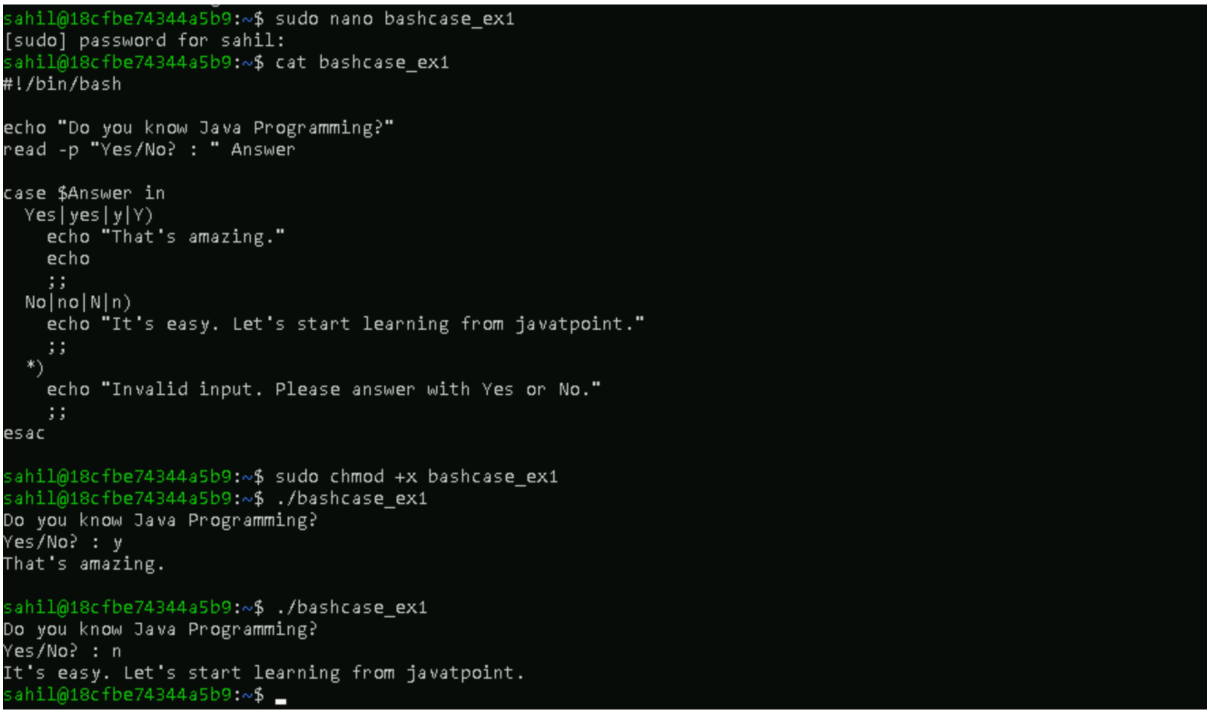
BASH PROJECT

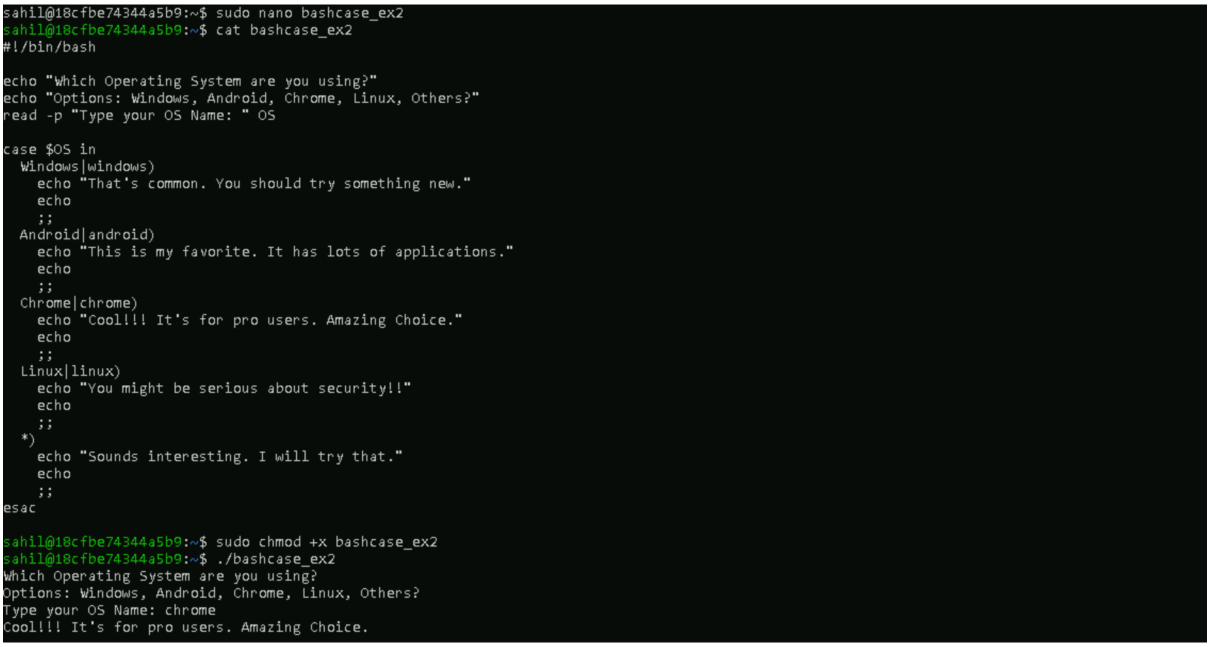
BASH CASE:

BASH CASE EXAMPLE 1:



```
sahil@18cfbe74344a5b9:~$ sudo nano bashcase_ex1
[sudo] password for sahil:
sahil@18cfbe74344a5b9:~$ cat bashcase_ex1
#!/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."
    ;;
  *)
    echo "Invalid input. Please answer with Yes or No."
    ;;
esac

sahil@18cfbe74344a5b9:~$ sudo chmod +x bashcase_ex1
sahil@18cfbe74344a5b9:~$ ./bashcase_ex1
Do you know Java Programming?
Yes/No? : y
That's amazing.

sahil@18cfbe74344a5b9:~$ ./bashcase_ex1
Do you know Java Programming?
Yes/No? : n
It's easy. Let's start learning from javatpoint.
sahil@18cfbe74344a5b9:~$ _
```

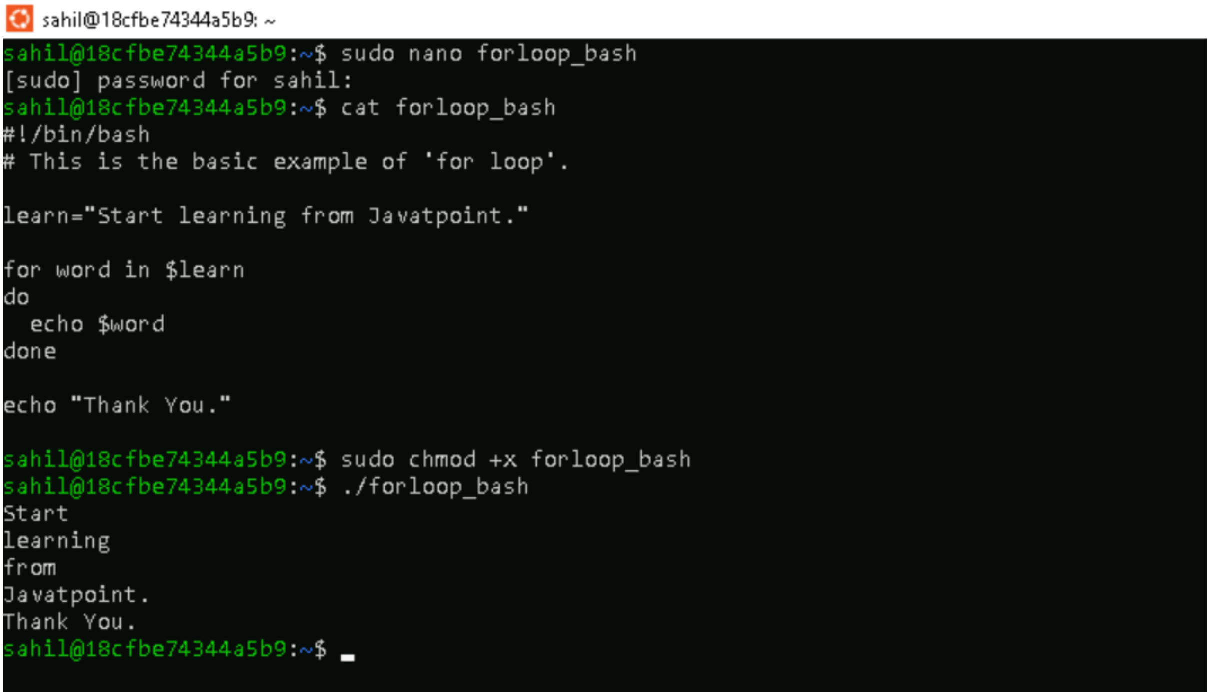
BASH CASE EXAMPLE 2:



```
sahil@18cfbe74344a5b9:~$ sudo nano bashcase_ex2
sahil@18cfbe74344a5b9:~$ cat bashcase_ex2
#!/bin/bash

echo "Which Operating System are you using?"
echo "Options: 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

sahil@18cfbe74344a5b9:~$ sudo chmod +x bashcase_ex2
sahil@18cfbe74344a5b9:~$ ./bashcase_ex2
Which Operating System are you using?
Options: Windows, Android, Chrome, Linux, Others?
Type your OS Name: chrome
Cool!!! It's for pro users. Amazing Choice.
```

BASH FOR LOOP:

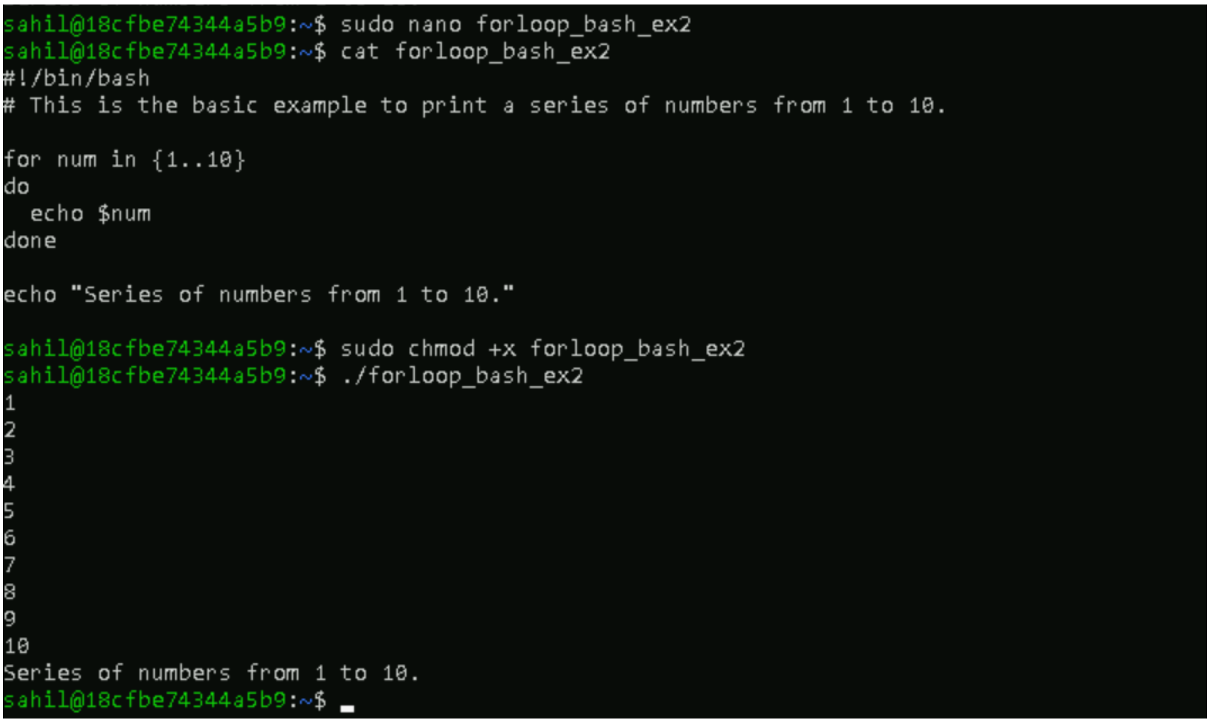
BASH FOR LOOP EXAMPLE 1:



sahil@18cfbe74344a5b9: ~

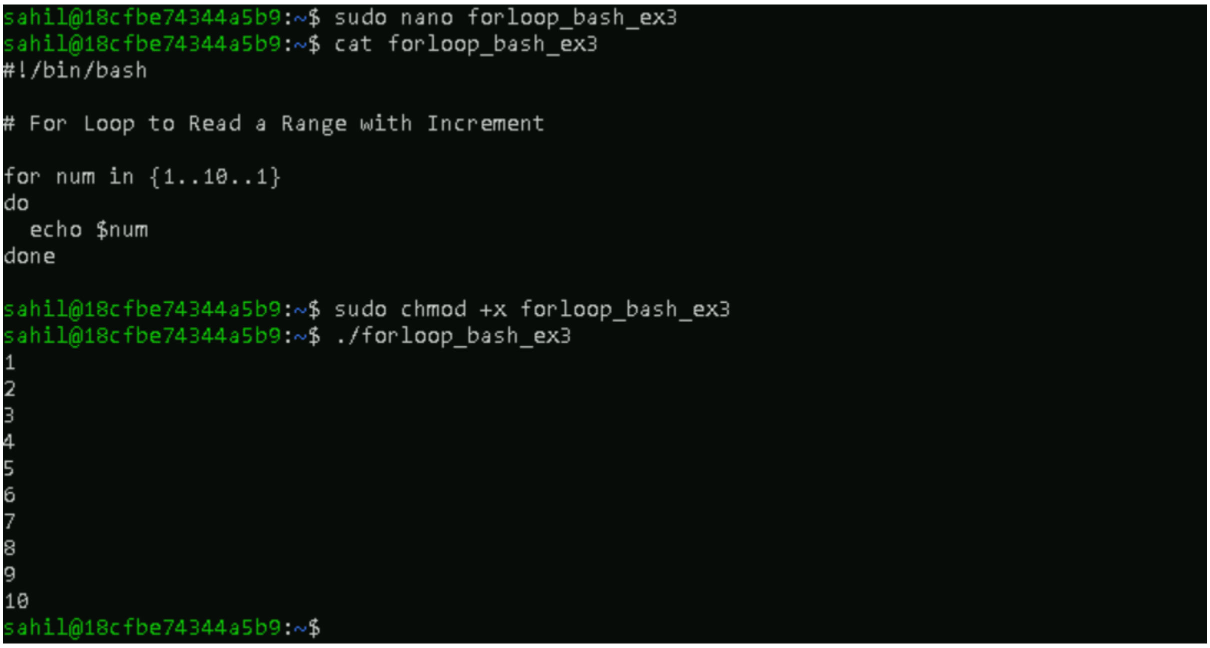
```
sahil@18cfbe74344a5b9:~$ sudo nano forloop_bash
[sudo] password for sahil:
sahil@18cfbe74344a5b9:~$ cat forloop_bash
#!/bin/bash
# This is the basic example of 'for loop'.

learn="Start learning from Javatpoint."

for word in $learn
do
  echo $word
done

echo "Thank You."

sahil@18cfbe74344a5b9:~$ sudo chmod +x forloop_bash
sahil@18cfbe74344a5b9:~$ ./forloop_bash
Start
learning
from
Javatpoint.
Thank You.
sahil@18cfbe74344a5b9:~$ _
```

BASH FOR LOOP EXAMPLE 2:



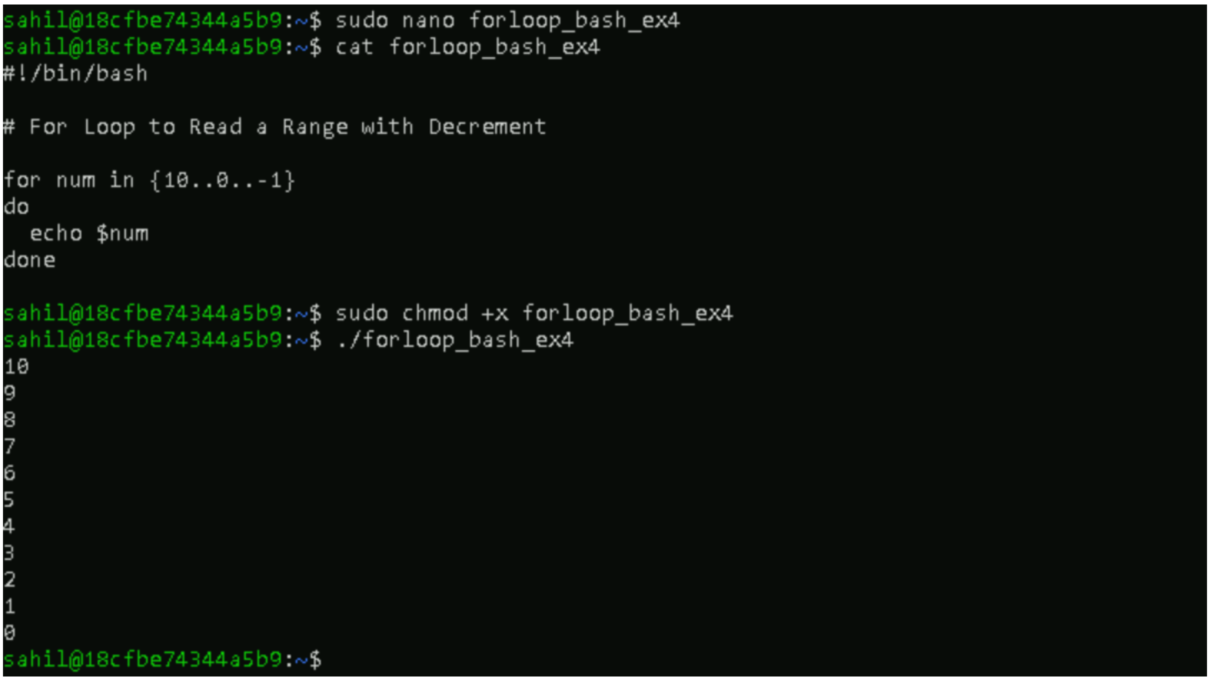
```
sahil@18cfbe74344a5b9:~$ sudo nano forloop_bash_ex2
sahil@18cfbe74344a5b9:~$ cat forloop_bash_ex2
#!/bin/bash
# This is the basic example to print a series of numbers from 1 to 10.

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

echo "Series of numbers from 1 to 10."

sahil@18cfbe74344a5b9:~$ sudo chmod +x forloop_bash_ex2
sahil@18cfbe74344a5b9:~$ ./forloop_bash_ex2
1
2
3
4
5
6
7
8
9
10
Series of numbers from 1 to 10.
sahil@18cfbe74344a5b9:~$ _
```

BASH FOR LOOP EXAMPLE 3:



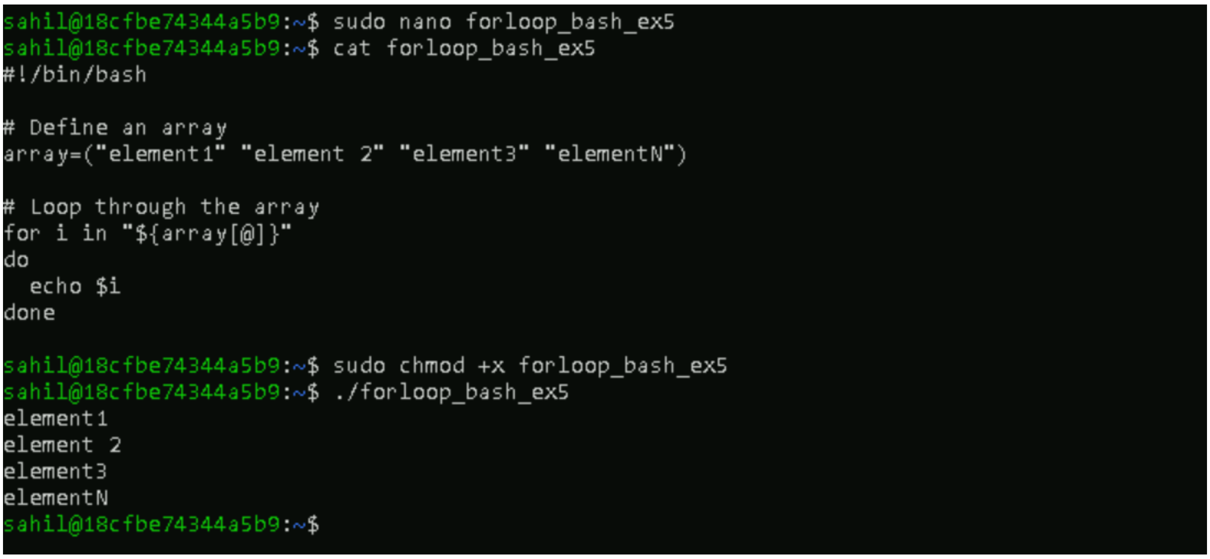
```
sahil@18cfbe74344a5b9:~$ sudo nano forloop_bash_ex3
sahil@18cfbe74344a5b9:~$ cat forloop_bash_ex3
#!/bin/bash

# For Loop to Read a Range with Increment

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

sahil@18cfbe74344a5b9:~$ sudo chmod +x forloop_bash_ex3
sahil@18cfbe74344a5b9:~$ ./forloop_bash_ex3
1
2
3
4
5
6
7
8
9
10
sahil@18cfbe74344a5b9:~$
```

