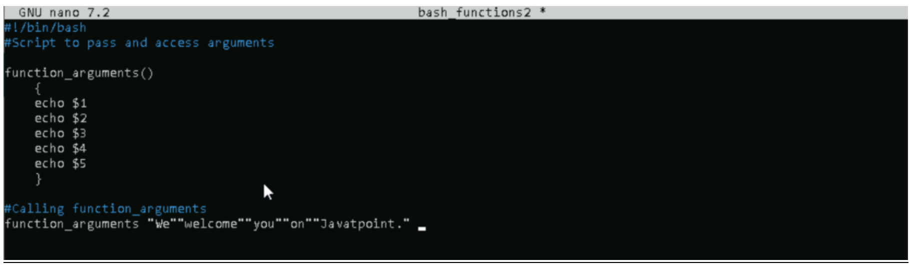
```
jith123@a6cf8aee28ce504: ~
 GNU nano 7.2                         bash_functions1 *
#!/bin/bash

JTP () {
    echo 'Welcome to Javatpoint.'
}

JTP
```

OUTPUT:



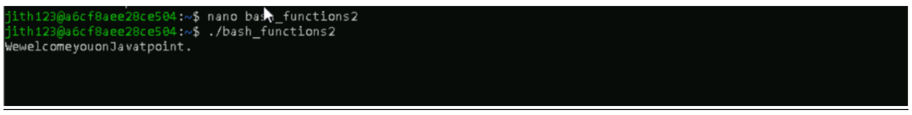
```
jith123@a6cf8aee28ce504:~$ nano bash_functions1
jith123@a6cf8aee28ce504:~$ ./bash_functions1
Welcome to Javatpoint.
```

HOW TO PASS ARGUMENTS:



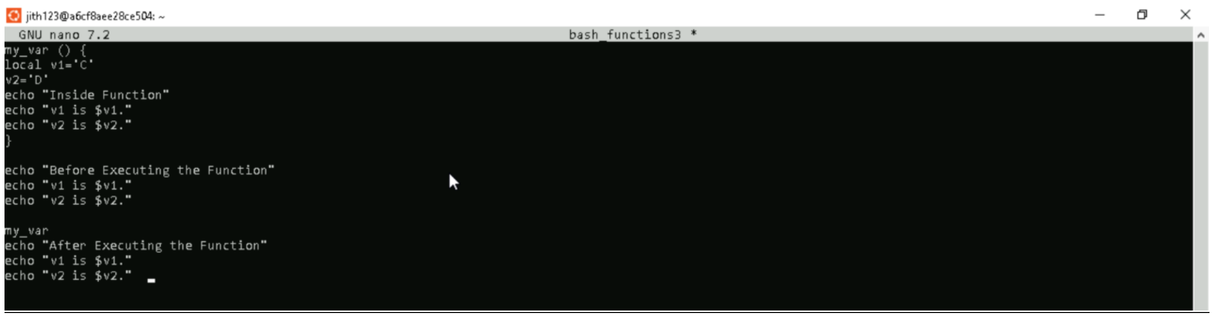
```
 GNU nano 7.2                         bash_functions2 *
#!/bin/bash
#Script to pass and access arguments

function_arguments()
    {
    echo $1
    echo $2
    echo $3
    echo $4
    echo $5
    }

#Calling function_arguments
function_arguments "We""welcome""you""on""Javatpoint."
```

OUTPUT:



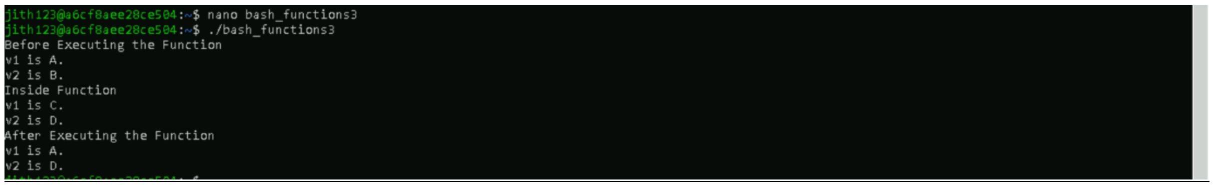
```
jith123@a6cf8aee28ce504:~$ nano bash_functions2
jith123@a6cf8aee28ce504:~$ ./bash_functions2
WewelcomeyouonJavatpoint.
```

