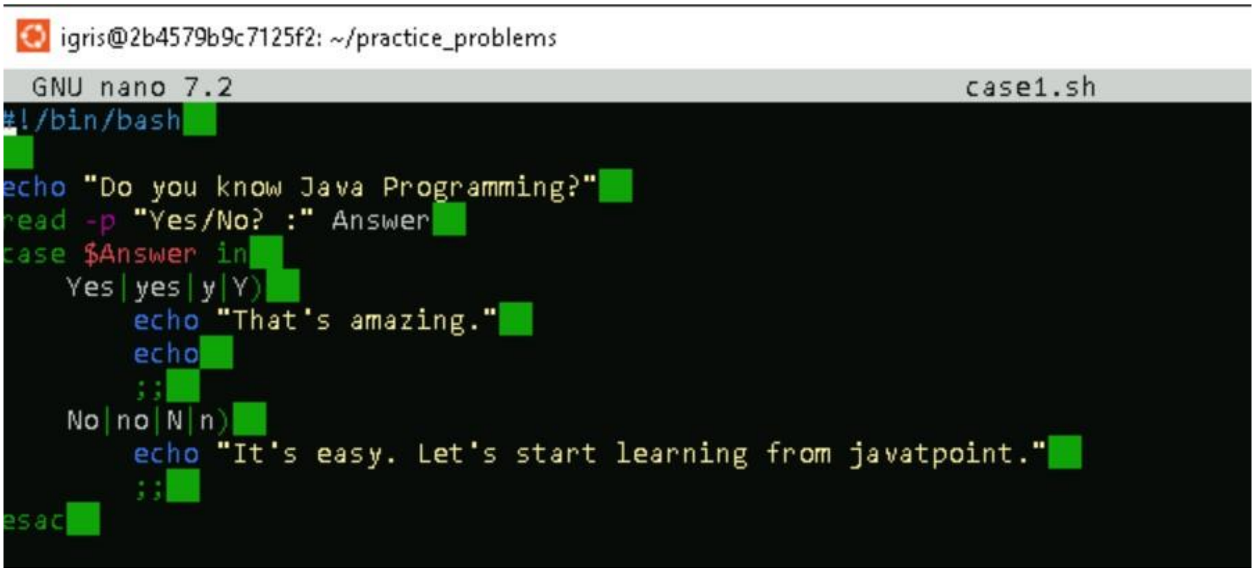
Pratheek U B
289226

Bash Project

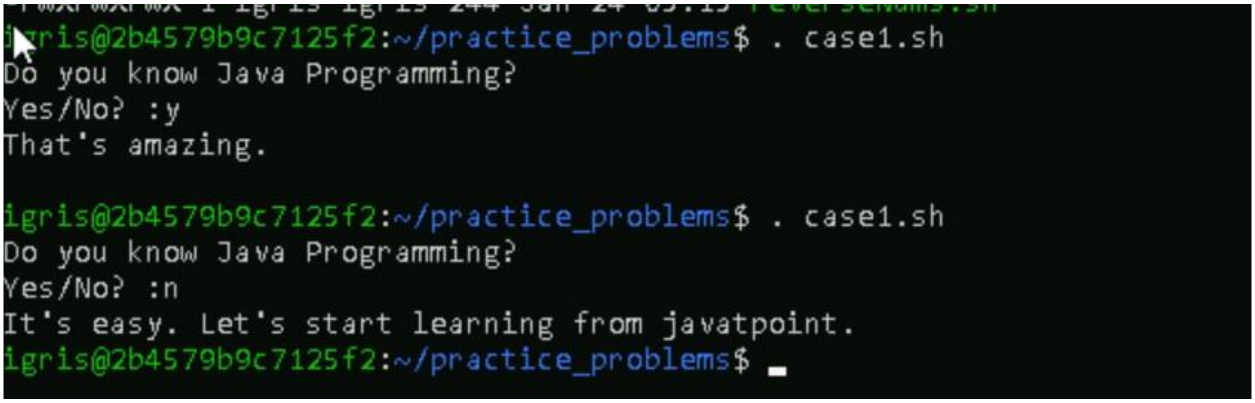
Bash CASE

Case script 1:



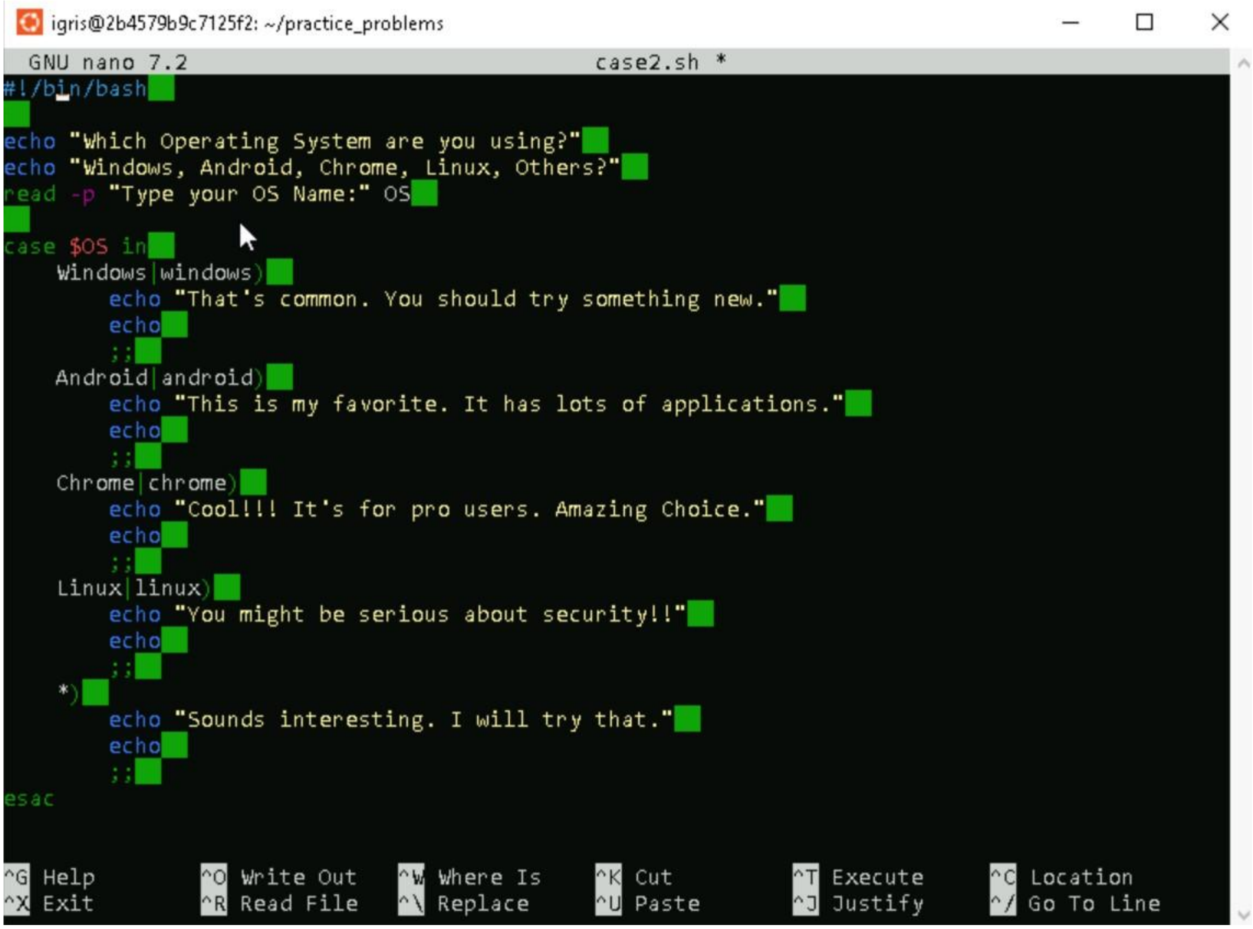
igris@2b4579b9c7125f2: ~/practice_problems

```
  GNU nano 7.2                                                      case1.sh
#!/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
```