BASH FOR LOOP EXAMPLE 4:



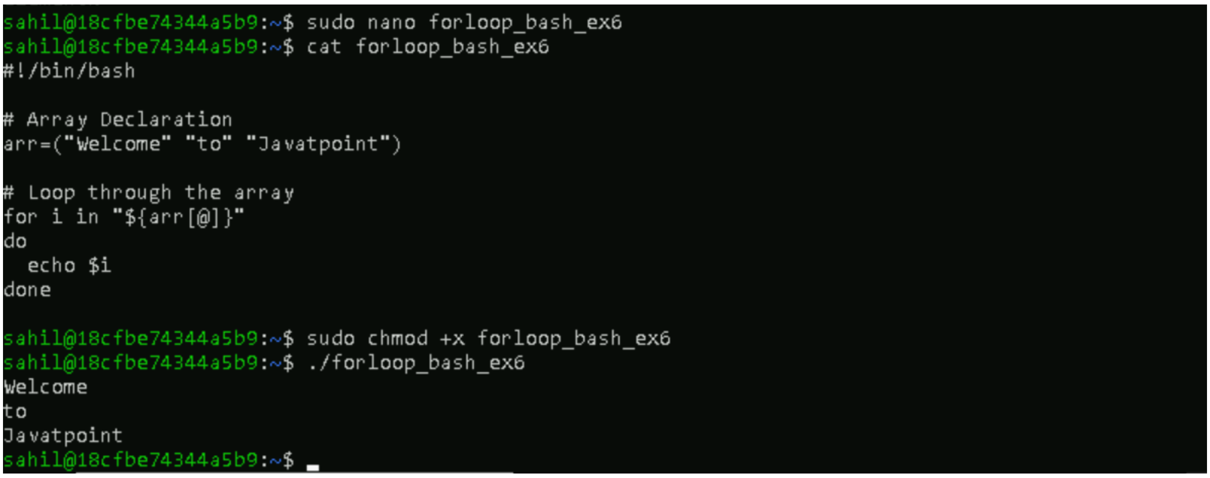
```
sahil@18cfbe74344a5b9:~$ sudo nano forloop_bash_ex4
sahil@18cfbe74344a5b9:~$ cat forloop_bash_ex4
#!/bin/bash

# For Loop to Read a Range with Decrement

for num in {10..0..-1}
do
  echo $num
done

sahil@18cfbe74344a5b9:~$ sudo chmod +x forloop_bash_ex4
sahil@18cfbe74344a5b9:~$ ./forloop_bash_ex4
10
9
8
7
6
5
4
3
2
1
0
sahil@18cfbe74344a5b9:~$
```

BASH FOR LOOP EXAMPLE 5:



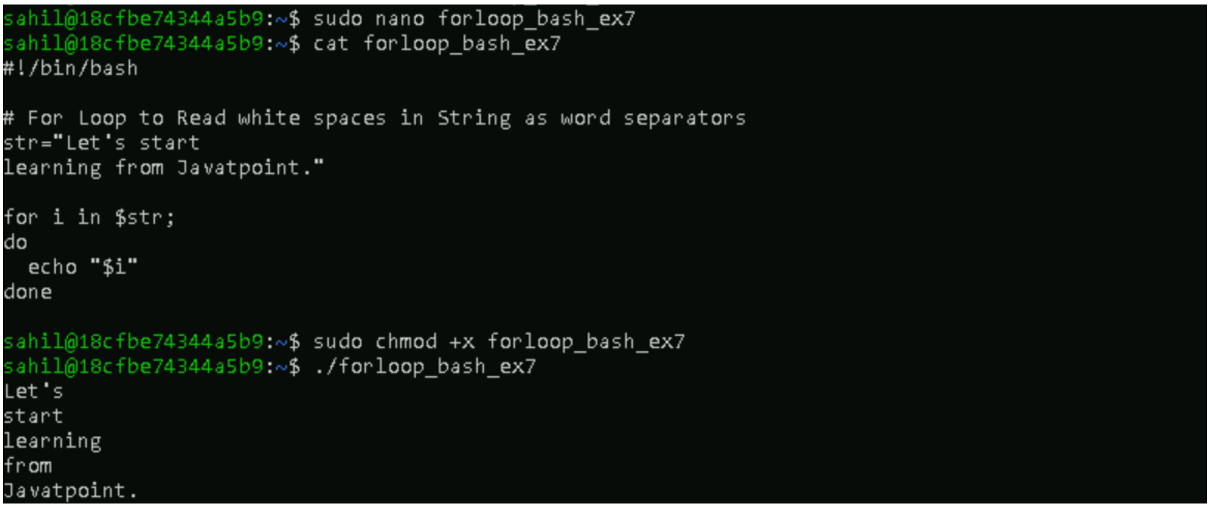
```
sahil@18cfbe74344a5b9:~$ sudo nano forloop_bash_ex5
sahil@18cfbe74344a5b9:~$ cat forloop_bash_ex5
#!/bin/bash

# Define an array
array=("element1" "element 2" "element3" "elementN")

# Loop through the array
for i in "${array[@]}"
do
  echo $i
done

sahil@18cfbe74344a5b9:~$ sudo chmod +x forloop_bash_ex5
sahil@18cfbe74344a5b9:~$ ./forloop_bash_ex5
element1
element 2
element3
elementN
sahil@18cfbe74344a5b9:~$
```

BASH FOR LOOP EXAMPLE 6:



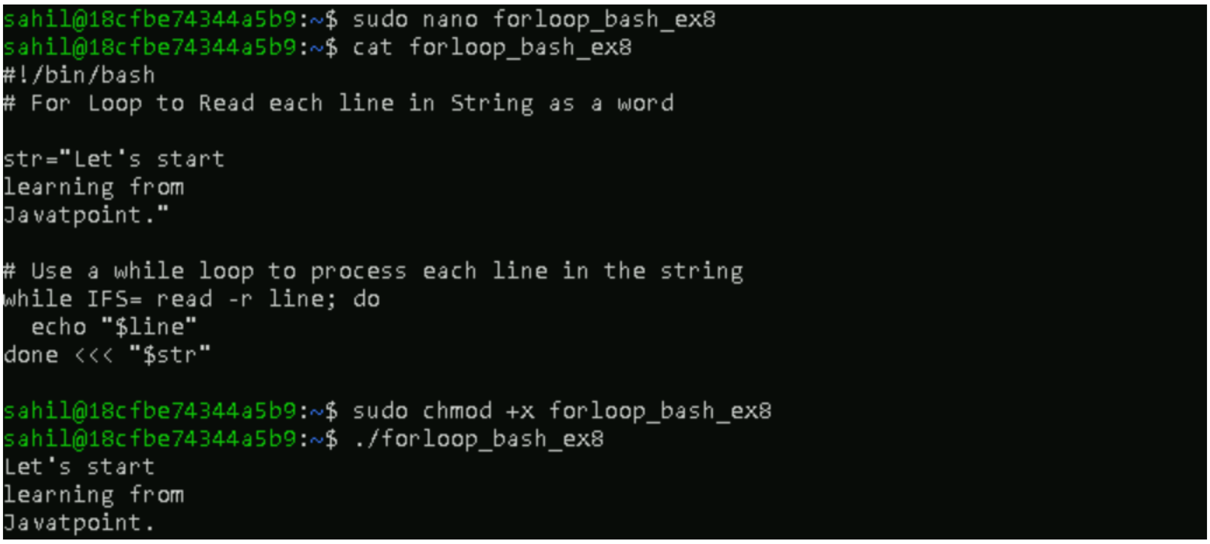
```
sahil@18cfbe74344a5b9:~$ sudo nano forloop_bash_ex6
sahil@18cfbe74344a5b9:~$ cat forloop_bash_ex6
#!/bin/bash

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

# Loop through the array
for i in "${arr[@]}"
do
  echo $i
done

sahil@18cfbe74344a5b9:~$ sudo chmod +x forloop_bash_ex6
sahil@18cfbe74344a5b9:~$ ./forloop_bash_ex6
Welcome
to
Javatpoint
sahil@18cfbe74344a5b9:~$ _
```

BASH FOR LOOP EXAMPLE 7:



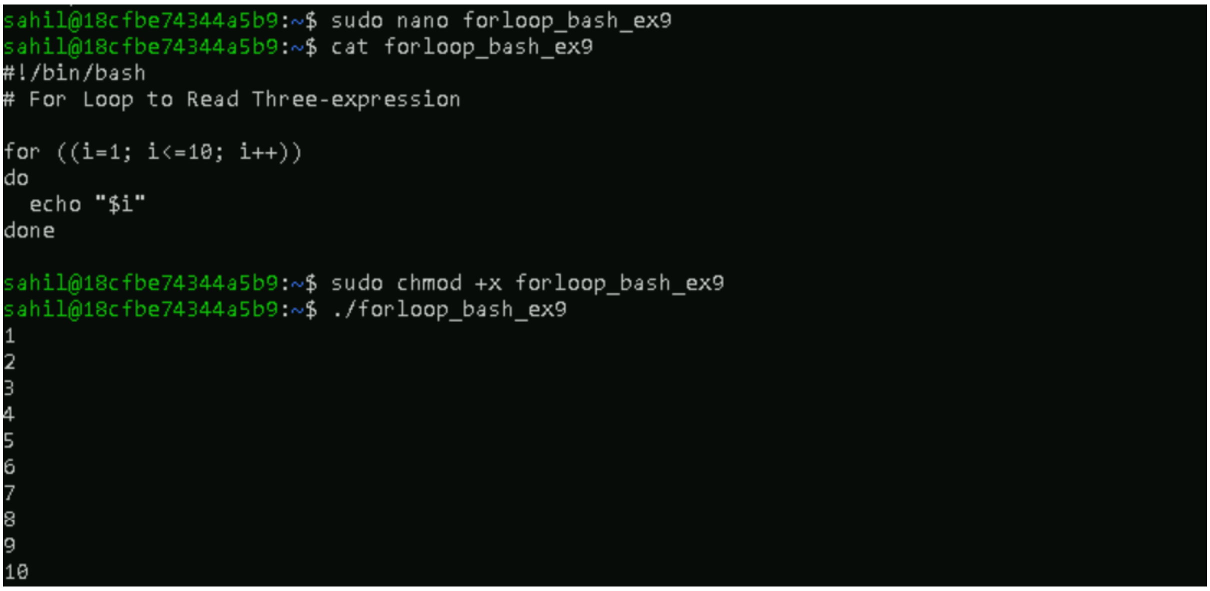
```
sahil@18cfbe74344a5b9:~$ sudo nano forloop_bash_ex7
sahil@18cfbe74344a5b9:~$ cat forloop_bash_ex7
#!/bin/bash

# For Loop to Read white spaces in String as word separators
str="Let's start
learning from Javatpoint."

for i in $str;
do
  echo "$i"
done

sahil@18cfbe74344a5b9:~$ sudo chmod +x forloop_bash_ex7
sahil@18cfbe74344a5b9:~$ ./forloop_bash_ex7
Let's
start
learning
from
Javatpoint.
```

BASH FOR LOOP EXAMPLE 8:



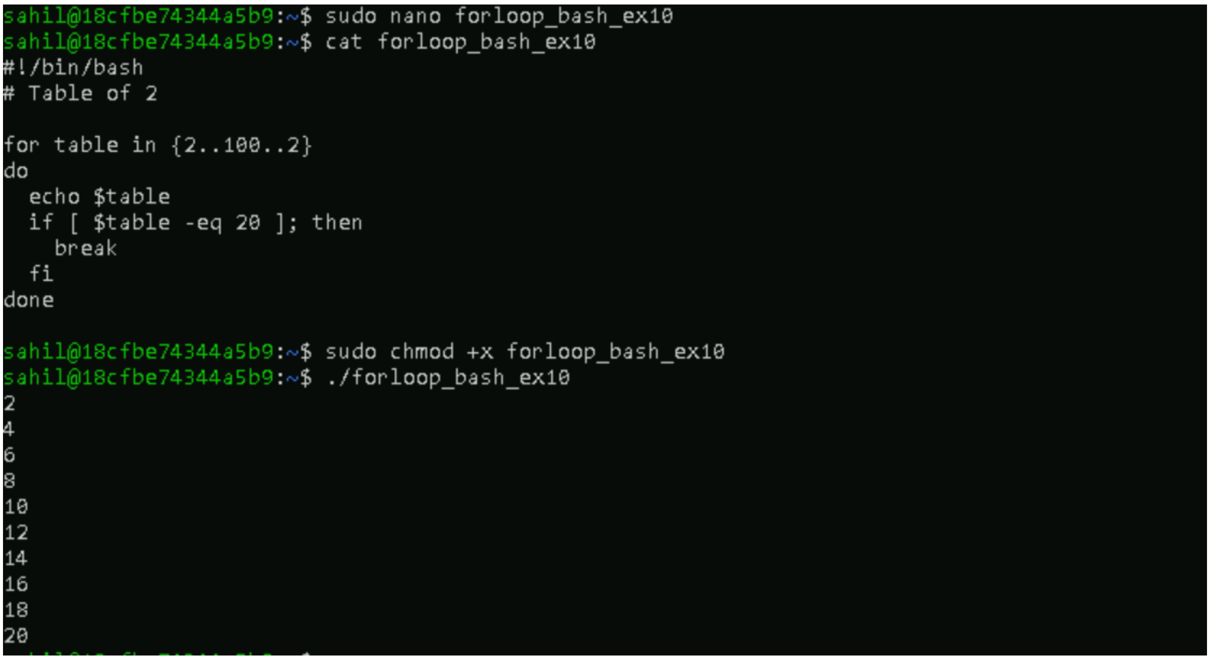
```
sahil@18cfbe74344a5b9:~$ sudo nano forloop_bash_ex8
sahil@18cfbe74344a5b9:~$ cat forloop_bash_ex8
#!/bin/bash
# For Loop to Read each line in String as a word

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

# Use a while loop to process each line in the string
while IFS= read -r line; do
  echo "$line"
done <<< "$str"

sahil@18cfbe74344a5b9:~$ sudo chmod +x forloop_bash_ex8
sahil@18cfbe74344a5b9:~$ ./forloop_bash_ex8
Let's start
learning from
Javatpoint.
```

BASH FOR LOOP EXAMPLE 9:



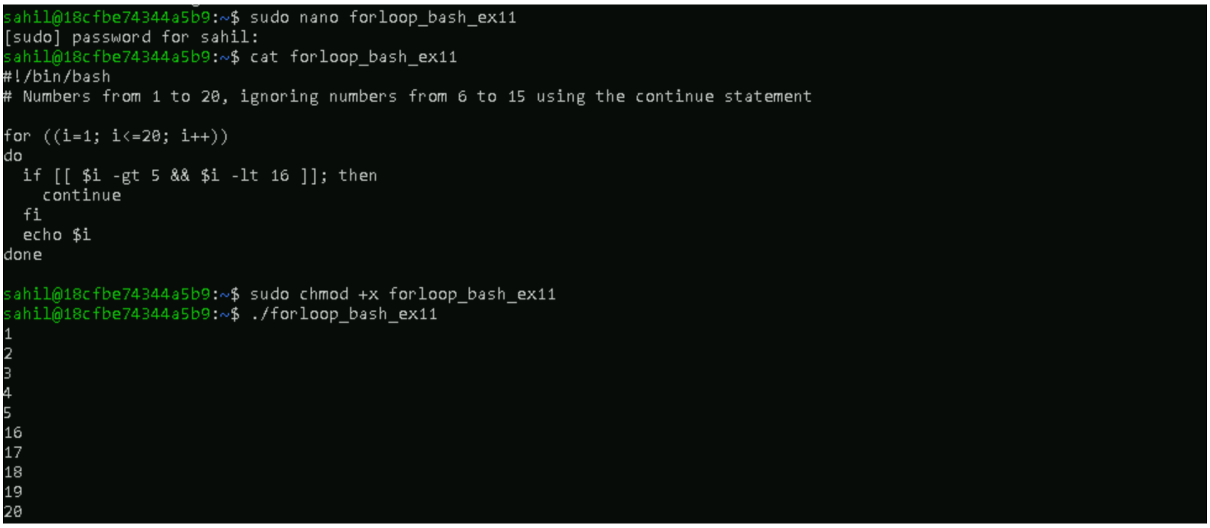
```
sahil@18cfbe74344a5b9:~$ sudo nano forloop_bash_ex9
sahil@18cfbe74344a5b9:~$ cat forloop_bash_ex9
#!/bin/bash
# For Loop to Read Three-expression

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

sahil@18cfbe74344a5b9:~$ sudo chmod +x forloop_bash_ex9
sahil@18cfbe74344a5b9:~$ ./forloop_bash_ex9
1
2
3
4
5
6
7
8
9
10
```

BASH FOR LOOP EXAMPLE 10:



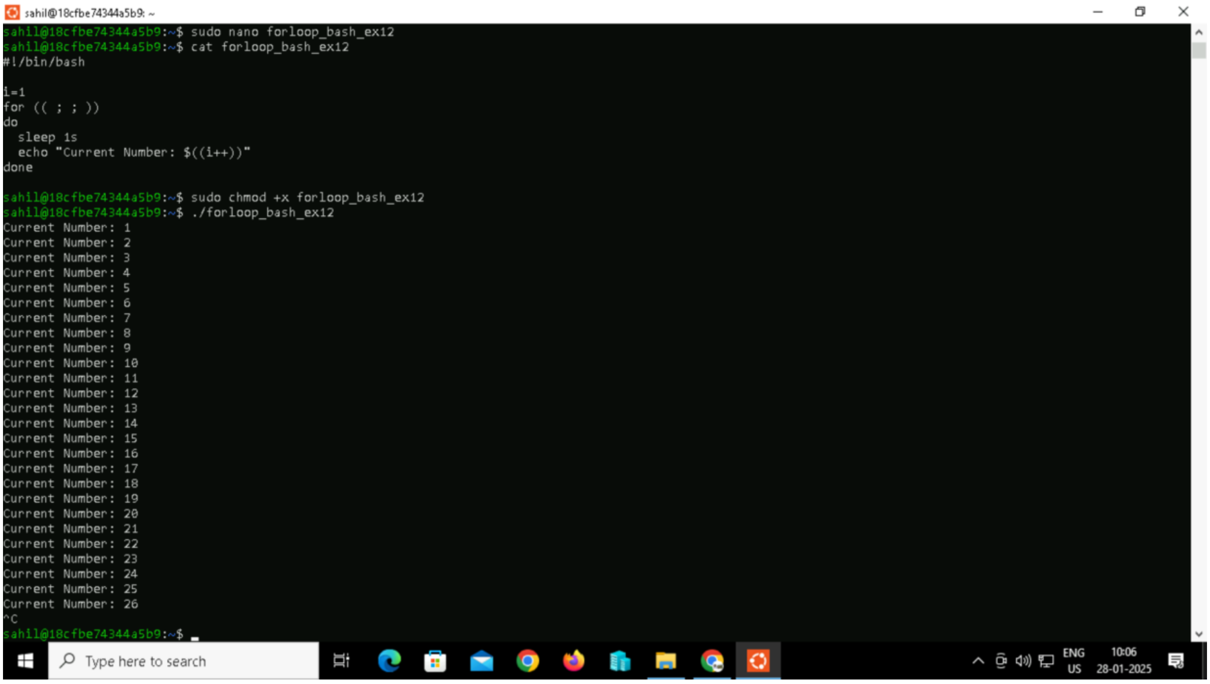
```
sahil@18cfbe74344a5b9:~$ sudo nano forloop_bash_ex10
sahil@18cfbe74344a5b9:~$ cat forloop_bash_ex10
#!/bin/bash
# Table of 2

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

sahil@18cfbe74344a5b9:~$ sudo chmod +x forloop_bash_ex10
sahil@18cfbe74344a5b9:~$ ./forloop_bash_ex10
2
4
6
8
10
12
14
16
18
20
```

BASH FOR LOOP EXAMPLE 11:



```
sahil@18cfbe74344a5b9:~$ sudo nano forloop_bash_ex11
[sudo] password for sahil:
sahil@18cfbe74344a5b9:~$ cat forloop_bash_ex11
#!/bin/bash
# Numbers from 1 to 20, ignoring numbers from 6 to 15 using the continue statement

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

sahil@18cfbe74344a5b9:~$ sudo chmod +x forloop_bash_ex11
sahil@18cfbe74344a5b9:~$ ./forloop_bash_ex11
1
2
3
4
5
16
17
18
19
20
```