VARIABLE SCOPE:



```
jith123@a6cf8aee28ce504: ~
  GNU nano 7.2                                              bash_functions3 *
my_var () {
local v1='C'
v2='D'
echo "Inside Function"
echo "v1 is $v1."
echo "v2 is $v2."
}
echo "Before Executing the Function"
echo "v1 is $v1."
echo "v2 is $v2."

my_var
echo "After Executing the Function"
echo "v1 is $v1."
echo "v2 is $v2."
```

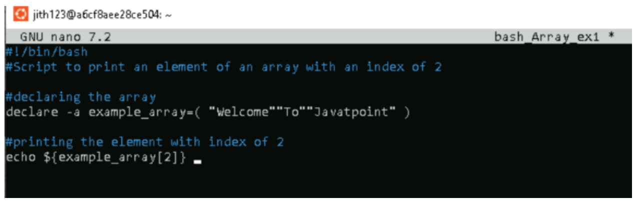
OUTPUT:



```
jith123@a6cf8aee28ce504:~$ nano bash_functions3
jith123@a6cf8aee28ce504:~$ ./bash_functions3
Before Executing the Function
v1 is A.
v2 is B.
Inside Function
v1 is C.
v2 is D.
After Executing the Function
v1 is A.
v2 is D.
```

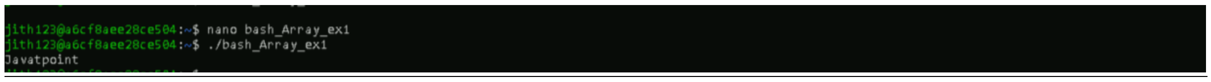
Array:

Print an array with an index of 2:



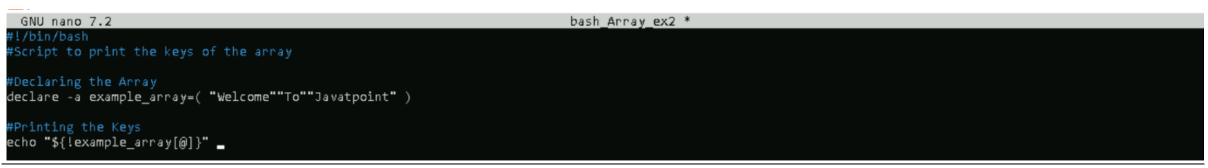
```
jith123@a6cf8aee28ce504: ~
  GNU nano 7.2                                   bash_Array_ex1 *
#!/bin/bash
#Script to print an element of an array with an index of 2

#declaring the array
declare -a example_array=( "Welcome""To""Javatpoint" )

#printing the element with index of 2
echo ${example_array[2]}
```

Output:



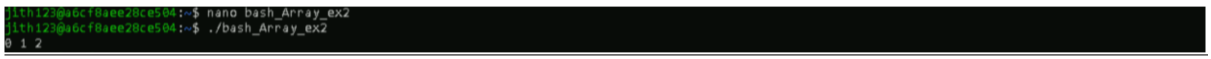
```
jith123@a6cf8aee28ce504:~$ nano bash_Array_ex1
jith123@a6cf8aee28ce504:~$ ./bash_Array_ex1
Javatpoint
```

Print the keys of the array:



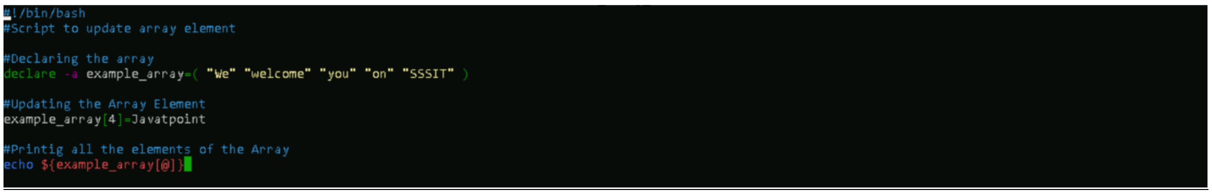
```
  GNU nano 7.2                                              bash_Array_ex2 *
#!/bin/bash
#Script to print the keys of the array

#Declaring the Array
declare -a example_array=( "Welcome""To""Javatpoint" )

#Printing the Keys
echo "${!example_array[@]}"
```

Output:



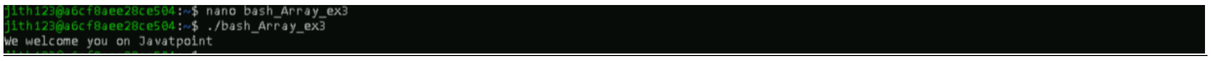
```
jith123@a6cf8aee28ce504:~$ nano bash_Array_ex2
jith123@a6cf8aee28ce504:~$ ./bash_Array_ex2
0 1 2
```