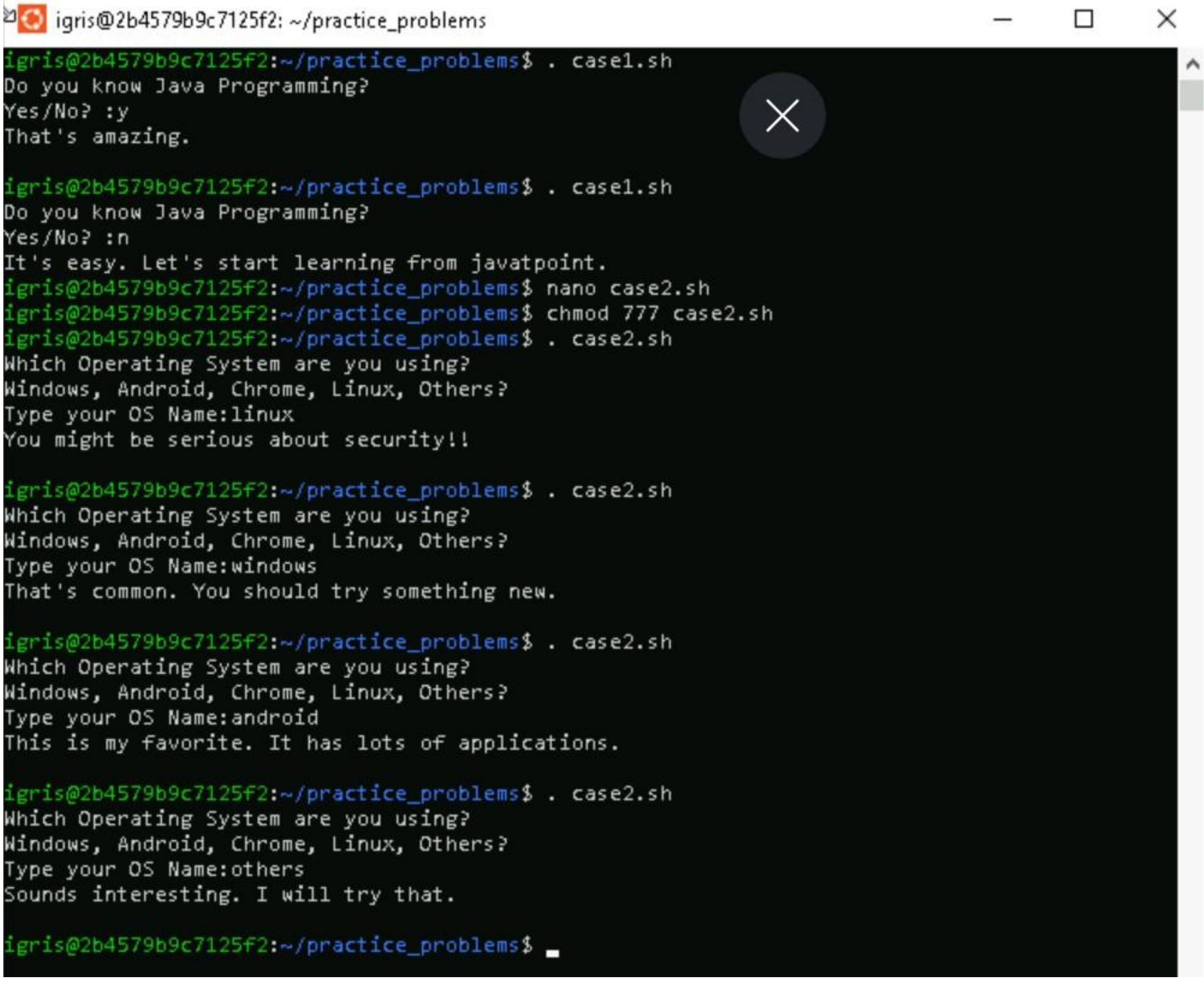


```
-rwxrwxrwx 1 igris igris 244 Jan 24 05:15 reverseNums.sh
igris@2b4579b9c7125f2:~/practice_problems$ . case1.sh
Do you know Java Programming?
Yes/No? :y
That's amazing.

igris@2b4579b9c7125f2:~/practice_problems$ . case1.sh
Do you know Java Programming?
Yes/No? :n
It's easy. Let's start learning from javatpoint.
igris@2b4579b9c7125f2:~/practice_problems$
```