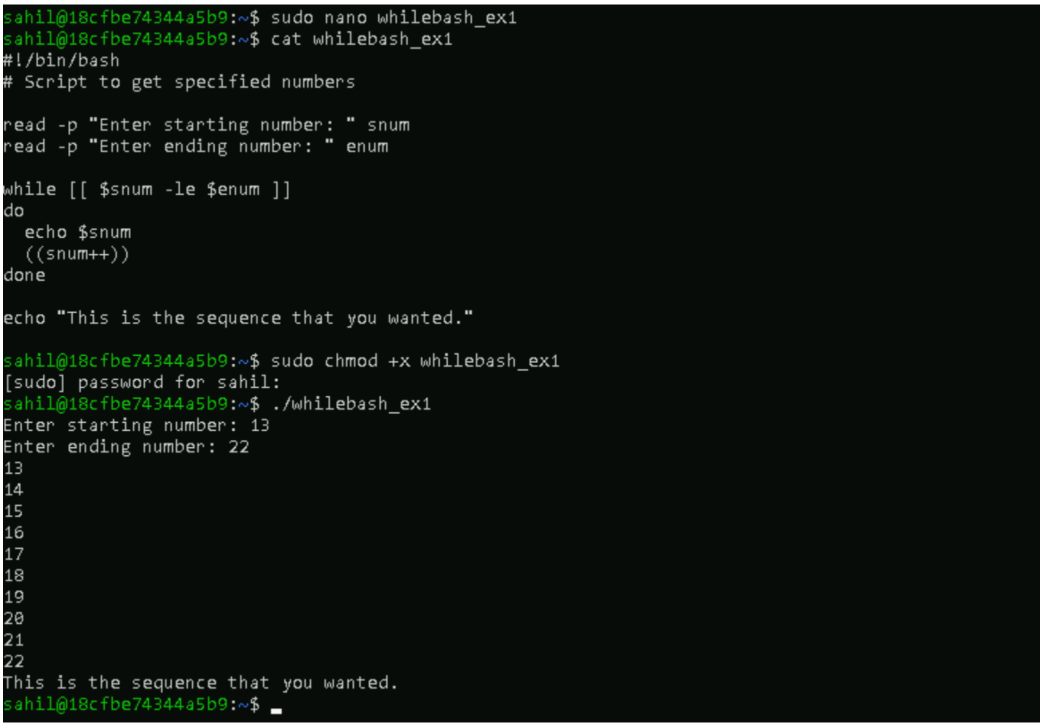
BASH FOR LOOP EXAMPLE 12:



```
sahil@18cfbe74344a5b9:~$ sudo nano forloop_bash_ex12
sahil@18cfbe74344a5b9:~$ cat forloop_bash_ex12
#!/bin/bash

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

sahil@18cfbe74344a5b9:~$ sudo chmod +x forloop_bash_ex12
sahil@18cfbe74344a5b9:~$ ./forloop_bash_ex12
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
Current Number: 17
Current Number: 18
Current Number: 19
Current Number: 20
Current Number: 21
Current Number: 22
Current Number: 23
Current Number: 24
Current Number: 25
Current Number: 26
^C
sahil@18cfbe74344a5b9:~$
```

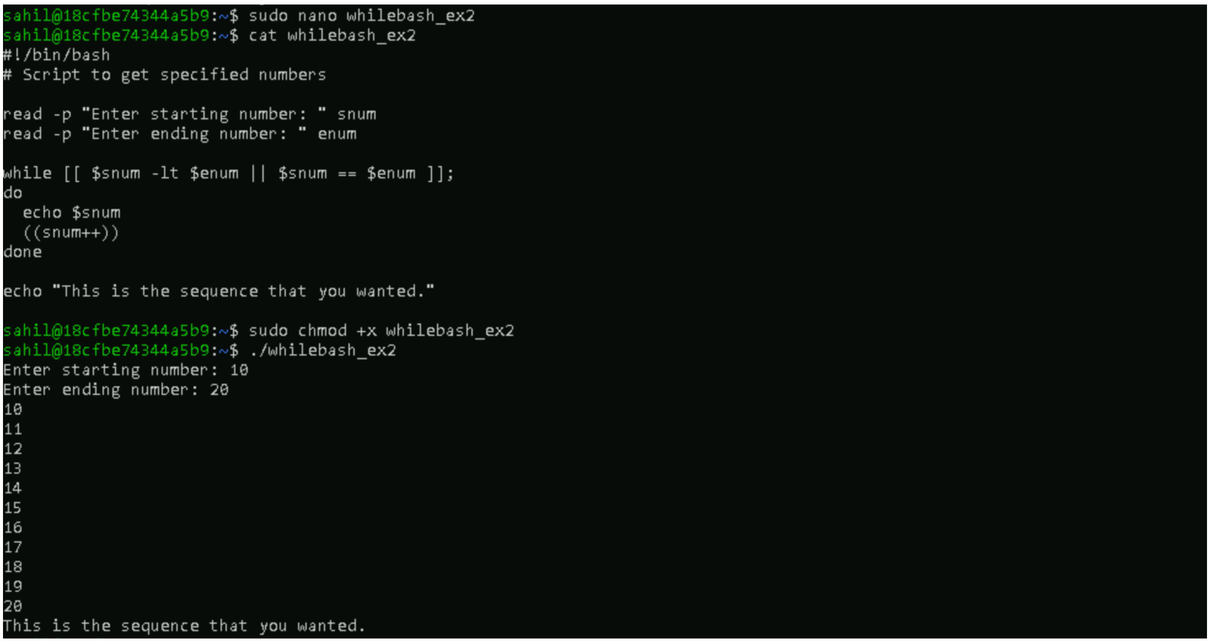
BASH WHILE LOOP:

BASH WHILE LOOP EXAMPLE 1:



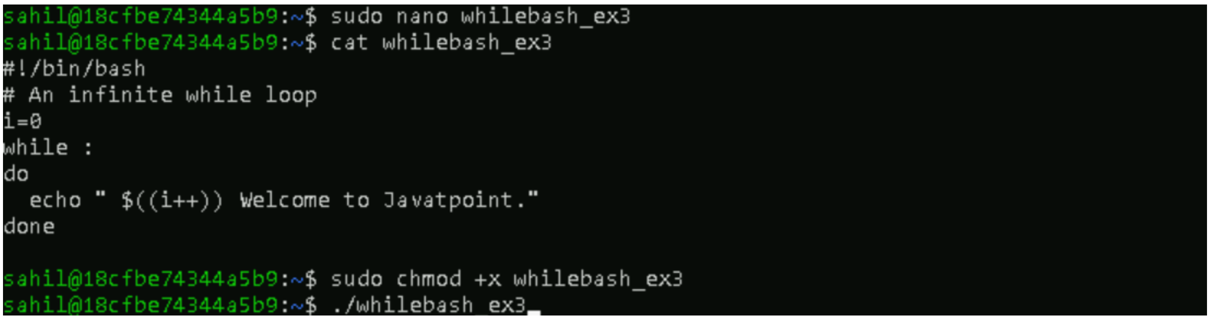
```
sahil@18cfbe74344a5b9:~$ sudo nano whilebash_ex1
sahil@18cfbe74344a5b9:~$ cat whilebash_ex1
#!/bin/bash
# Script to get specified numbers

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."

sahil@18cfbe74344a5b9:~$ sudo chmod +x whilebash_ex1
[sudo] password for sahil:
sahil@18cfbe74344a5b9:~$ ./whilebash_ex1
Enter starting number: 13
Enter ending number: 22
13
14
15
16
17
18
19
20
21
22
This is the sequence that you wanted.
sahil@18cfbe74344a5b9:~$
```

BASH WHILE LOOP EXAMPLE 2:

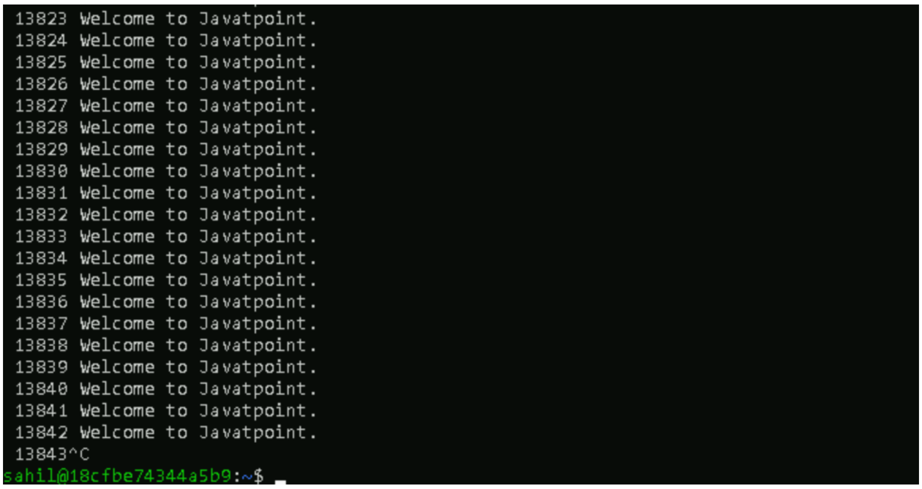


```
sahil@18cfbe74344a5b9:~$ sudo nano whilebash_ex2
sahil@18cfbe74344a5b9:~$ cat whilebash_ex2
#!/bin/bash
# Script to get specified numbers

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."

sahil@18cfbe74344a5b9:~$ sudo chmod +x whilebash_ex2
sahil@18cfbe74344a5b9:~$ ./whilebash_ex2
Enter starting number: 10
Enter ending number: 20
10
11
12
13
14
15
16
17
18
19
20
This is the sequence that you wanted.
```

BASH WHILE LOOP EXAMPLE 3:

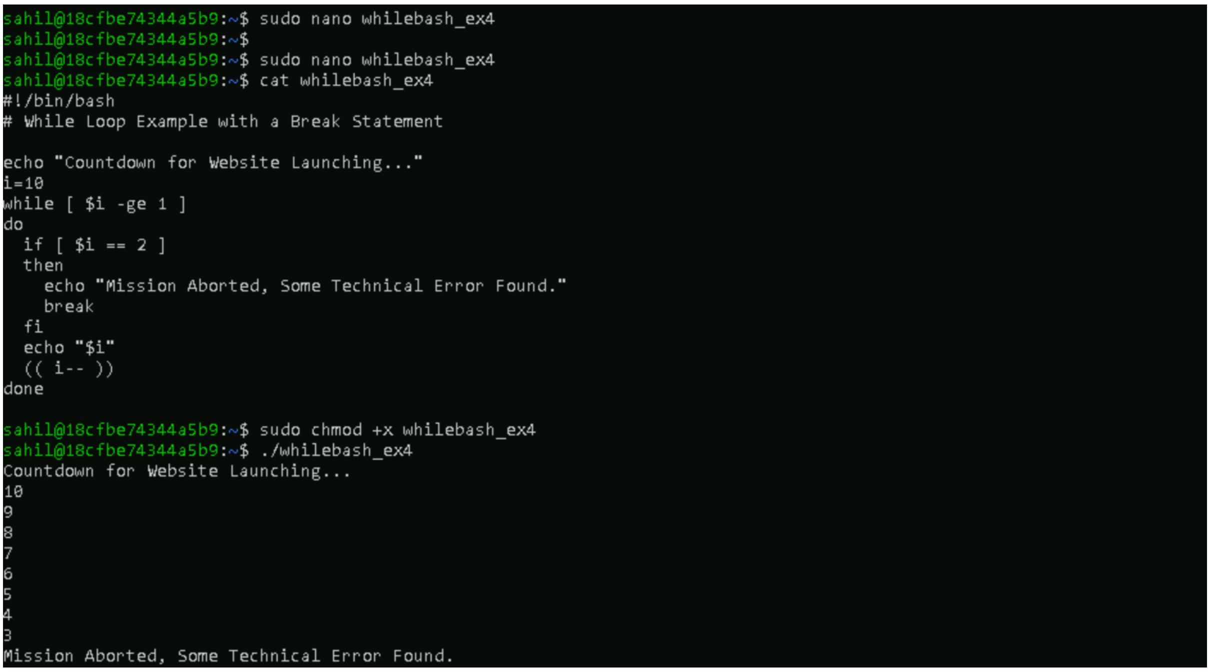


```
sahil@18cfbe74344a5b9:~$ sudo nano whilebash_ex3
sahil@18cfbe74344a5b9:~$ cat whilebash_ex3
#!/bin/bash
# An infinite while loop
i=0
while :
do
  echo " $((i++)) Welcome to Javatpoint."
done

sahil@18cfbe74344a5b9:~$ sudo chmod +x whilebash_ex3
sahil@18cfbe74344a5b9:~$ ./whilebash_ex3
```



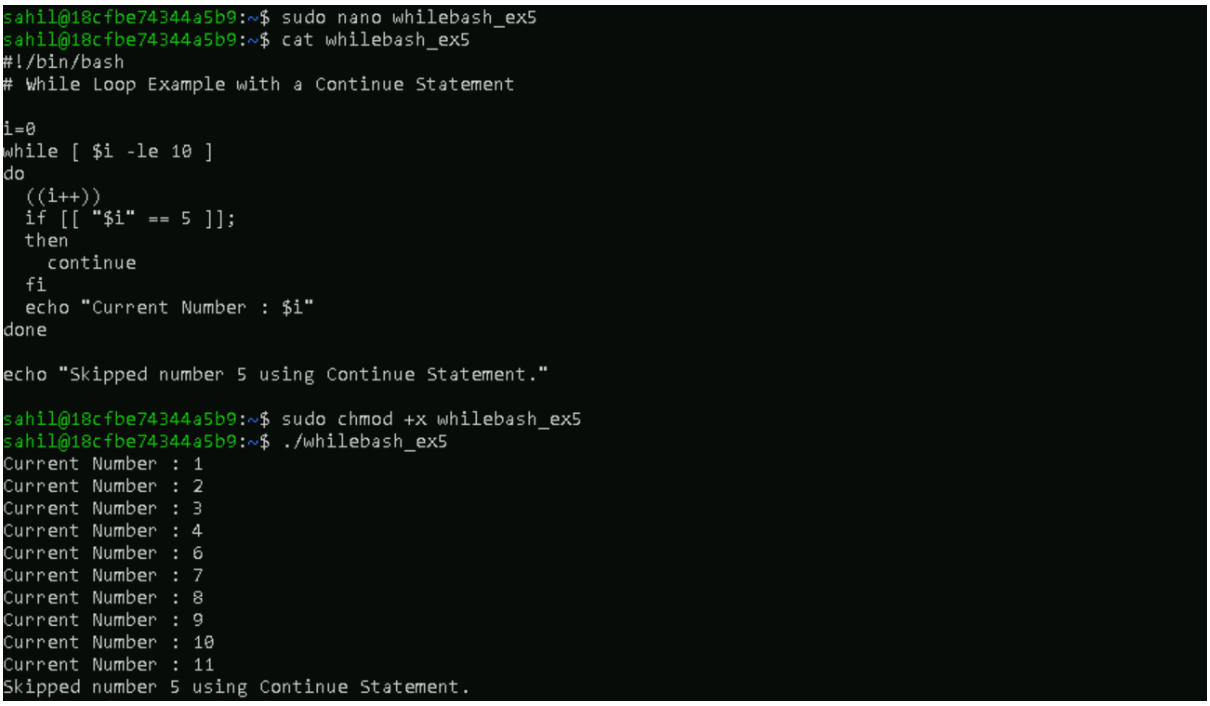
```
 13823 Welcome to Javatpoint.
 13824 Welcome to Javatpoint.
 13825 Welcome to Javatpoint.
 13826 Welcome to Javatpoint.
 13827 Welcome to Javatpoint.
 13828 Welcome to Javatpoint.
 13829 Welcome to Javatpoint.
 13830 Welcome to Javatpoint.
 13831 Welcome to Javatpoint.
 13832 Welcome to Javatpoint.
 13833 Welcome to Javatpoint.
 13834 Welcome to Javatpoint.
 13835 Welcome to Javatpoint.
 13836 Welcome to Javatpoint.
 13837 Welcome to Javatpoint.
 13838 Welcome to Javatpoint.
 13839 Welcome to Javatpoint.
 13840 Welcome to Javatpoint.
 13841 Welcome to Javatpoint.
 13842 Welcome to Javatpoint.
 13843^C
sahil@18cfbe74344a5b9:~$ 
```

BASH WHILE LOOP EXAMPLE 4:



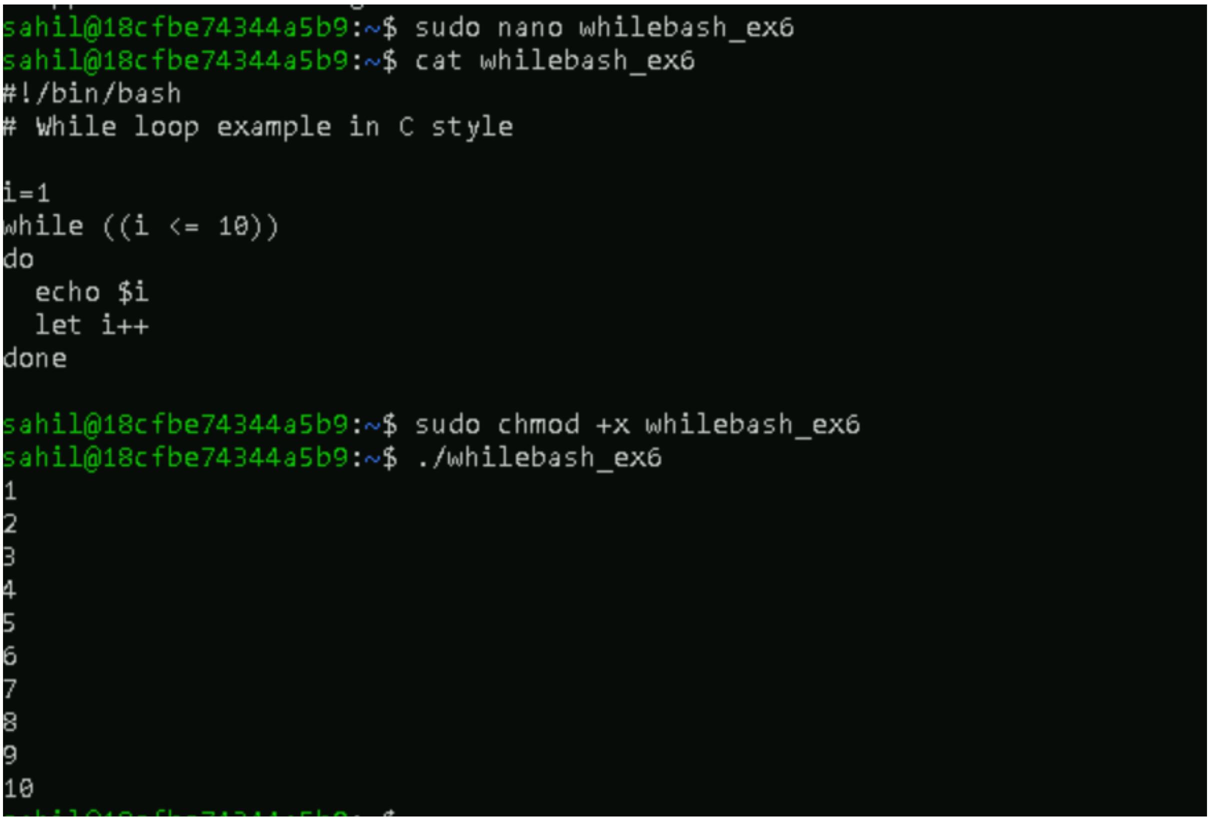
```
sahil@18cfbe74344a5b9:~$ sudo nano whilebash_ex4
sahil@18cfbe74344a5b9:~$
sahil@18cfbe74344a5b9:~$ sudo nano whilebash_ex4
sahil@18cfbe74344a5b9:~$ cat whilebash_ex4
#!/bin/bash
# While Loop Example with a Break Statement

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

sahil@18cfbe74344a5b9:~$ sudo chmod +x whilebash_ex4
sahil@18cfbe74344a5b9:~$ ./whilebash_ex4
Countdown for Website Launching...
10
9
8
7
6
5
4
3
Mission Aborted, Some Technical Error Found.
```

BASH WHILE LOOP EXAMPLE 5:



```
sahil@18cfbe74344a5b9:~$ sudo nano whilebash_ex5
sahil@18cfbe74344a5b9:~$ cat whilebash_ex5
#!/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."

sahil@18cfbe74344a5b9:~$ sudo chmod +x whilebash_ex5
sahil@18cfbe74344a5b9:~$ ./whilebash_ex5
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.
```

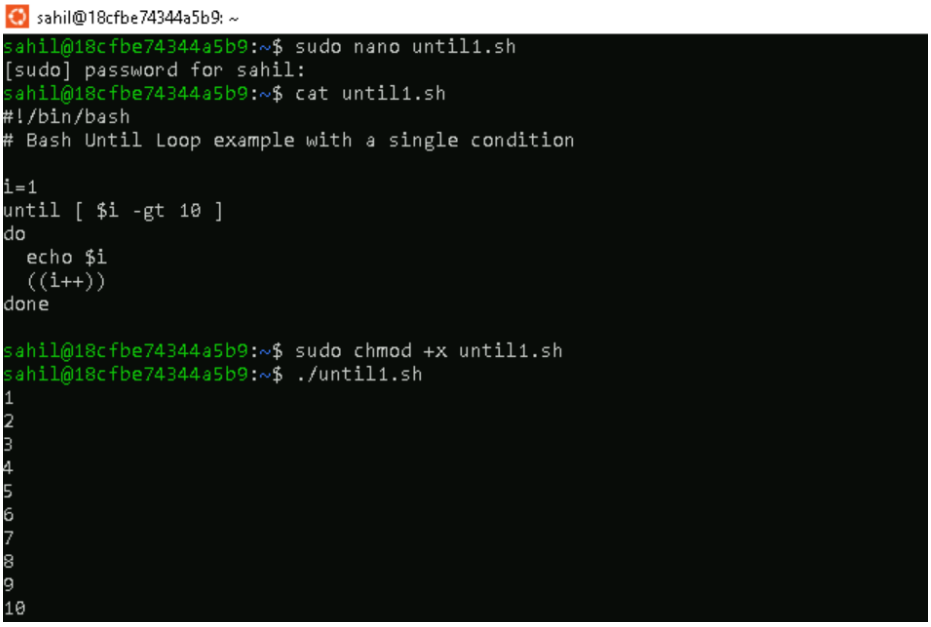
BASH WHILE LOOP EXAMPLE 6:



```
sahil@18cfbe74344a5b9:~$ sudo nano whilebash_ex6
sahil@18cfbe74344a5b9:~$ cat whilebash_ex6
#!/bin/bash
# While loop example in C style

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

sahil@18cfbe74344a5b9:~$ sudo chmod +x whilebash_ex6
sahil@18cfbe74344a5b9:~$ ./whilebash_ex6
1
2
3
4
5
6
7
8
9
10
```