Update the array element:



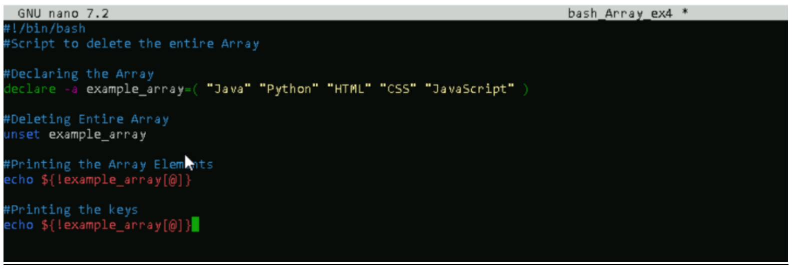
```
#!/bin/bash
#Script to update array element

#Declaring the array
declare -a example_array=( "We" "welcome" "you" "on" "SSSIT" )

#Updating the Array Element
example_array[4]=Javatpoint

#Printig all the elements of the Array
echo ${example_array[@]}
```

Output:



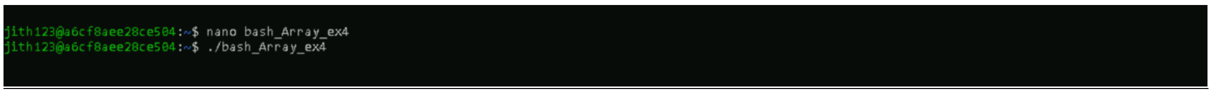
```
jith123@a6cf8aee28ce504:~$ nano bash_Array_ex3
jith123@a6cf8aee28ce504:~$ ./bash_Array_ex3
We welcome you on Javatpoint
```

Delete the entire array:



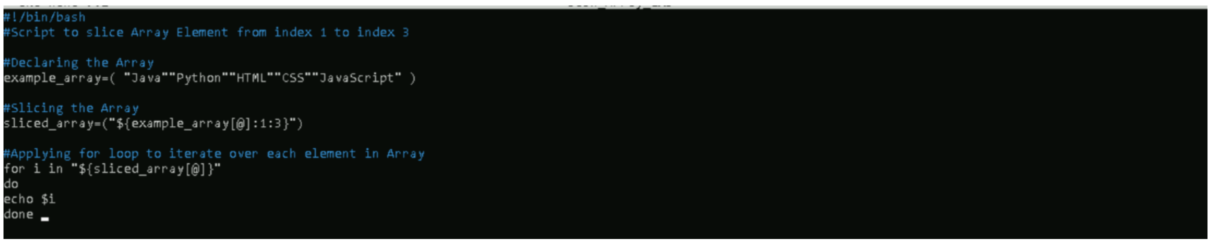
```
  GNU nano 7.2                                          bash_Array_ex4 *
#!/bin/bash
#Script to delete the entire Array

#Declaring the Array
declare -a example_array=( "Java" "Python" "HTML" "CSS" "JavaScript" )

#Deleting Entire Array
unset example_array

#Printing the Array Elements
echo ${!example_array[@]}

#Printing the keys
echo ${!example_array[@]}
```

Output:



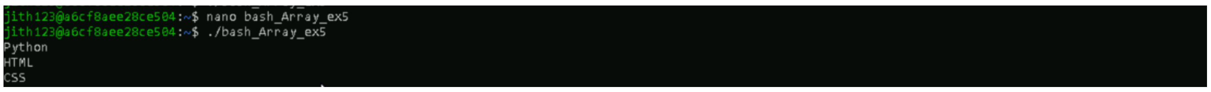
```
jith123@a6cf8aee28ce504:~$ nano bash_Array_ex4
jith123@a6cf8aee28ce504:~$ ./bash_Array_ex4
```

Slicing array elements:



```
#!/bin/bash
#Script to slice Array Element from index 1 to index 3

#Declaring the Array
example_array=( "Java""Python""HTML""CSS""JavaScript" )

#Slicing the Array
sliced_array=("${example_array[@]:1:3}")

#Applying for loop to iterate over each element in Array
for i in "${sliced_array[@]}"
do
echo $i
done
```

Output:



```
jith123@a6cf8aee28ce504:~$ nano bash_Array_ex5
jith123@a6cf8aee28ce504:~$ ./bash_Array_ex5
Python
HTML
CSS
```