Case script 2:



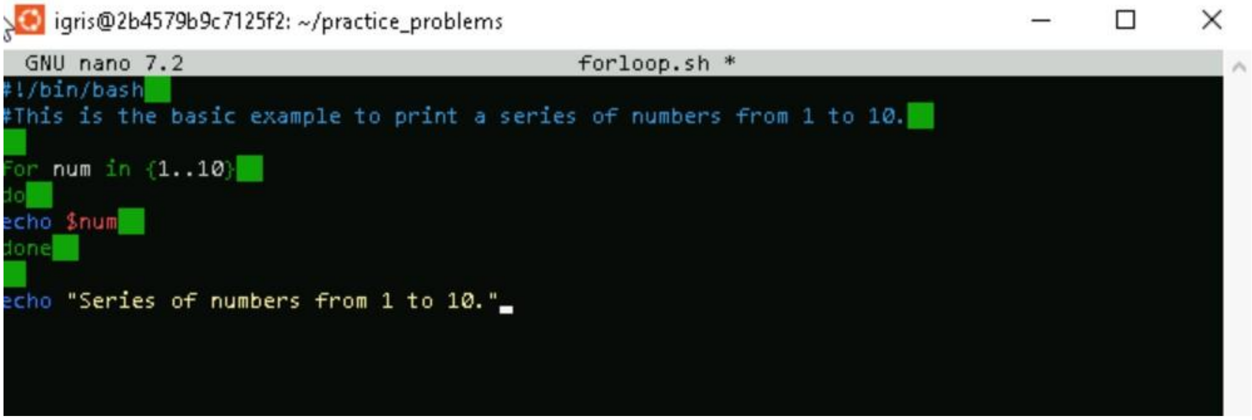
```
#!/bin/bash

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
```