BASH UNTIL LOOP:

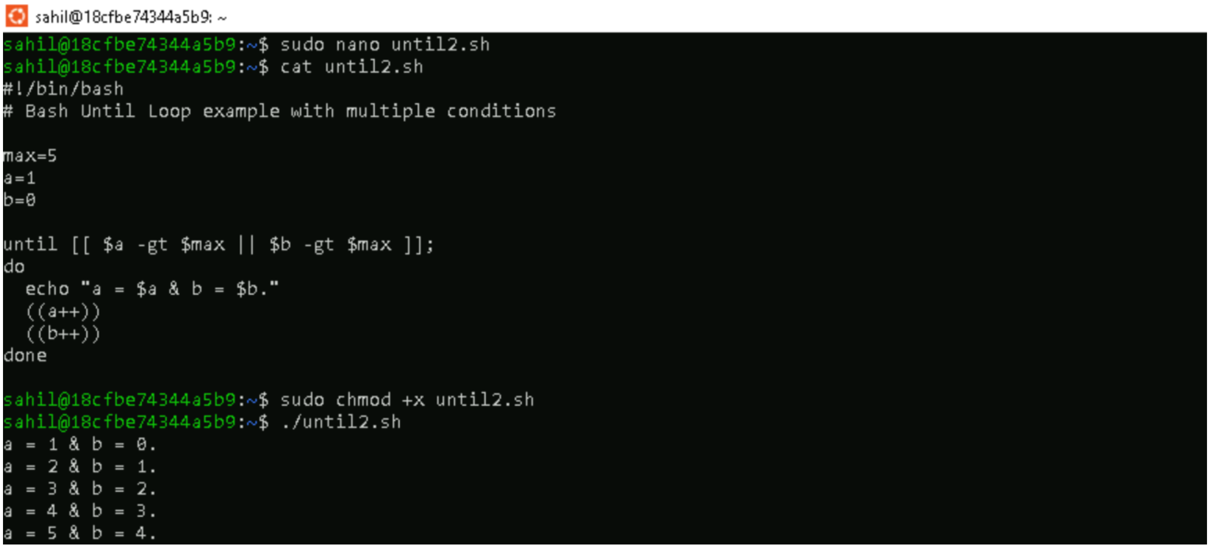
BASH UNTIL LOOP EXAMPLE 1:



sahil@18cfbe74344a5b9: ~

```
sahil@18cfbe74344a5b9:~$ sudo nano until1.sh
[sudo] password for sahil:
sahil@18cfbe74344a5b9:~$ cat until1.sh
#!/bin/bash
# Bash Until Loop example with a single condition

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

sahil@18cfbe74344a5b9:~$ sudo chmod +x until1.sh
sahil@18cfbe74344a5b9:~$ ./until1.sh
1
2
3
4
5
6
7
8
9
10
```

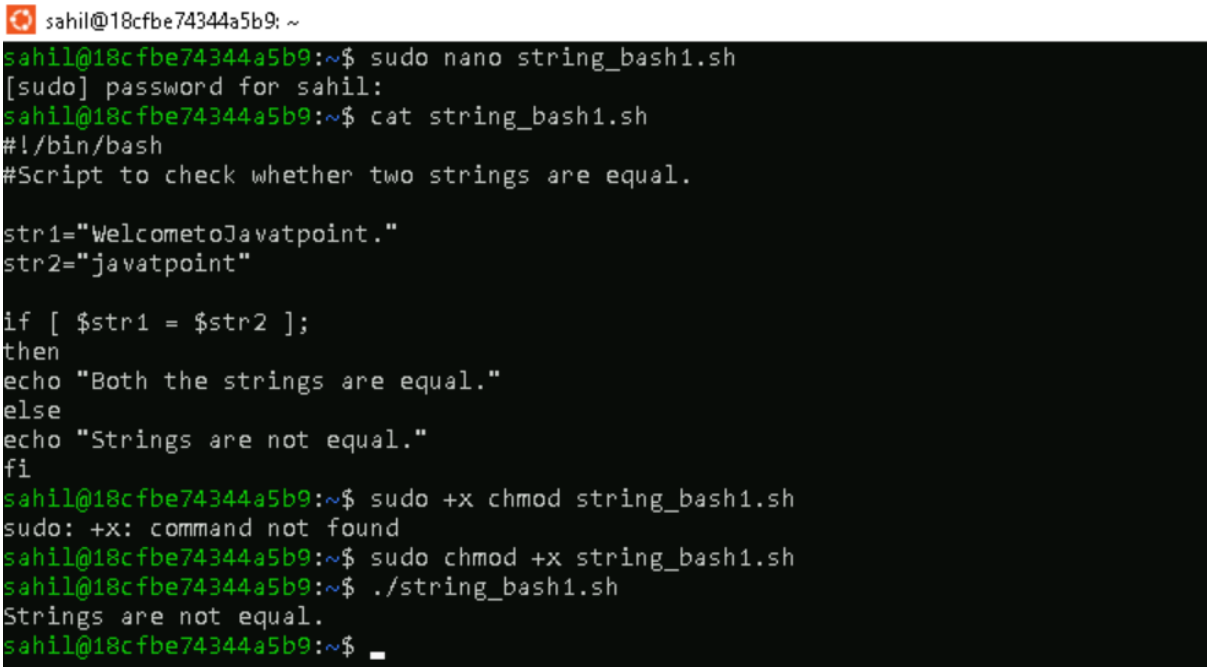
BASH UNTIL LOOP EXAMPLE 2:



```
sahil@18cfbe74344a5b9:~$ sudo nano until2.sh
sahil@18cfbe74344a5b9:~$ cat until2.sh
#!/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

sahil@18cfbe74344a5b9:~$ sudo chmod +x until2.sh
sahil@18cfbe74344a5b9:~$ ./until2.sh
a = 1 & b = 0.
a = 2 & b = 1.
a = 3 & b = 2.
a = 4 & b = 3.
a = 5 & b = 4.
```

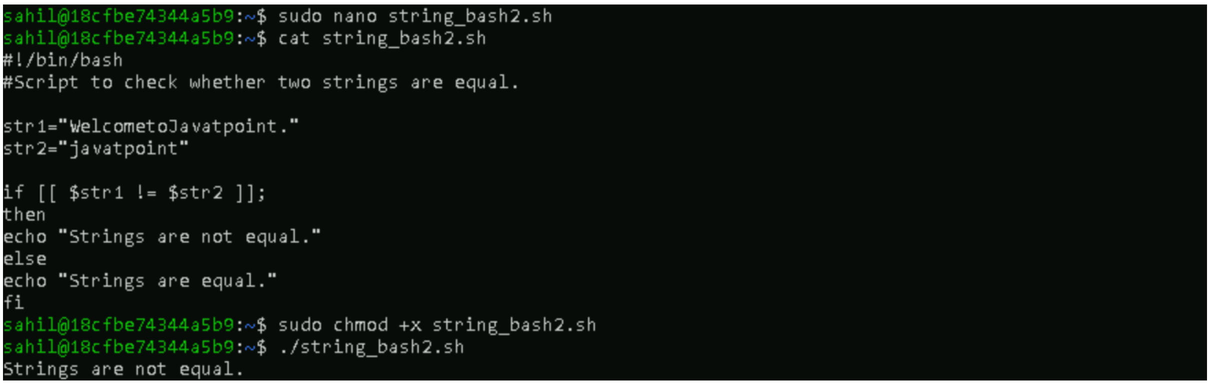
BASH STRING:

BASH STRING EXAMPLE 1:



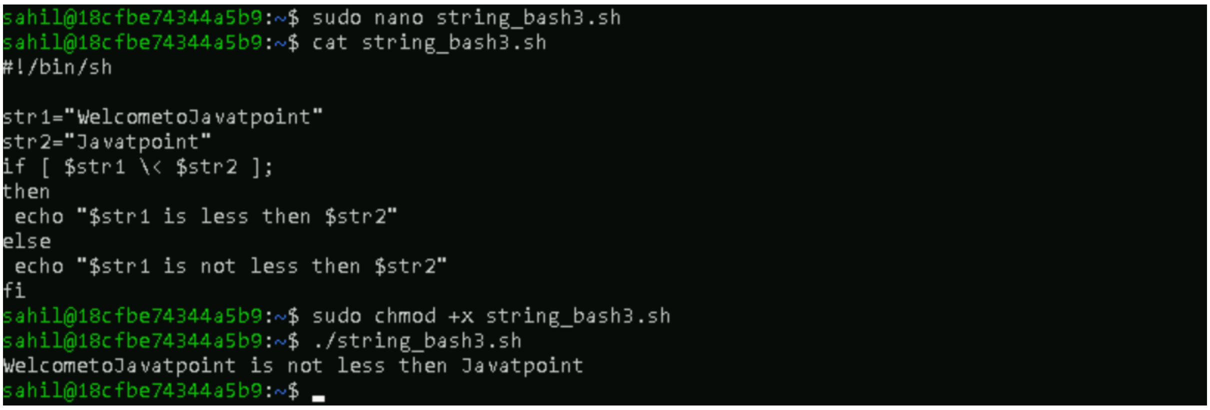
```
sahil@18cfbe74344a5b9:~$ sudo nano string_bash1.sh
[sudo] password for sahil:
sahil@18cfbe74344a5b9:~$ cat string_bash1.sh
#!/bin/bash
#Script to check whether two strings are equal.

str1="WelcometoJavatpoint."
str2="javatpoint"

if [ $str1 = $str2 ];
then
echo "Both the strings are equal."
else
echo "Strings are not equal."
fi
sahil@18cfbe74344a5b9:~$ sudo +x chmod string_bash1.sh
sudo: +x: command not found
sahil@18cfbe74344a5b9:~$ sudo chmod +x string_bash1.sh
sahil@18cfbe74344a5b9:~$ ./string_bash1.sh
Strings are not equal.
sahil@18cfbe74344a5b9:~$ _
```

BASH STRING EXAMPLE 2:



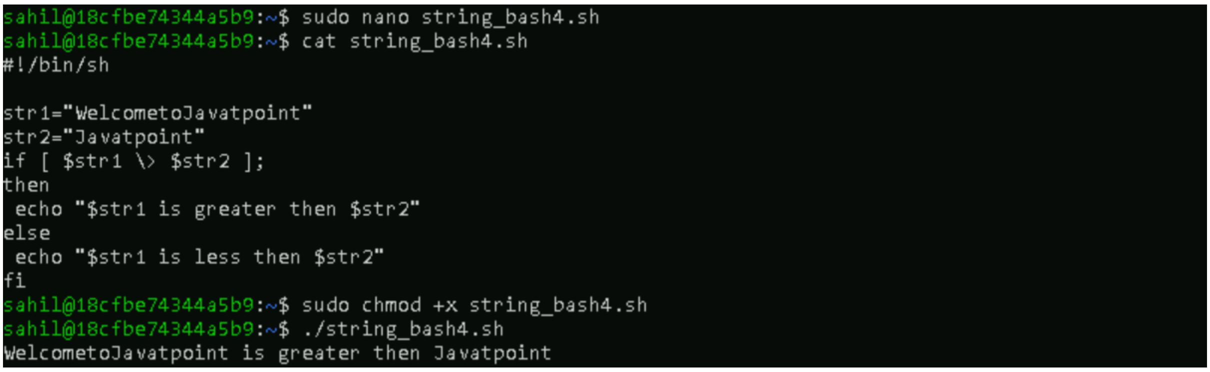
```
sahil@18cfbe74344a5b9:~$ sudo nano string_bash2.sh
sahil@18cfbe74344a5b9:~$ cat string_bash2.sh
#!/bin/bash
#Script to check whether two strings are equal.

str1="WelcometoJavatpoint."
str2="javatpoint"

if [[ $str1 != $str2 ]];
then
echo "Strings are not equal."
else
echo "Strings are equal."
fi
sahil@18cfbe74344a5b9:~$ sudo chmod +x string_bash2.sh
sahil@18cfbe74344a5b9:~$ ./string_bash2.sh
Strings are not equal.
```

BASH STRING EXAMPLE 3:



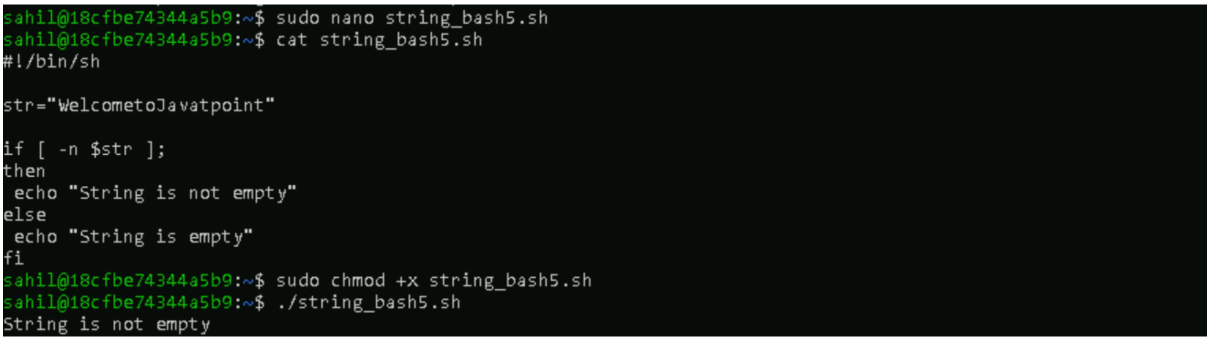
```
sahil@18cfbe74344a5b9:~$ sudo nano string_bash3.sh
sahil@18cfbe74344a5b9:~$ cat string_bash3.sh
#!/bin/sh

str1="WelcometoJavatpoint"
str2="Javatpoint"
if [ $str1 \< $str2 ];
then
 echo "$str1 is less then $str2"
else
 echo "$str1 is not less then $str2"
fi
sahil@18cfbe74344a5b9:~$ sudo chmod +x string_bash3.sh
sahil@18cfbe74344a5b9:~$ ./string_bash3.sh
WelcometoJavatpoint is not less then Javatpoint
sahil@18cfbe74344a5b9:~$ _
```

BASH STRING EXAMPLE 4:



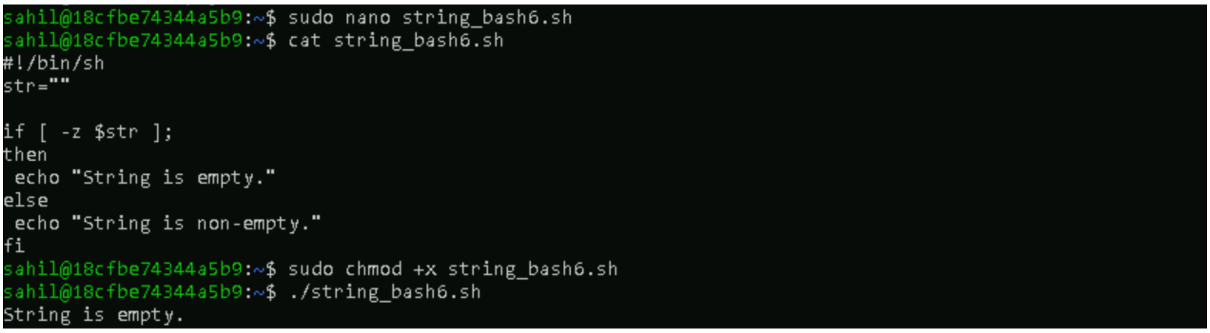
```
sahil@18cfbe74344a5b9:~$ sudo nano string_bash4.sh
sahil@18cfbe74344a5b9:~$ cat string_bash4.sh
#!/bin/sh

str1="WelcometoJavatpoint"
str2="Javatpoint"
if [ $str1 \> $str2 ];
then
 echo "$str1 is greater then $str2"
else
 echo "$str1 is less then $str2"
fi
sahil@18cfbe74344a5b9:~$ sudo chmod +x string_bash4.sh
sahil@18cfbe74344a5b9:~$ ./string_bash4.sh
WelcometoJavatpoint is greater then Javatpoint
```

BASH STRING EXAMPLE 5:



```
sahil@18cfbe74344a5b9:~$ sudo nano string_bash5.sh
sahil@18cfbe74344a5b9:~$ cat string_bash5.sh
#!/bin/sh

str="WelcometoJavatpoint"

if [ -n $str ];
then
 echo "String is not empty"
else
 echo "String is empty"
fi
sahil@18cfbe74344a5b9:~$ sudo chmod +x string_bash5.sh
sahil@18cfbe74344a5b9:~$ ./string_bash5.sh
String is not empty
```

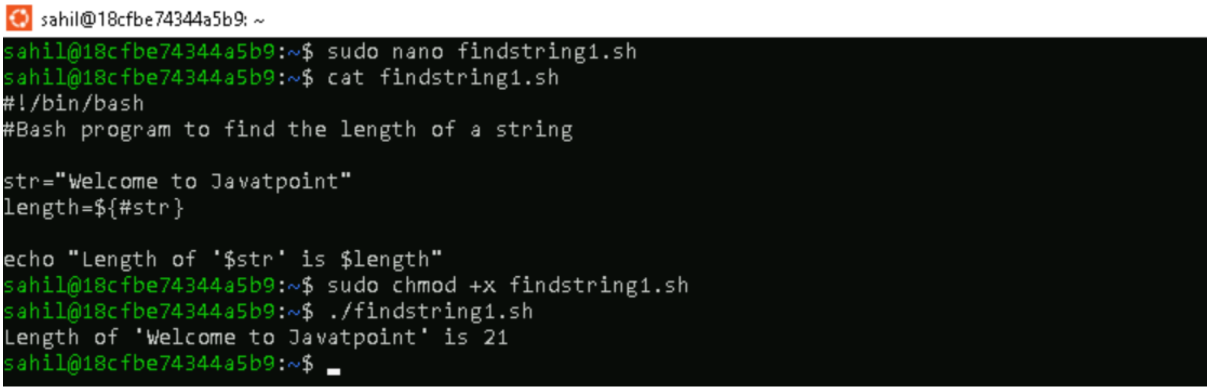
BASH STRING EXAMPLE 6:



```
sahil@18cfbe74344a5b9:~$ sudo nano string_bash6.sh
sahil@18cfbe74344a5b9:~$ cat string_bash6.sh
#!/bin/sh
str=""

if [ -z $str ];
then
 echo "String is empty."
else
 echo "String is non-empty."
fi
sahil@18cfbe74344a5b9:~$ sudo chmod +x string_bash6.sh
sahil@18cfbe74344a5b9:~$ ./string_bash6.sh
String is empty.
```

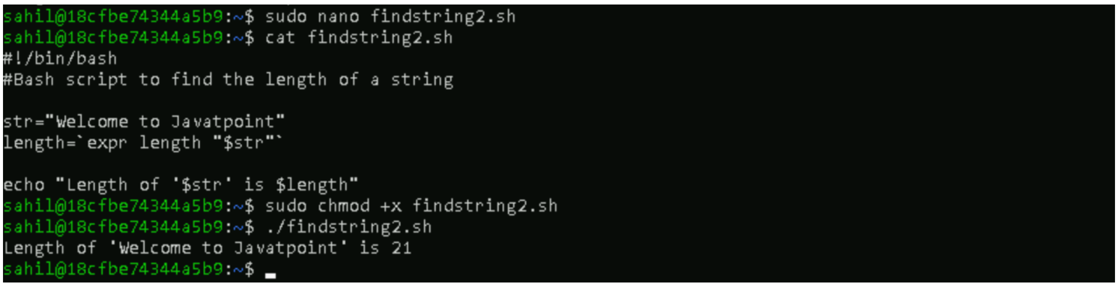
BASH FIND STRING:

BASH FIND STRING EXAMPLE 1:



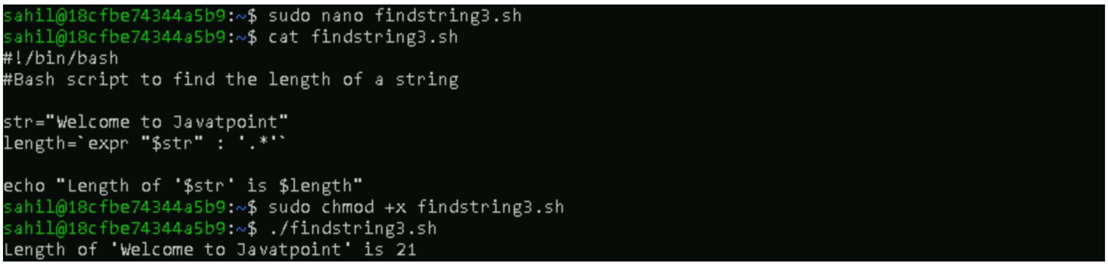
```
sahil@18cfbe74344a5b9:~$ sudo nano findstring1.sh
sahil@18cfbe74344a5b9:~$ cat findstring1.sh
#!/bin/bash
#Bash program to find the length of a string

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

echo "Length of '$str' is $length"
sahil@18cfbe74344a5b9:~$ sudo chmod +x findstring1.sh
sahil@18cfbe74344a5b9:~$ ./findstring1.sh
Length of 'Welcome to Javatpoint' is 21
sahil@18cfbe74344a5b9:~$ 
```