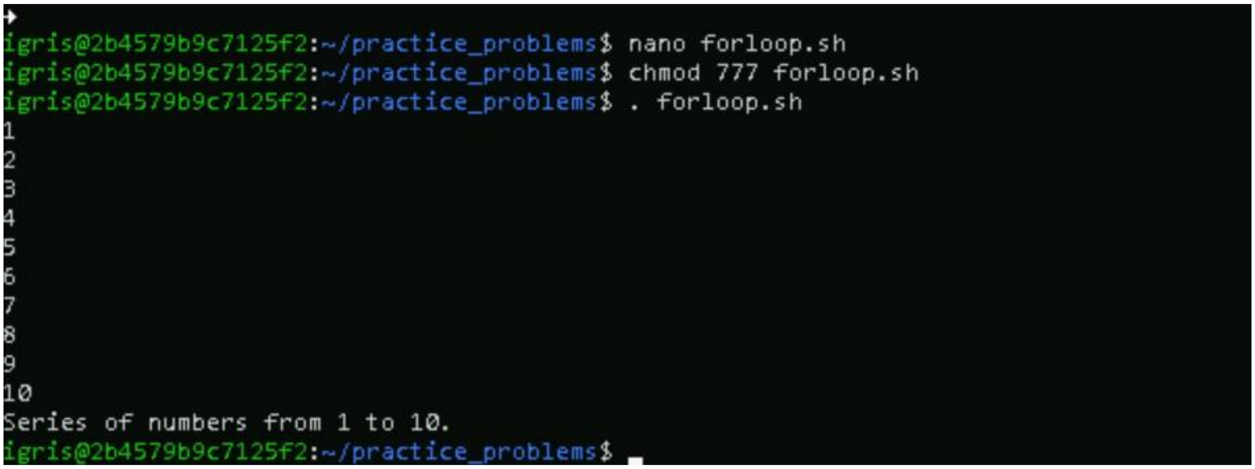
```
igris@2b4579b9c7125f2:~/practice_problems$ . case1.sh
Do you know Java Programming?
Yes/No? :y
That's amazing.

igris@2b4579b9c7125f2:~/practice_problems$ . case1.sh
Do you know Java Programming?
Yes/No? :n
It's easy. Let's start learning from javatpoint.
igris@2b4579b9c7125f2:~/practice_problems$ nano case2.sh
igris@2b4579b9c7125f2:~/practice_problems$ chmod 777 case2.sh
igris@2b4579b9c7125f2:~/practice_problems$ . case2.sh
Which Operating System are you using?
Windows, Android, Chrome, Linux, Others?
Type your OS Name:linux
You might be serious about security!!

igris@2b4579b9c7125f2:~/practice_problems$ . case2.sh
Which Operating System are you using?
Windows, Android, Chrome, Linux, Others?
Type your OS Name:windows
That's common. You should try something new.

igris@2b4579b9c7125f2:~/practice_problems$ . case2.sh
Which Operating System are you using?
Windows, Android, Chrome, Linux, Others?
Type your OS Name:android
This is my favorite. It has lots of applications.

igris@2b4579b9c7125f2:~/practice_problems$ . case2.sh
Which Operating System are you using?
Windows, Android, Chrome, Linux, Others?
Type your OS Name:others
Sounds interesting. I will try that.

igris@2b4579b9c7125f2:~/practice_problems$
```

Bash FOR LOOP

Script 1: For loop to read a range

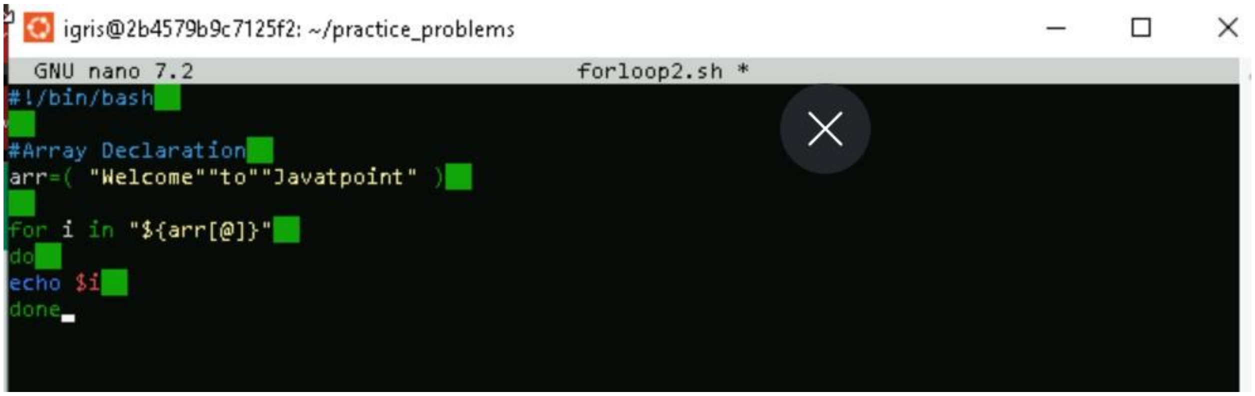


```
#!/bin/bash
#This is the basic example to print a series of numbers from 1 to 10.

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

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

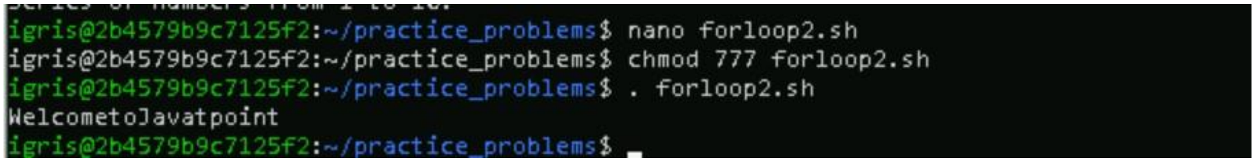


```
igris@2b4579b9c7125f2:~/practice_problems$ nano forloop.sh
igris@2b4579b9c7125f2:~/practice_problems$ chmod 777 forloop.sh
igris@2b4579b9c7125f2:~/practice_problems$ . forloop.sh
1
2
3
4
5
6
7
8
9
10
Series of numbers from 1 to 10.
igris@2b4579b9c7125f2:~/practice_problems$ _
```

Script 2: For loop to read array variables

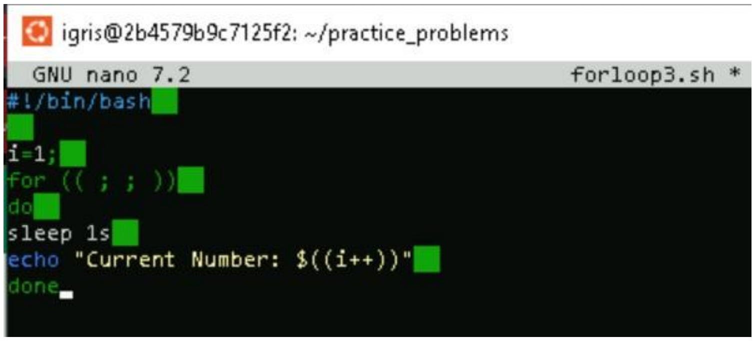


```
  GNU nano 7.2                        forloop2.sh *
#!/bin/bash

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

for i in "${arr[@]}"
do
echo $i
done
```



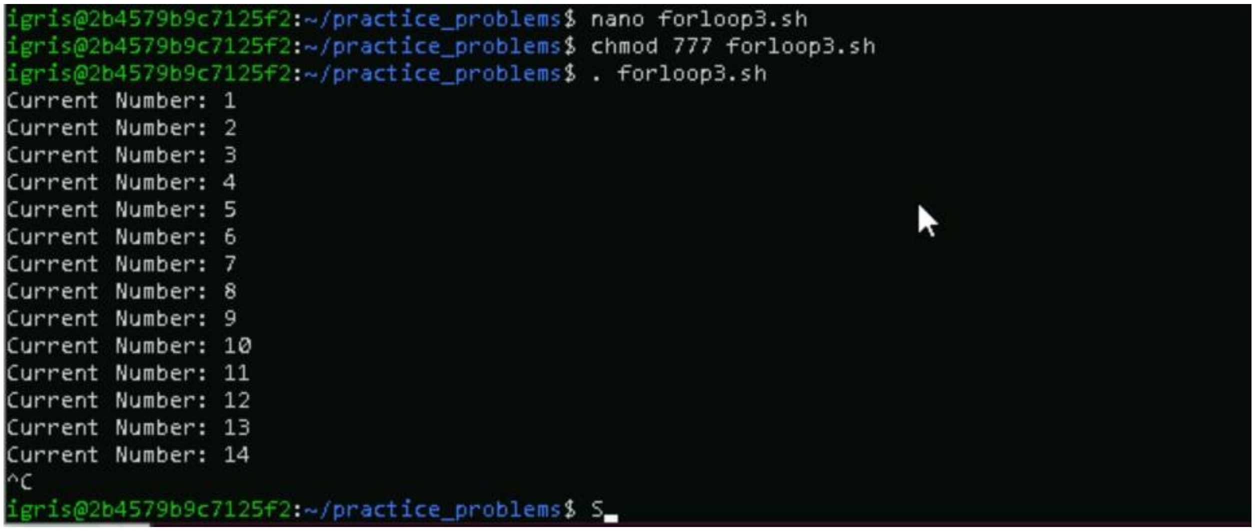
```
igris@2b4579b9c7125f2:~/practice_problems$ nano forloop2.sh
igris@2b4579b9c7125f2:~/practice_problems$ chmod 777 forloop2.sh
igris@2b4579b9c7125f2:~/practice_problems$ . forloop2.sh
WelcometoJavatpoint
igris@2b4579b9c7125f2:~/practice_problems$ _
```

Script 3: Infinite Bash For Loop



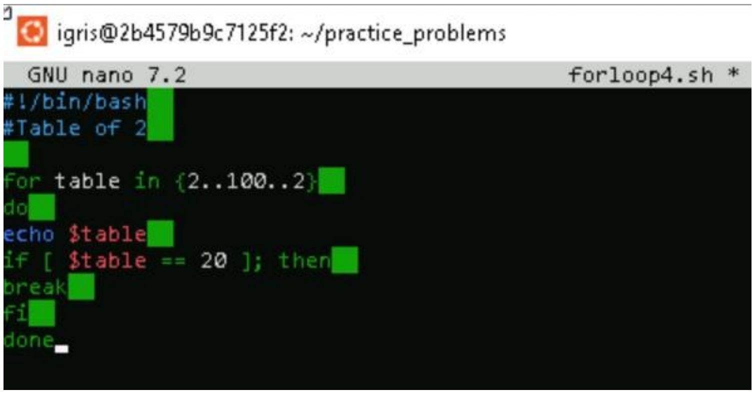
igris@2b4579b9c7125f2: ~/practice_problems