BASH FIND STRING EXAMPLE 2:



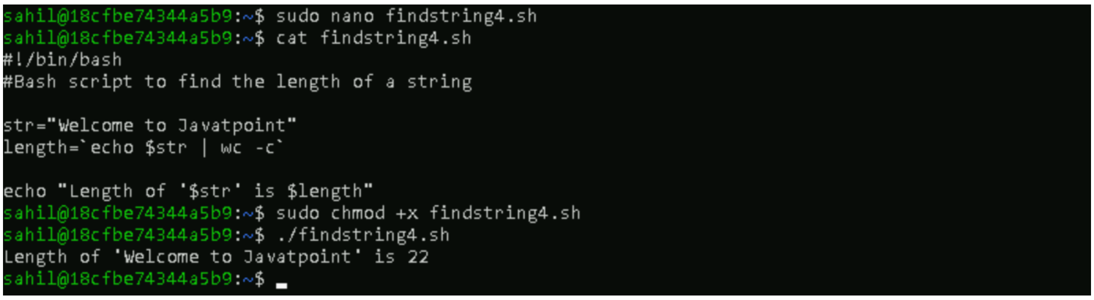
```
sahil@18cfbe74344a5b9:~$ sudo nano findstring2.sh
sahil@18cfbe74344a5b9:~$ cat findstring2.sh
#!/bin/bash
#Bash script to find the length of a string

str="Welcome to Javatpoint"
length=`expr length "$str"`

echo "Length of '$str' is $length"
sahil@18cfbe74344a5b9:~$ sudo chmod +x findstring2.sh
sahil@18cfbe74344a5b9:~$ ./findstring2.sh
Length of 'Welcome to Javatpoint' is 21
sahil@18cfbe74344a5b9:~$
```

BASH FIND STRING EXAMPLE 3:



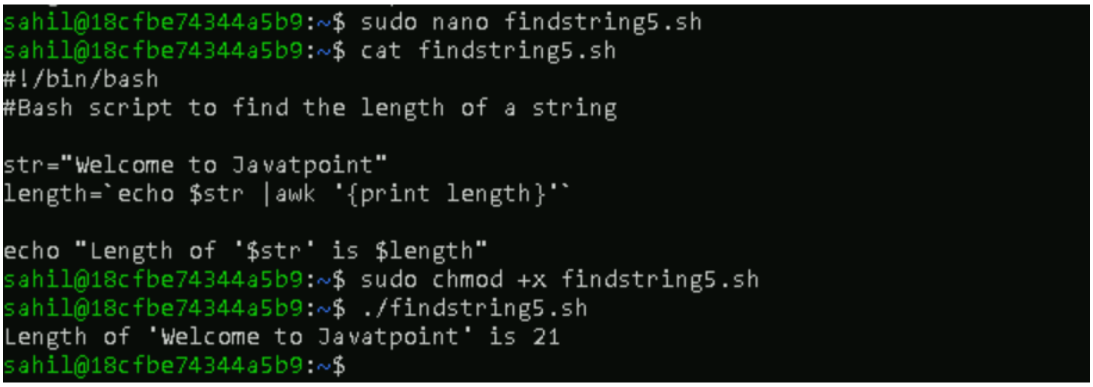
```
sahil@18cfbe74344a5b9:~$ sudo nano findstring3.sh
sahil@18cfbe74344a5b9:~$ cat findstring3.sh
#!/bin/bash
#Bash script to find the length of a string

str="Welcome to Javatpoint"
length=`expr "$str" : '.*'`

echo "Length of '$str' is $length"
sahil@18cfbe74344a5b9:~$ sudo chmod +x findstring3.sh
sahil@18cfbe74344a5b9:~$ ./findstring3.sh
Length of 'Welcome to Javatpoint' is 21
```

BASH FIND STRING EXAMPLE 4:



```
sahil@18cfbe74344a5b9:~$ sudo nano findstring4.sh
sahil@18cfbe74344a5b9:~$ cat findstring4.sh
#!/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 $length"
sahil@18cfbe74344a5b9:~$ sudo chmod +x findstring4.sh
sahil@18cfbe74344a5b9:~$ ./findstring4.sh
Length of 'Welcome to Javatpoint' is 22
sahil@18cfbe74344a5b9:~$
```

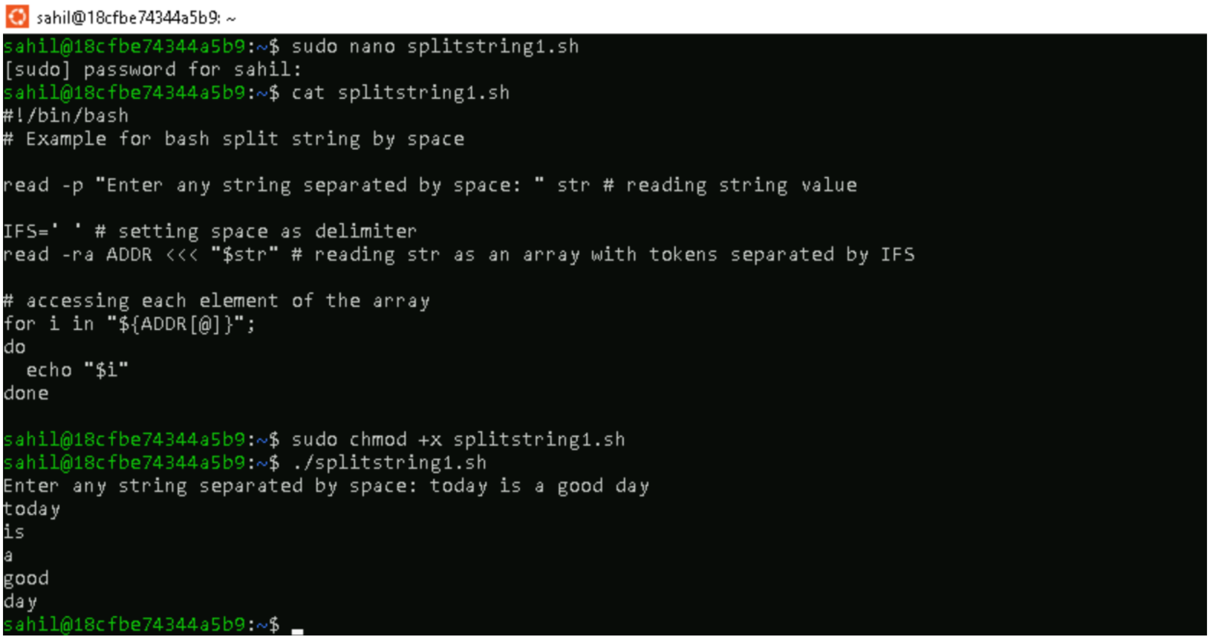
BASH FIND STRING EXAMPLE 5:



```
sahil@18cfbe74344a5b9:~$ sudo nano findstring5.sh
sahil@18cfbe74344a5b9:~$ cat findstring5.sh
#!/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 $length"
sahil@18cfbe74344a5b9:~$ sudo chmod +x findstring5.sh
sahil@18cfbe74344a5b9:~$ ./findstring5.sh
Length of 'Welcome to Javatpoint' is 21
sahil@18cfbe74344a5b9:~$
```

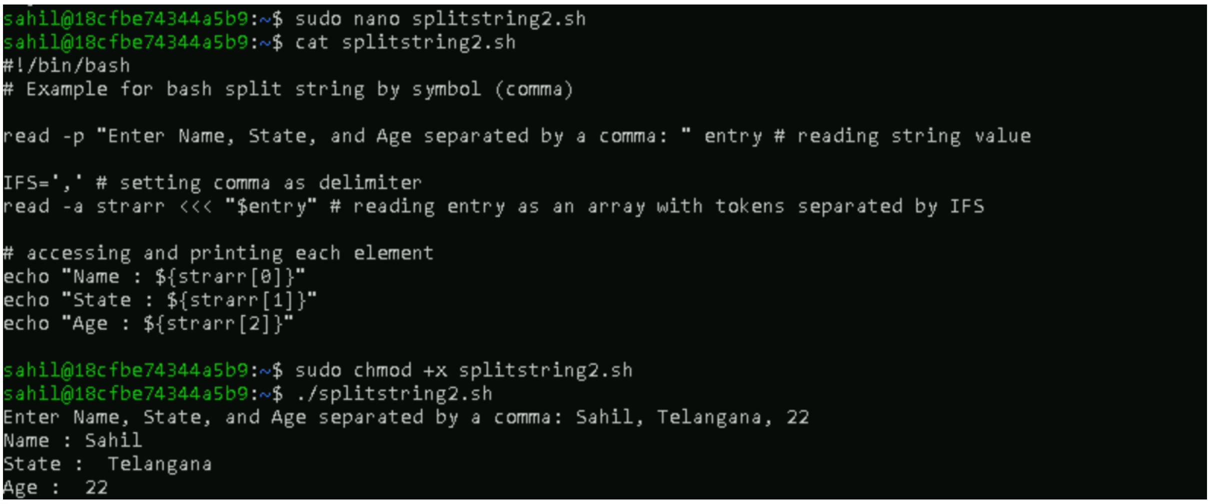
BASH SPLIT STRING:

BASH SPLIT STRING EXAMPLE 1:



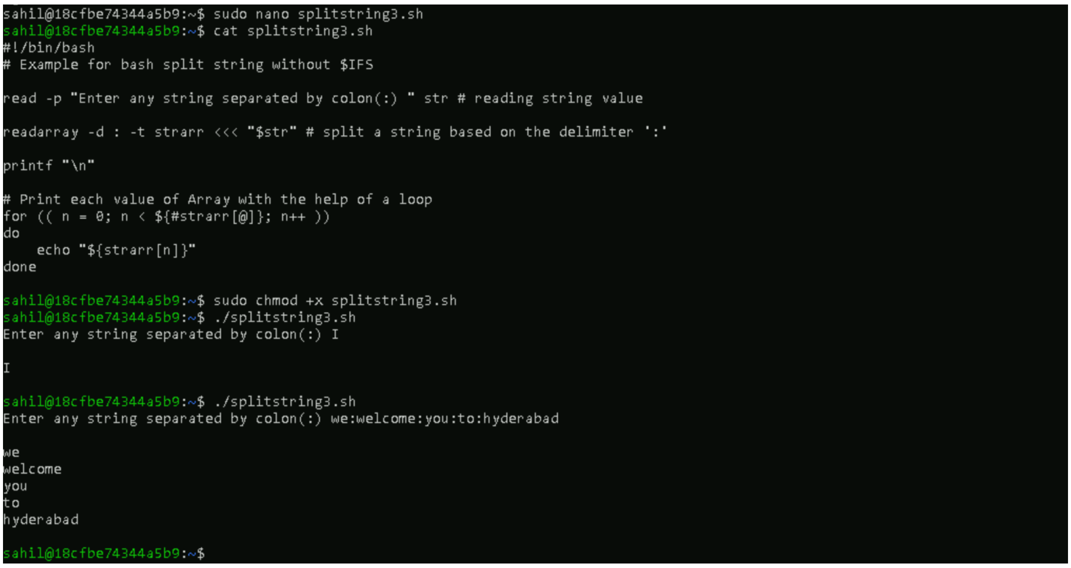
```
sahil@18cfbe74344a5b9:~$ sudo nano splitstring1.sh
[sudo] password for sahil:
sahil@18cfbe74344a5b9:~$ cat splitstring1.sh
#!/bin/bash
# Example for bash split string by space

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 with tokens separated by IFS

# accessing each element of the array
for i in "${ADDR[@]}";
do
  echo "$i"
done

sahil@18cfbe74344a5b9:~$ sudo chmod +x splitstring1.sh
sahil@18cfbe74344a5b9:~$ ./splitstring1.sh
Enter any string separated by space: today is a good day
today
is
a
good
day
sahil@18cfbe74344a5b9:~$ _
```

BASH SPLIT STRING EXAMPLE 2:



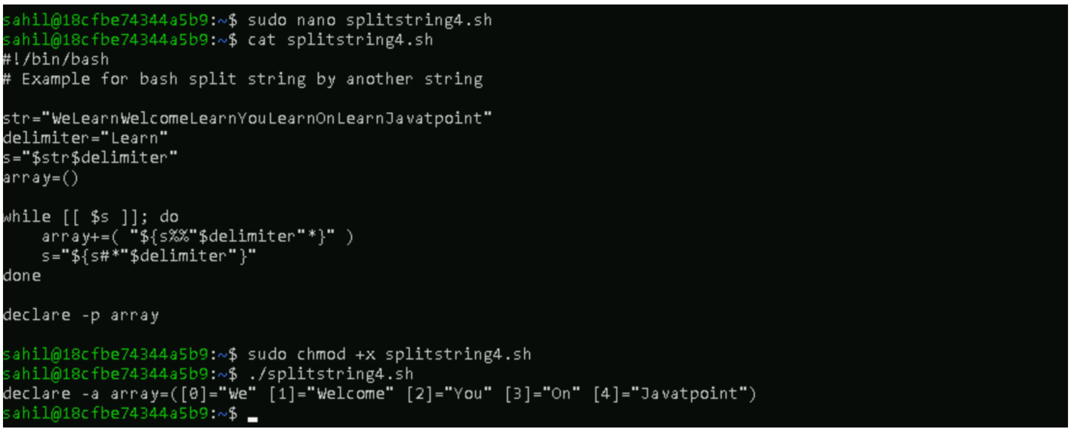
```
sahil@18cfbe74344a5b9:~$ sudo nano splitstring2.sh
sahil@18cfbe74344a5b9:~$ cat splitstring2.sh
#!/bin/bash
# Example for bash split string by symbol (comma)

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 entry as an array with tokens separated by IFS

# accessing and printing each element
echo "Name : ${strarr[0]}"
echo "State : ${strarr[1]}"
echo "Age : ${strarr[2]}"

sahil@18cfbe74344a5b9:~$ sudo chmod +x splitstring2.sh
sahil@18cfbe74344a5b9:~$ ./splitstring2.sh
Enter Name, State, and Age separated by a comma: Sahil, Telangana, 22
Name : Sahil
State :  Telangana
Age :  22
```

BASH SPLIT STRING EXAMPLE 3:



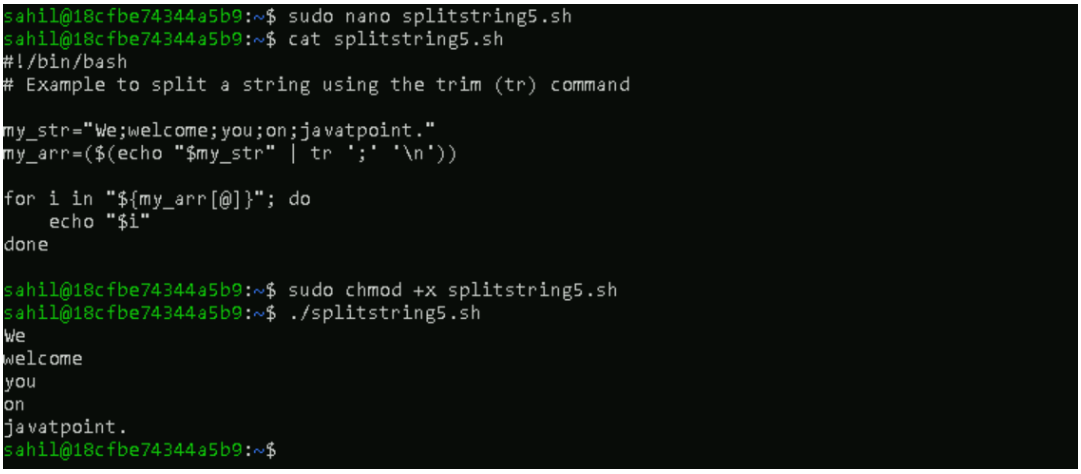
```
sahil@18cfbe74344a5b9:~$ sudo nano splitstring3.sh
sahil@18cfbe74344a5b9:~$ cat splitstring3.sh
#!/bin/bash
# Example for bash split string without $IFS

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 a loop
for (( n = 0; n < ${#strarr[@]}; n++ ))
do
    echo "${strarr[n]}"
done

sahil@18cfbe74344a5b9:~$ sudo chmod +x splitstring3.sh
sahil@18cfbe74344a5b9:~$ ./splitstring3.sh
Enter any string separated by colon(:) I

I

sahil@18cfbe74344a5b9:~$ ./splitstring3.sh
Enter any string separated by colon(:) we:welcome:you:to:hyderabad

we
welcome
you
to
hyderabad

sahil@18cfbe74344a5b9:~$
```

BASH SPLIT STRING EXAMPLE 4:



```
sahil@18cfbe74344a5b9:~$ sudo nano splitstring4.sh
sahil@18cfbe74344a5b9:~$ cat splitstring4.sh
#!/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

sahil@18cfbe74344a5b9:~$ sudo chmod +x splitstring4.sh
sahil@18cfbe74344a5b9:~$ ./splitstring4.sh
declare -a array=([0]="We" [1]="Welcome" [2]="You" [3]="On" [4]="Javatpoint")
sahil@18cfbe74344a5b9:~$ _
```

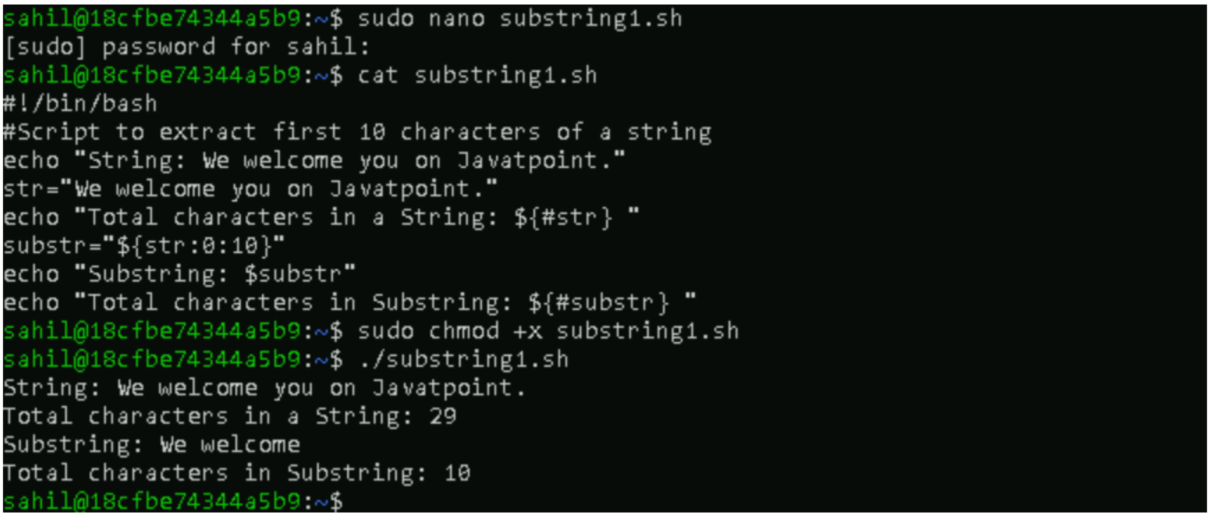
BASH SPLIT STRING EXAMPLE 5:



```
sahil@18cfbe74344a5b9:~$ sudo nano splitstring5.sh
sahil@18cfbe74344a5b9:~$ cat splitstring5.sh
#!/bin/bash
# Example to split a string using the trim (tr) command

my_str="We;welcome;you;on;javatpoint."
my_arr=($(echo "$my_str" | tr ';' '\n'))

for i in "${my_arr[@]}"; do
    echo "$i"
done

sahil@18cfbe74344a5b9:~$ sudo chmod +x splitstring5.sh
sahil@18cfbe74344a5b9:~$ ./splitstring5.sh
We
welcome
you
on
javatpoint.
sahil@18cfbe74344a5b9:~$
```

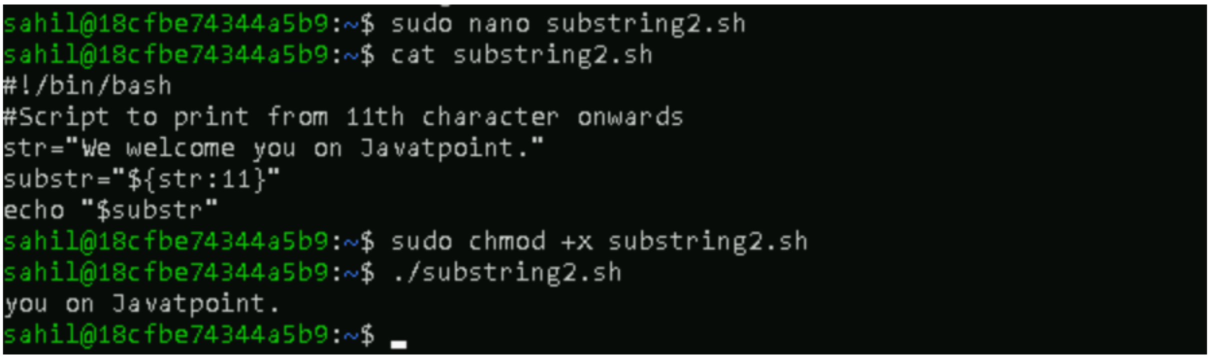
BASH SUB STRING:

BASH SUB STRING EXAMPLE 1:



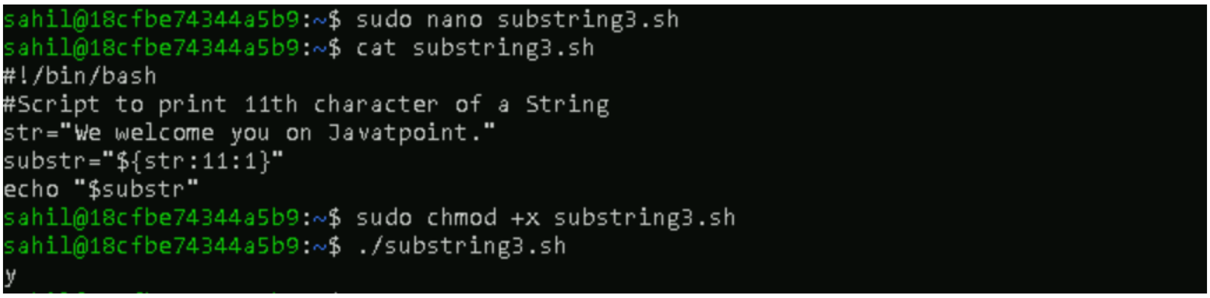
```
sahil@18cfbe74344a5b9:~$ sudo nano substring1.sh
[sudo] password for sahil:
sahil@18cfbe74344a5b9:~$ cat substring1.sh
#!/bin/bash
#Script to extract first 10 characters of a string
echo "String: We welcome you on Javatpoint."
str="We welcome you on Javatpoint."
echo "Total characters in a String: ${#str} "
substr="${str:0:10}"
echo "Substring: $substr"
echo "Total characters in Substring: ${#substr} "
sahil@18cfbe74344a5b9:~$ sudo chmod +x substring1.sh
sahil@18cfbe74344a5b9:~$ ./substring1.sh
String: We welcome you on Javatpoint.
Total characters in a String: 29
Substring: We welcome
Total characters in Substring: 10
sahil@18cfbe74344a5b9:~$
```

BASH SUB STRING EXAMPLE 2:



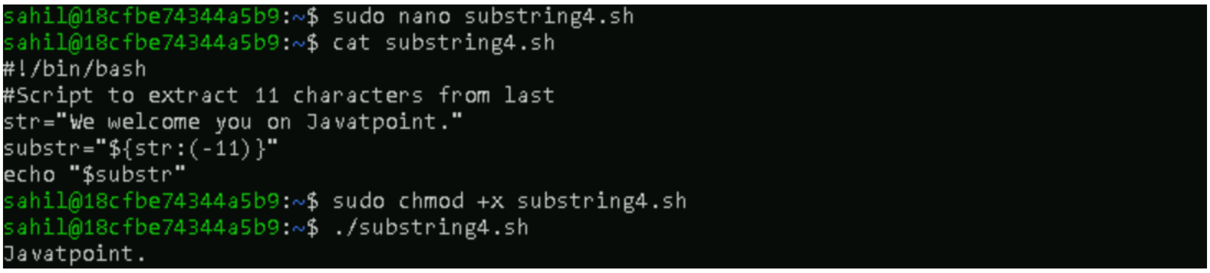
```
sahil@18cfbe74344a5b9:~$ sudo nano substring2.sh
sahil@18cfbe74344a5b9:~$ cat substring2.sh
#!/bin/bash
#Script to print from 11th character onwards
str="We welcome you on Javatpoint."
substr="${str:11}"
echo "$substr"
sahil@18cfbe74344a5b9:~$ sudo chmod +x substring2.sh
sahil@18cfbe74344a5b9:~$ ./substring2.sh
you on Javatpoint.
sahil@18cfbe74344a5b9:~$ _
```

BASH SUB STRING EXAMPLE 3:



```
sahil@18cfbe74344a5b9:~$ sudo nano substring3.sh
sahil@18cfbe74344a5b9:~$ cat substring3.sh
#!/bin/bash
#Script to print 11th character of a String
str="We welcome you on Javatpoint."
substr="${str:11:1}"
echo "$substr"
sahil@18cfbe74344a5b9:~$ sudo chmod +x substring3.sh
sahil@18cfbe74344a5b9:~$ ./substring3.sh
y
```

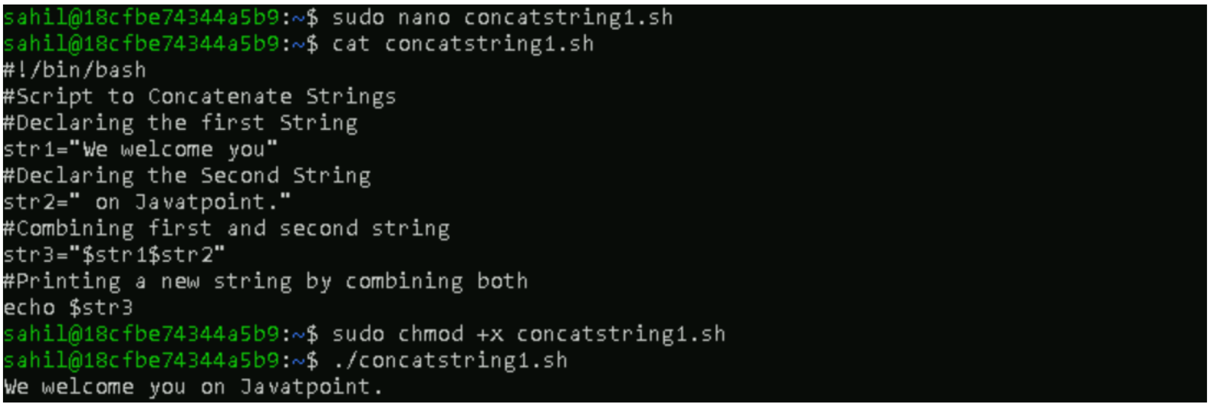
BASH SUB STRING EXAMPLE 4:



```
sahil@18cfbe74344a5b9:~$ sudo nano substring4.sh
sahil@18cfbe74344a5b9:~$ cat substring4.sh
#!/bin/bash
#Script to extract 11 characters from last
str="We welcome you on Javatpoint."
substr="${str:(-11)}"
echo "$substr"
sahil@18cfbe74344a5b9:~$ sudo chmod +x substring4.sh
sahil@18cfbe74344a5b9:~$ ./substring4.sh
Javatpoint.
```

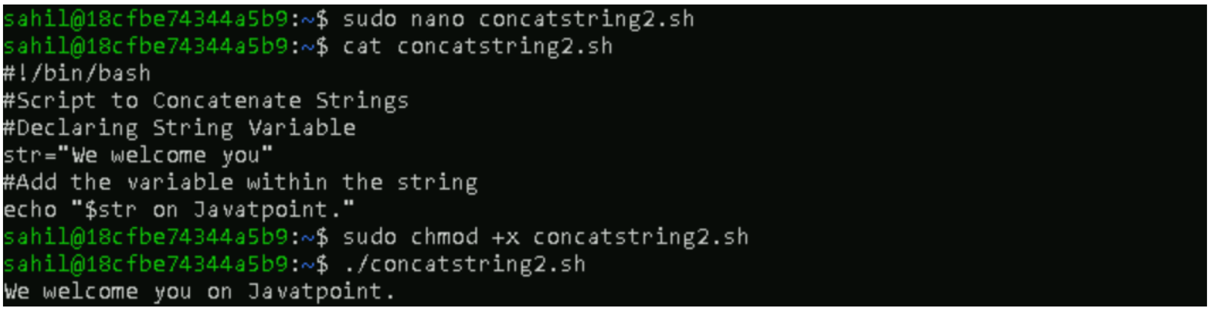
BASH STRING CONCATINATION:

BASH STRING CONCATINATION EXAMPLE 1:



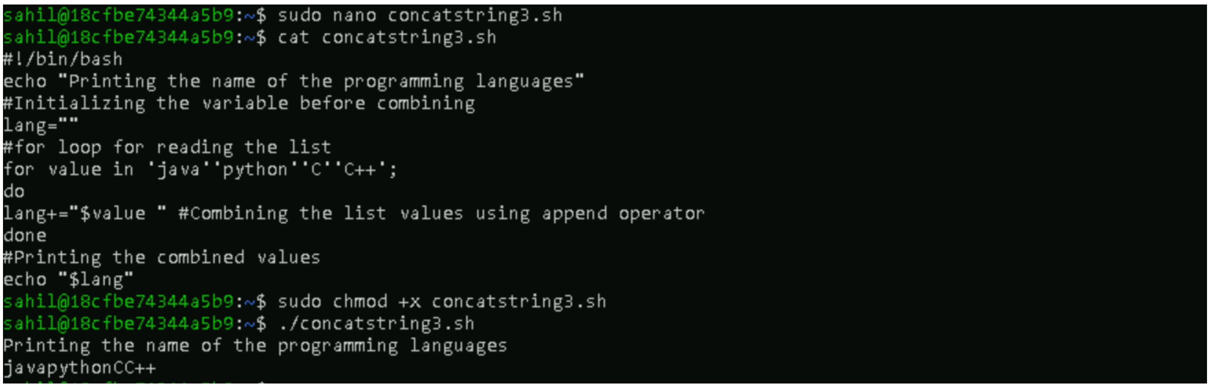
```
sahil@18cfbe74344a5b9:~$ sudo nano concatstring1.sh
sahil@18cfbe74344a5b9:~$ cat concatstring1.sh
#!/bin/bash
#Script to Concatenate Strings
#Declaring the first String
str1="We welcome you"
#Declaring the Second String
str2=" on Javatpoint."
#Combining first and second string
str3="$str1$str2"
#Printing a new string by combining both
echo $str3
sahil@18cfbe74344a5b9:~$ sudo chmod +x concatstring1.sh
sahil@18cfbe74344a5b9:~$ ./concatstring1.sh
We welcome you on Javatpoint.
```

BASH STRING CONCATINATION EXAMPLE 2:



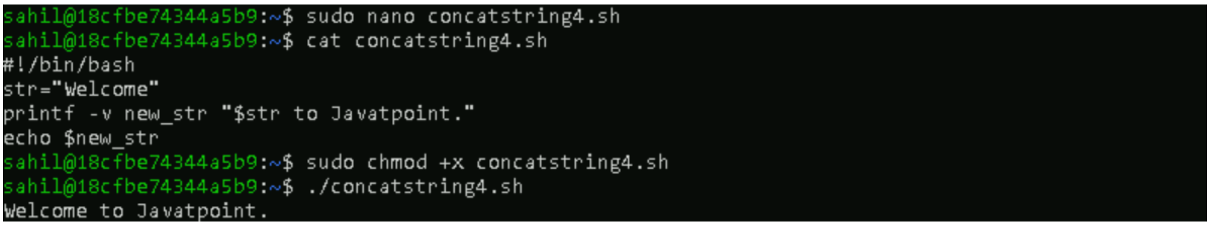
```
sahil@18cfbe74344a5b9:~$ sudo nano concatstring2.sh
sahil@18cfbe74344a5b9:~$ cat concatstring2.sh
#!/bin/bash
#Script to Concatenate Strings
#Declaring String Variable
str="We welcome you"
#Add the variable within the string
echo "$str on Javatpoint."
sahil@18cfbe74344a5b9:~$ sudo chmod +x concatstring2.sh
sahil@18cfbe74344a5b9:~$ ./concatstring2.sh
We welcome you on Javatpoint.
```

BASH STRING CONCATINATION EXAMPLE 3:



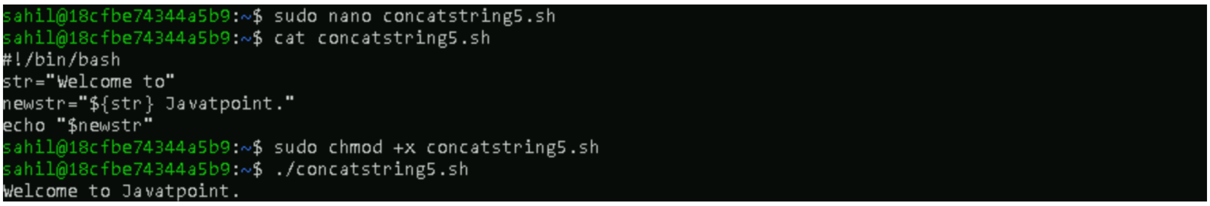
```
sahil@18cfbe74344a5b9:~$ sudo nano concatstring3.sh
sahil@18cfbe74344a5b9:~$ cat concatstring3.sh
#!/bin/bash
echo "Printing the name of the programming languages"
#Initializing the variable before combining
lang=""
#for loop for reading the list
for value in 'java''python''C''C++';
do
lang+="$value " #Combining the list values using append operator
done
#Printing the combined values
echo "$lang"
sahil@18cfbe74344a5b9:~$ sudo chmod +x concatstring3.sh
sahil@18cfbe74344a5b9:~$ ./concatstring3.sh
Printing the name of the programming languages
javapythonCC++
```

BASH STRING CONCATINATION EXAMPLE 4:



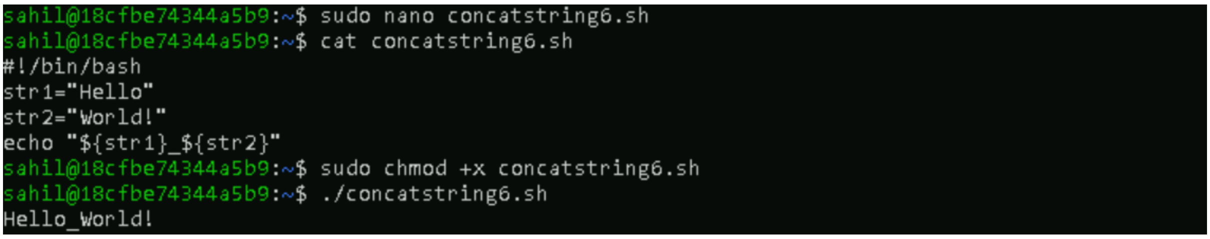
```
sahil@18cfbe74344a5b9:~$ sudo nano concatstring4.sh
sahil@18cfbe74344a5b9:~$ cat concatstring4.sh
#!/bin/bash
str="Welcome"
printf -v new_str "$str to Javatpoint."
echo $new_str
sahil@18cfbe74344a5b9:~$ sudo chmod +x concatstring4.sh
sahil@18cfbe74344a5b9:~$ ./concatstring4.sh
Welcome to Javatpoint.
```

BASH STRING CONCATINATION EXAMPLE 5:



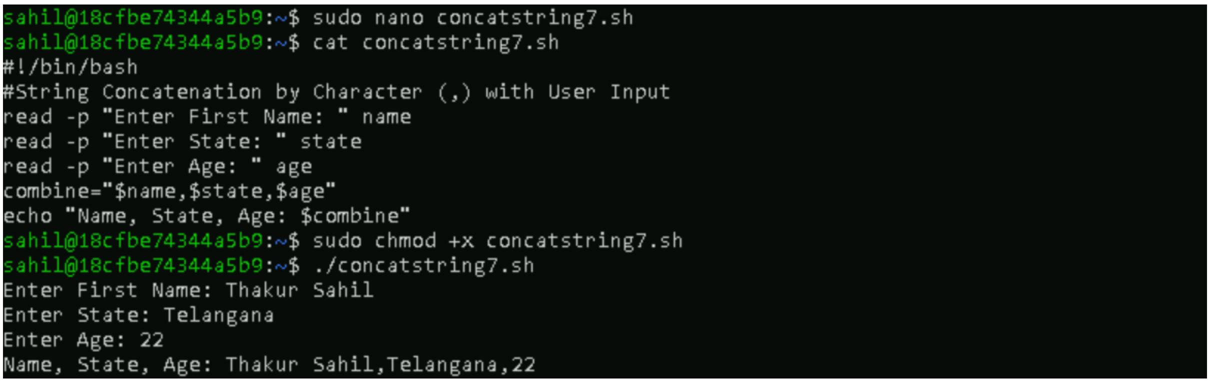
```
sahil@18cfbe74344a5b9:~$ sudo nano concatstring5.sh
sahil@18cfbe74344a5b9:~$ cat concatstring5.sh
#!/bin/bash
str="Welcome to"
newstr="${str} Javatpoint."
echo "$newstr"
sahil@18cfbe74344a5b9:~$ sudo chmod +x concatstring5.sh
sahil@18cfbe74344a5b9:~$ ./concatstring5.sh
Welcome to Javatpoint.
```

BASH STRING CONCATINATION EXAMPLE 6:



```
sahil@18cfbe74344a5b9:~$ sudo nano concatstring6.sh
sahil@18cfbe74344a5b9:~$ cat concatstring6.sh
#!/bin/bash
str1="Hello"
str2="World!"
echo "${str1}_${str2}"
sahil@18cfbe74344a5b9:~$ sudo chmod +x concatstring6.sh
sahil@18cfbe74344a5b9:~$ ./concatstring6.sh
Hello_World!
```

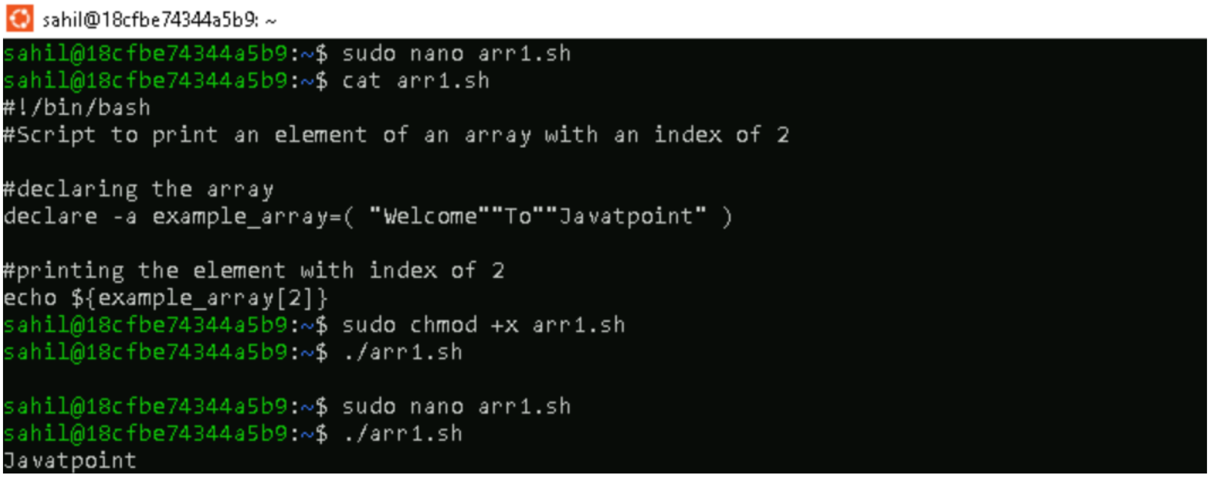
BASH STRING CONCATINATION EXAMPLE 7:



```
sahil@18cfbe74344a5b9:~$ sudo nano concatstring7.sh
sahil@18cfbe74344a5b9:~$ cat concatstring7.sh
#!/bin/bash
#String Concatenation by Character (,) with User Input
read -p "Enter First Name: " name
read -p "Enter State: " state
read -p "Enter Age: " age
combine="$name,$state,$age"
echo "Name, State, Age: $combine"
sahil@18cfbe74344a5b9:~$ sudo chmod +x concatstring7.sh
sahil@18cfbe74344a5b9:~$ ./concatstring7.sh
Enter First Name: Thakur Sahil
Enter State: Telangana
Enter Age: 22
Name, State, Age: Thakur Sahil,Telangana,22
```

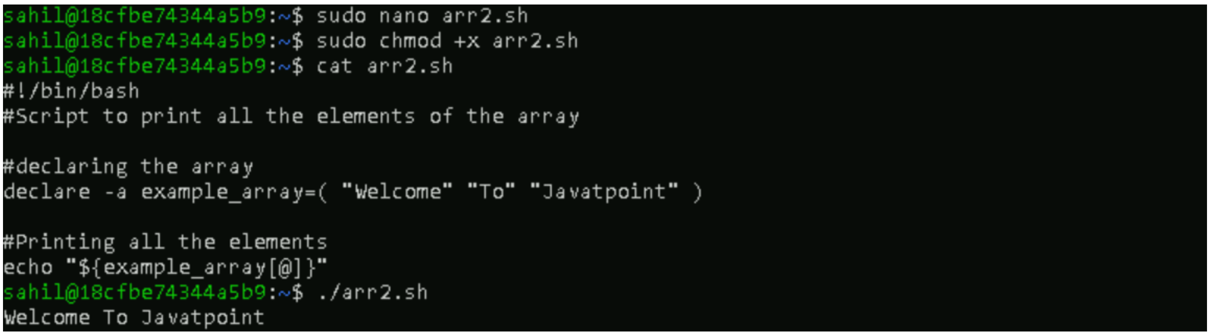
BASH ARRAY:

BASH ARRAY EXAMPLE 1:



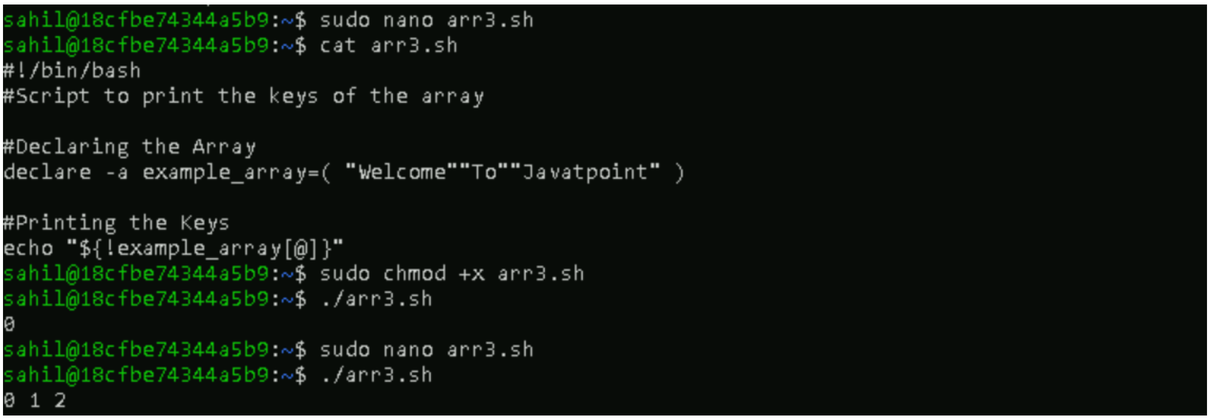
```
sahil@18cfbe74344a5b9:~$ sudo nano arr1.sh
sahil@18cfbe74344a5b9:~$ cat arr1.sh
#!/bin/bash
#Script to print an element of an array with an index of 2

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

#printing the element with index of 2
echo ${example_array[2]}
sahil@18cfbe74344a5b9:~$ sudo chmod +x arr1.sh
sahil@18cfbe74344a5b9:~$ ./arr1.sh

sahil@18cfbe74344a5b9:~$ sudo nano arr1.sh
sahil@18cfbe74344a5b9:~$ ./arr1.sh
Javatpoint
```

BASH ARRAY EXAMPLE 2:



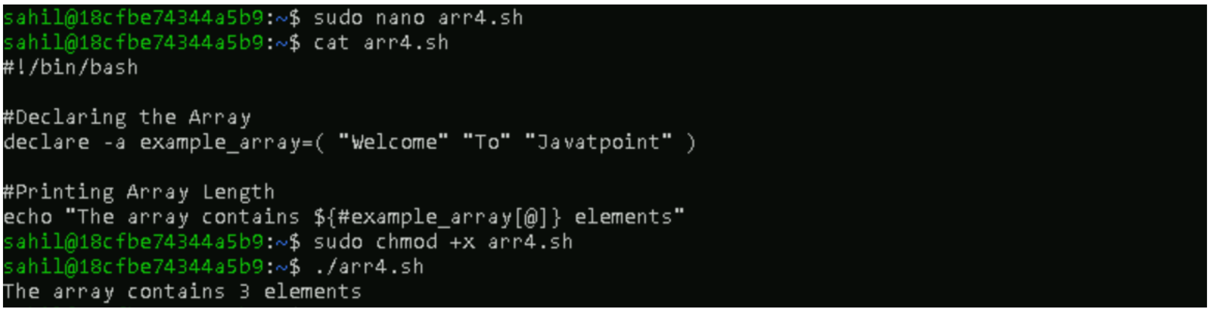
```
sahil@18cfbe74344a5b9:~$ sudo nano arr2.sh
sahil@18cfbe74344a5b9:~$ sudo chmod +x arr2.sh
sahil@18cfbe74344a5b9:~$ cat arr2.sh
#!/bin/bash
#Script to print all the elements of the array

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

#Printing all the elements
echo "${example_array[@]}"
sahil@18cfbe74344a5b9:~$ ./arr2.sh
Welcome To Javatpoint
```

BASH ARRAY EXAMPLE 3:



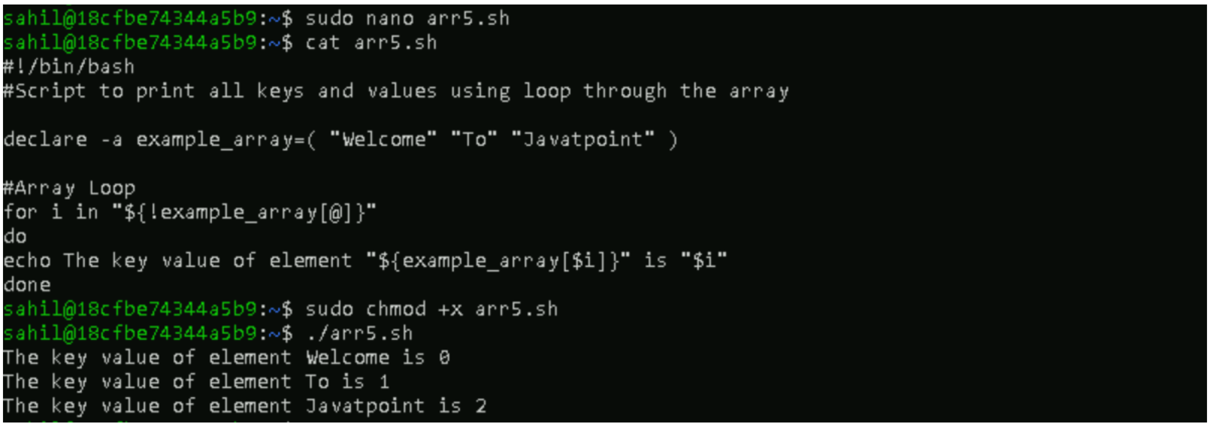
```
sahil@18cfbe74344a5b9:~$ sudo nano arr3.sh
sahil@18cfbe74344a5b9:~$ cat arr3.sh
#!/bin/bash
#Script to print the keys of the array

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

#Printing the Keys
echo "${!example_array[@]}"
sahil@18cfbe74344a5b9:~$ sudo chmod +x arr3.sh
sahil@18cfbe74344a5b9:~$ ./arr3.sh
0
sahil@18cfbe74344a5b9:~$ sudo nano arr3.sh
sahil@18cfbe74344a5b9:~$ ./arr3.sh
0 1 2
```

BASH ARRAY EXAMPLE 4:



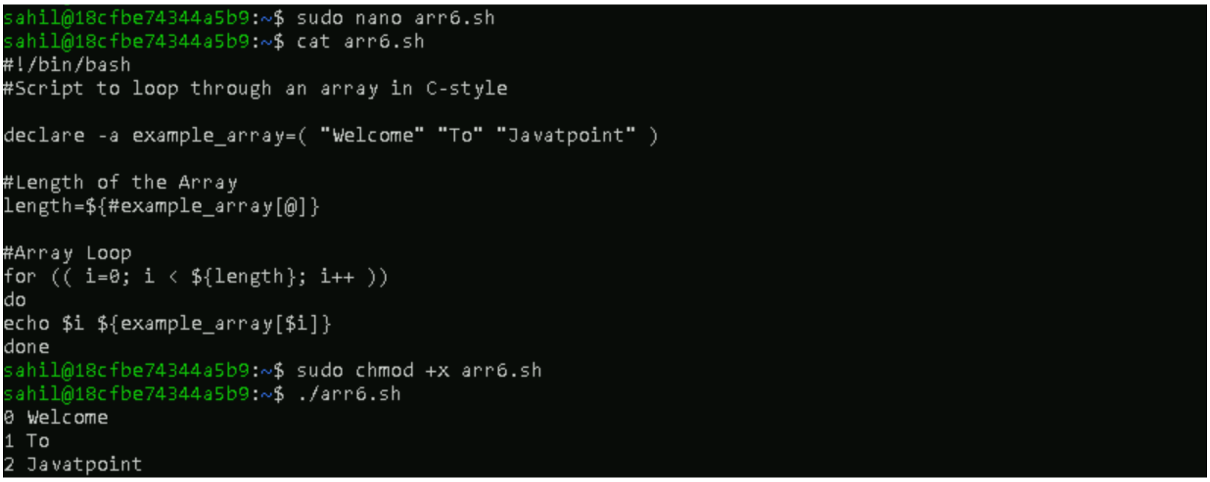
```
sahil@18cfbe74344a5b9:~$ sudo nano arr4.sh
sahil@18cfbe74344a5b9:~$ cat arr4.sh
#!/bin/bash

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

#Printing Array Length
echo "The array contains ${#example_array[@]} elements"
sahil@18cfbe74344a5b9:~$ sudo chmod +x arr4.sh
sahil@18cfbe74344a5b9:~$ ./arr4.sh
The array contains 3 elements
```

BASH ARRAY EXAMPLE 5:



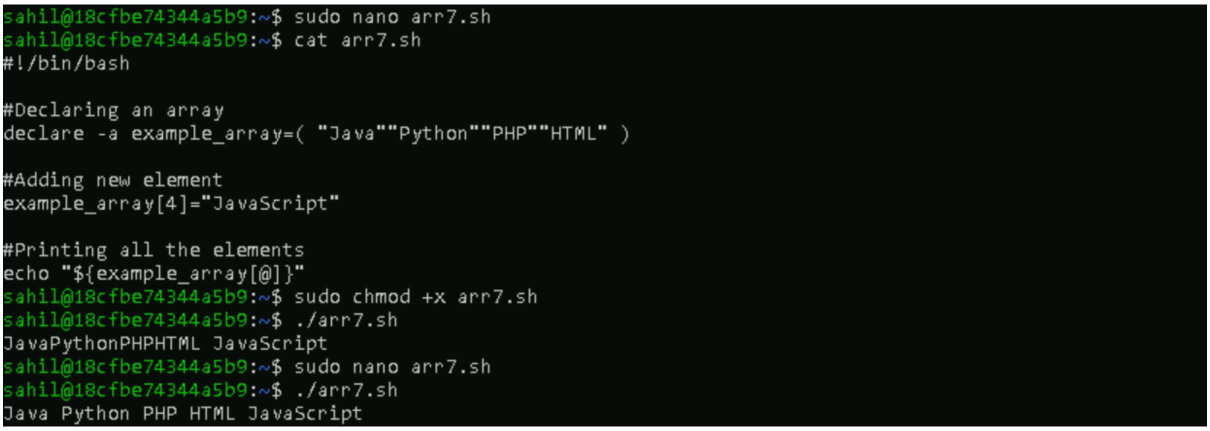
```
sahil@18cfbe74344a5b9:~$ sudo nano arr5.sh
sahil@18cfbe74344a5b9:~$ cat arr5.sh
#!/bin/bash
#Script to print all keys and values using loop through the array

declare -a example_array=( "Welcome" "To" "Javatpoint" )

#Array Loop
for i in "${!example_array[@]}"
do
echo The key value of element "${example_array[$i]}" is "$i"
done
sahil@18cfbe74344a5b9:~$ sudo chmod +x arr5.sh
sahil@18cfbe74344a5b9:~$ ./arr5.sh
The key value of element Welcome is 0
The key value of element To is 1
The key value of element Javatpoint is 2
```

BASH ARRAY EXAMPLE 6:



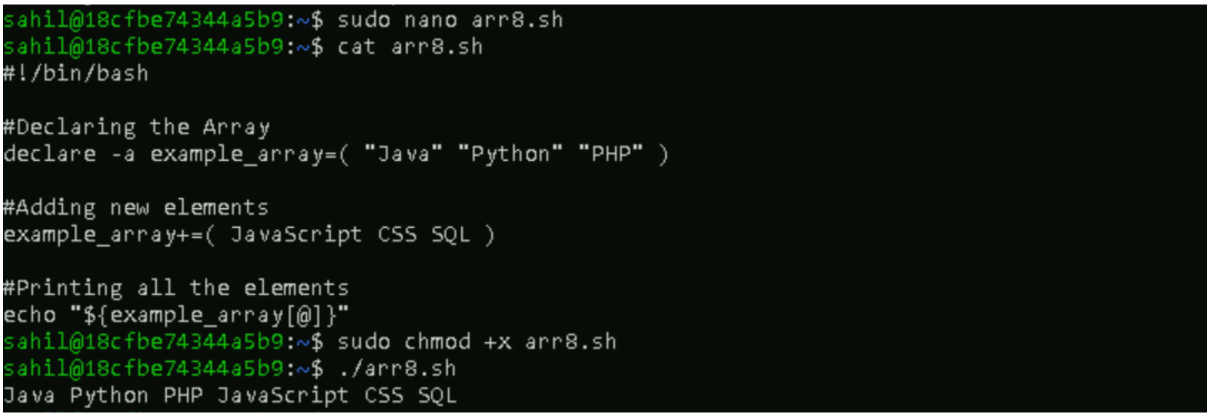
```
sahil@18cfbe74344a5b9:~$ sudo nano arr6.sh
sahil@18cfbe74344a5b9:~$ cat arr6.sh
#!/bin/bash
#Script to loop through an array in C-style

declare -a example_array=( "Welcome" "To" "Javatpoint" )

#Length of the Array
length=${#example_array[@]}

#Array Loop
for (( i=0; i < ${length}; i++ ))
do
echo $i ${example_array[$i]}
done
sahil@18cfbe74344a5b9:~$ sudo chmod +x arr6.sh
sahil@18cfbe74344a5b9:~$ ./arr6.sh
0 Welcome
1 To
2 Javatpoint
```

BASH ARRAY EXAMPLE 7:



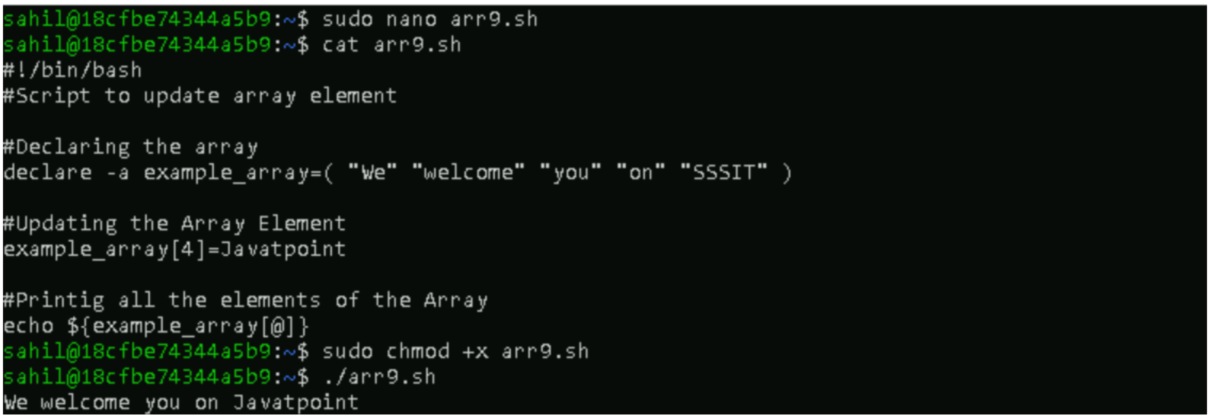
```
sahil@18cfbe74344a5b9:~$ sudo nano arr7.sh
sahil@18cfbe74344a5b9:~$ cat arr7.sh
#!/bin/bash

#Declaring an array
declare -a example_array=( "Java""Python""PHP""HTML" )

#Adding new element
example_array[4]="JavaScript"

#Printing all the elements
echo "${example_array[@]}"
sahil@18cfbe74344a5b9:~$ sudo chmod +x arr7.sh
sahil@18cfbe74344a5b9:~$ ./arr7.sh
JavaPythonPHPHTML JavaScript
sahil@18cfbe74344a5b9:~$ sudo nano arr7.sh
sahil@18cfbe74344a5b9:~$ ./arr7.sh
Java Python PHP HTML JavaScript
```

BASH ARRAY EXAMPLE 8:



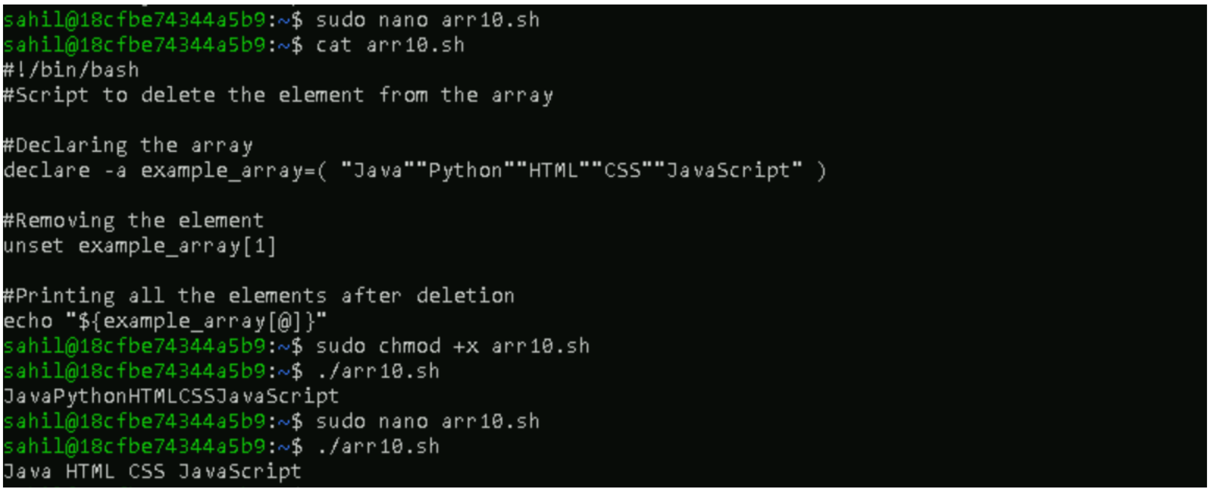
```
sahil@18cfbe74344a5b9:~$ sudo nano arr8.sh
sahil@18cfbe74344a5b9:~$ cat arr8.sh
#!/bin/bash

#Declaring the Array
declare -a example_array=( "Java" "Python" "PHP" )

#Adding new elements
example_array+=( JavaScript CSS SQL )

#Printing all the elements
echo "${example_array[@]}"
sahil@18cfbe74344a5b9:~$ sudo chmod +x arr8.sh
sahil@18cfbe74344a5b9:~$ ./arr8.sh
Java Python PHP JavaScript CSS SQL
```

BASH ARRAY EXAMPLE 9:



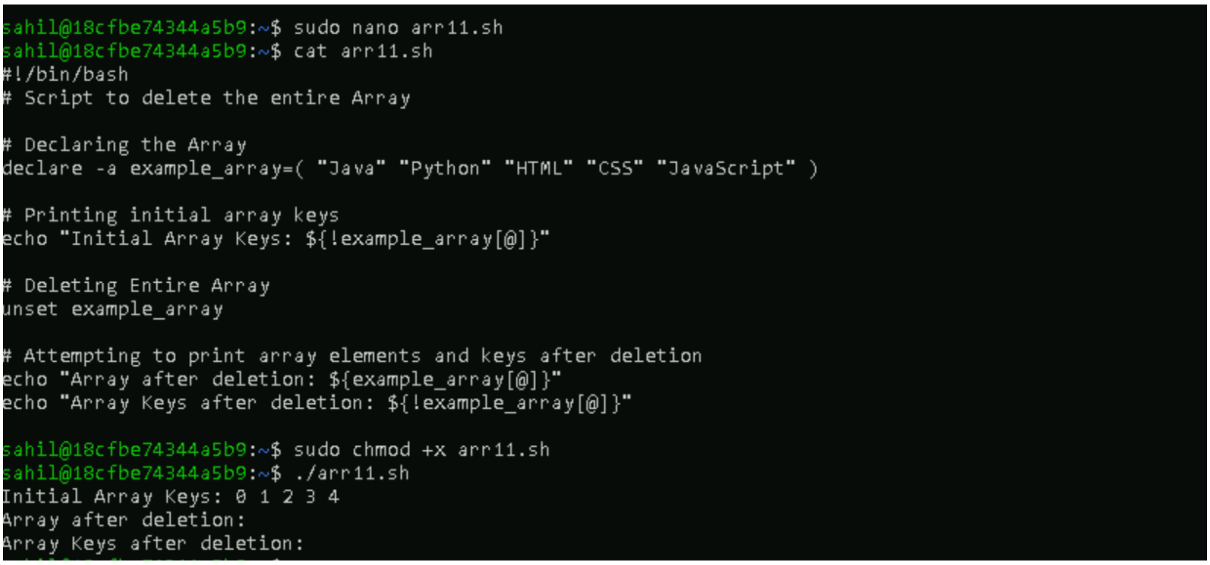
```
sahil@18cfbe74344a5b9:~$ sudo nano arr9.sh
sahil@18cfbe74344a5b9:~$ cat arr9.sh
#!/bin/bash
#Script to update array element

#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 ${example_array[@]}
sahil@18cfbe74344a5b9:~$ sudo chmod +x arr9.sh
sahil@18cfbe74344a5b9:~$ ./arr9.sh
We welcome you on Javatpoint
```

BASH ARRAY EXAMPLE 10:



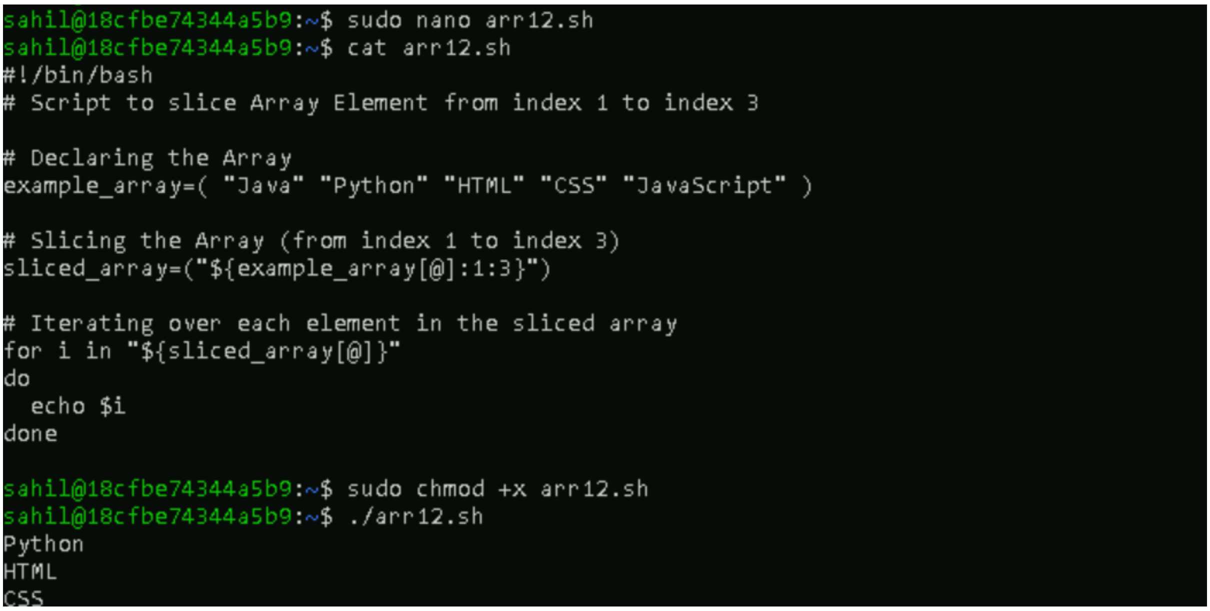
```
sahil@18cfbe74344a5b9:~$ sudo nano arr10.sh
sahil@18cfbe74344a5b9:~$ cat arr10.sh
#!/bin/bash
#Script to delete the element from the array

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

#Removing the element
unset example_array[1]

#Printing all the elements after deletion
echo "${example_array[@]}"
sahil@18cfbe74344a5b9:~$ sudo chmod +x arr10.sh
sahil@18cfbe74344a5b9:~$ ./arr10.sh
JavaPythonHTMLCSSJavaScript
sahil@18cfbe74344a5b9:~$ sudo nano arr10.sh
sahil@18cfbe74344a5b9:~$ ./arr10.sh
Java HTML CSS JavaScript
```

BASH ARRAY EXAMPLE 11:



```
sahil@18cfbe74344a5b9:~$ sudo nano arr11.sh
sahil@18cfbe74344a5b9:~$ cat arr11.sh
#!/bin/bash
# Script to delete the entire Array

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

# Printing initial array keys
echo "Initial Array Keys: ${!example_array[@]}"

# Deleting Entire Array
unset example_array

# Attempting to print array elements and keys after deletion
echo "Array after deletion: ${example_array[@]}"
echo "Array Keys after deletion: ${!example_array[@]}"

sahil@18cfbe74344a5b9:~$ sudo chmod +x arr11.sh
sahil@18cfbe74344a5b9:~$ ./arr11.sh
Initial Array Keys: 0 1 2 3 4
Array after deletion:
Array Keys after deletion:
```

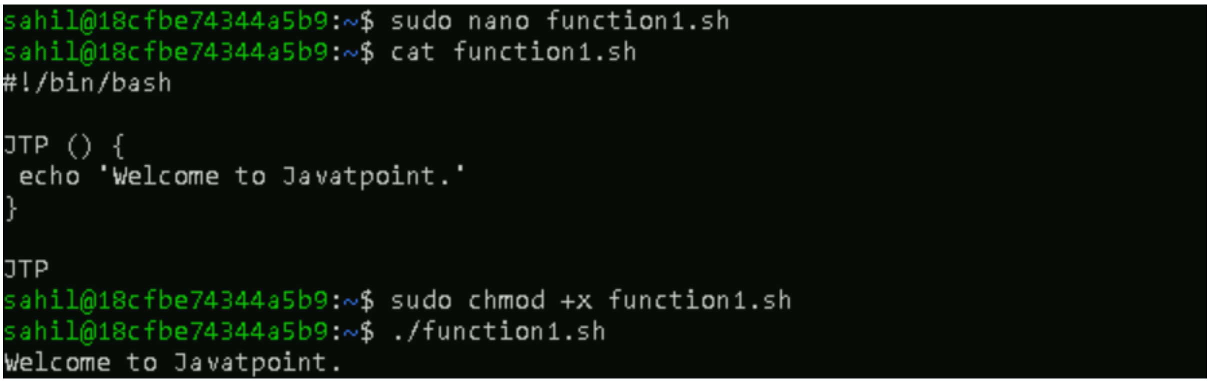
BASH ARRAY EXAMPLE 12:



```
sahil@18cfbe74344a5b9:~$ sudo nano arr12.sh
sahil@18cfbe74344a5b9:~$ cat arr12.sh
#!/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 (from index 1 to index 3)
sliced_array=("${example_array[@]:1:3}")

# Iterating over each element in the sliced array
for i in "${sliced_array[@]}"
do
  echo $i
done

sahil@18cfbe74344a5b9:~$ sudo chmod +x arr12.sh
sahil@18cfbe74344a5b9:~$ ./arr12.sh
Python
HTML
CSS
```

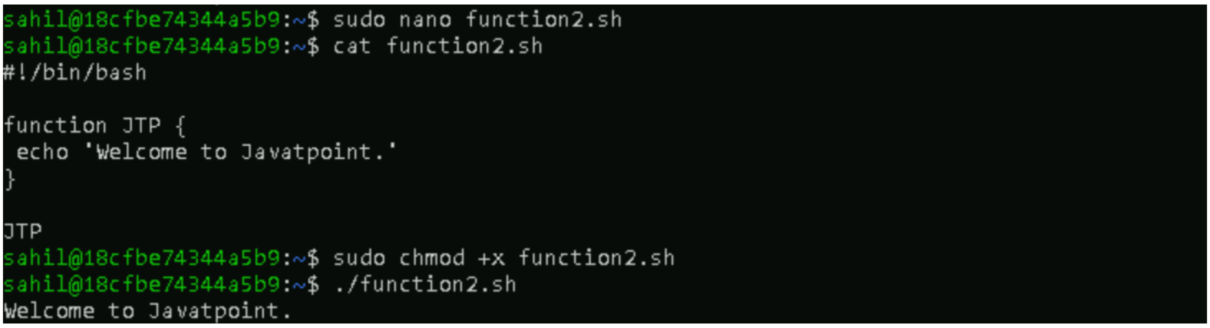
BASH FUNCTIONS:

BASH FUNCTIONS EXAMPLE 1:



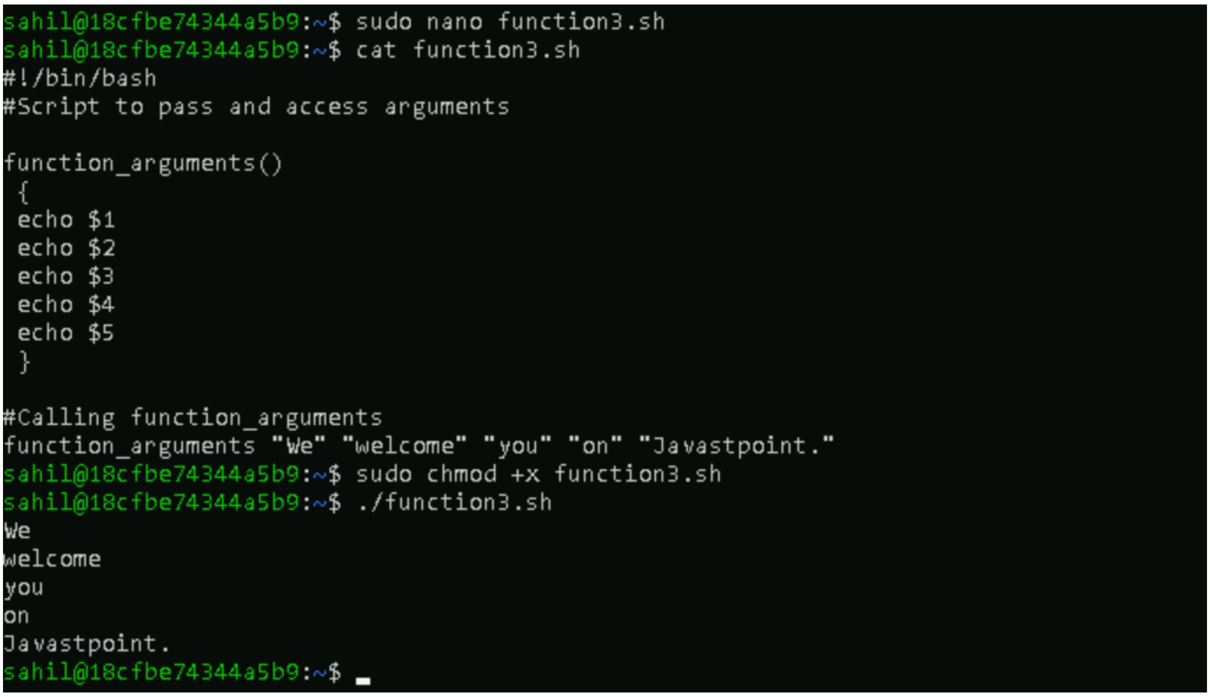
```
sahil@18cfbe74344a5b9:~$ sudo nano function1.sh
sahil@18cfbe74344a5b9:~$ cat function1.sh
#!/bin/bash

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

JTP
sahil@18cfbe74344a5b9:~$ sudo chmod +x function1.sh
sahil@18cfbe74344a5b9:~$ ./function1.sh
Welcome to Javatpoint.
```

BASH FUNCTIONS EXAMPLE 2:



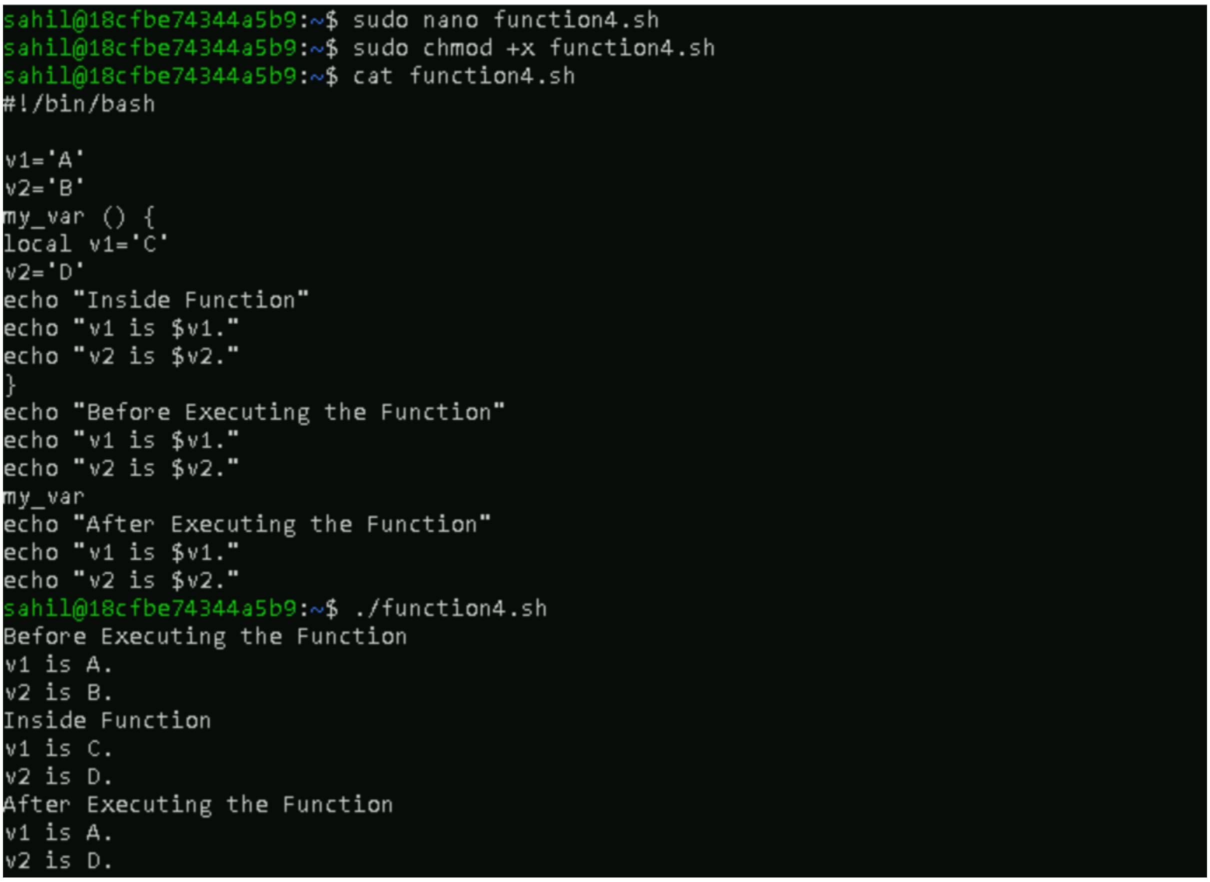
```
sahil@18cfbe74344a5b9:~$ sudo nano function2.sh
sahil@18cfbe74344a5b9:~$ cat function2.sh
#!/bin/bash

function JTP {
 echo 'Welcome to Javatpoint.'
}

JTP
sahil@18cfbe74344a5b9:~$ sudo chmod +x function2.sh
sahil@18cfbe74344a5b9:~$ ./function2.sh
Welcome to Javatpoint.
```

BASH FUNCTIONS EXAMPLE 3:



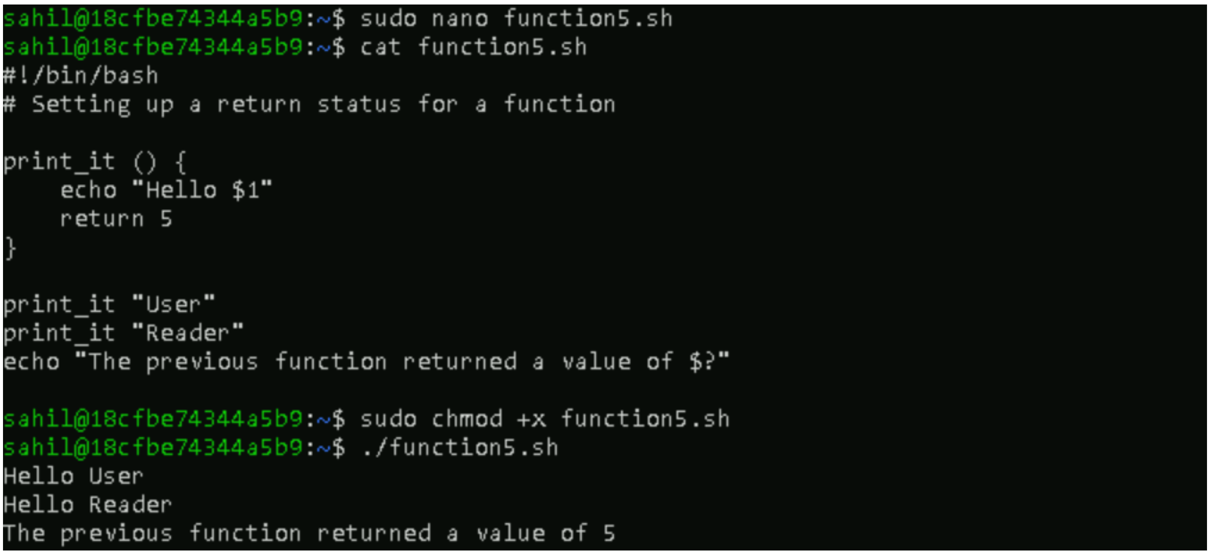
```
sahil@18cfbe74344a5b9:~$ sudo nano function3.sh
sahil@18cfbe74344a5b9:~$ cat function3.sh
#!/bin/bash
#Script to pass and access arguments

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

#Calling function_arguments
function_arguments "We" "welcome" "you" "on" "Javastpoint."
sahil@18cfbe74344a5b9:~$ sudo chmod +x function3.sh
sahil@18cfbe74344a5b9:~$ ./function3.sh
We
welcome
you
on
Javastpoint.
sahil@18cfbe74344a5b9:~$ _
```

BASH FUNCTIONS EXAMPLE 4:



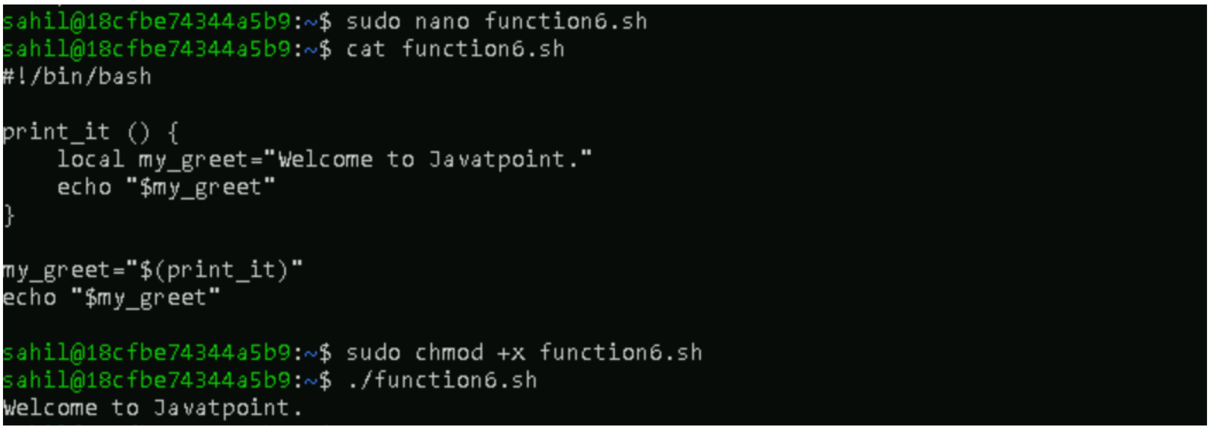
```
sahil@18cfbe74344a5b9:~$ sudo nano function4.sh
sahil@18cfbe74344a5b9:~$ sudo chmod +x function4.sh
sahil@18cfbe74344a5b9:~$ cat function4.sh
#!/bin/bash

v1='A'
v2='B'
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 $v2."
sahil@18cfbe74344a5b9:~$ ./function4.sh
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.
```

BASH FUNCTIONS EXAMPLE 5:



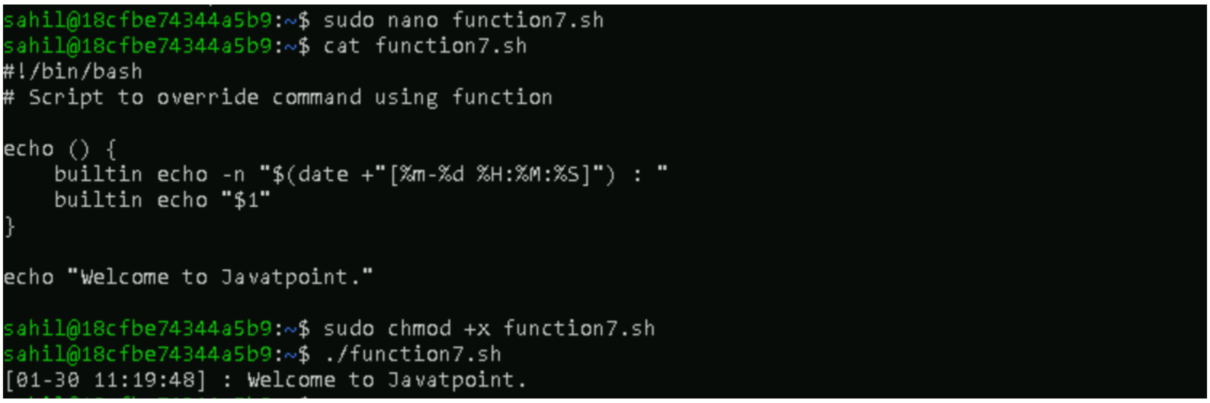
```
sahil@18cfbe74344a5b9:~$ sudo nano function5.sh
sahil@18cfbe74344a5b9:~$ cat function5.sh
#!/bin/bash
# Setting up a return status for a function

print_it () {
    echo "Hello $1"
    return 5
}

print_it "User"
print_it "Reader"
echo "The previous function returned a value of $?"

sahil@18cfbe74344a5b9:~$ sudo chmod +x function5.sh
sahil@18cfbe74344a5b9:~$ ./function5.sh
Hello User
Hello Reader
The previous function returned a value of 5
```

BASH FUNCTIONS EXAMPLE 6:



```
sahil@18cfbe74344a5b9:~$ sudo nano function6.sh
sahil@18cfbe74344a5b9:~$ cat function6.sh
#!/bin/bash

print_it () {
    local my_greet="Welcome to Javatpoint."
    echo "$my_greet"
}

my_greet="$(print_it)"
echo "$my_greet"

sahil@18cfbe74344a5b9:~$ sudo chmod +x function6.sh
sahil@18cfbe74344a5b9:~$ ./function6.sh
Welcome to Javatpoint.
```

BASH FUNCTIONS EXAMPLE 7:



```
sahil@18cfbe74344a5b9:~$ sudo nano function7.sh
sahil@18cfbe74344a5b9:~$ cat function7.sh
#!/bin/bash
# Script to override command using function

echo () {
    builtin echo -n "$(date +"[%m-%d %H:%M:%S]") : "
    builtin echo "$1"
}

echo "Welcome to Javatpoint."

sahil@18cfbe74344a5b9:~$ sudo chmod +x function7.sh
sahil@18cfbe74344a5b9:~$ ./function7.sh
[01-30 11:19:48] : Welcome to Javatpoint.
```