```
 GNU nano 7.2                                      forloop3.sh *
#!/bin/bash

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



```
igris@2b4579b9c7125f2:~/practice_problems$ nano forloop3.sh
igris@2b4579b9c7125f2:~/practice_problems$ chmod 777 forloop3.sh
igris@2b4579b9c7125f2:~/practice_problems$ . forloop3.sh
Current Number: 1
Current Number: 2
Current Number: 3
Current Number: 4
Current Number: 5
Current Number: 6
Current Number: 7
Current Number: 8
Current Number: 9
Current Number: 10
Current Number: 11
Current Number: 12
Current Number: 13
Current Number: 14
^C
igris@2b4579b9c7125f2:~/practice_problems$ S
```

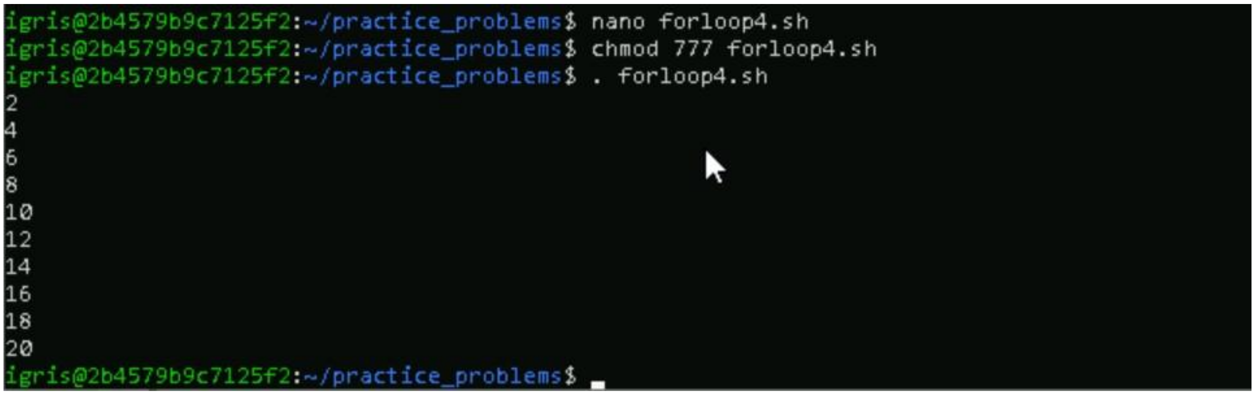
Script 4: For loop with break statement



igris@2b4579b9c7125f2: ~/practice_problems

```
 GNU nano 7.2                                      forloop4.sh *
#!/bin/bash
#Table of 2

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

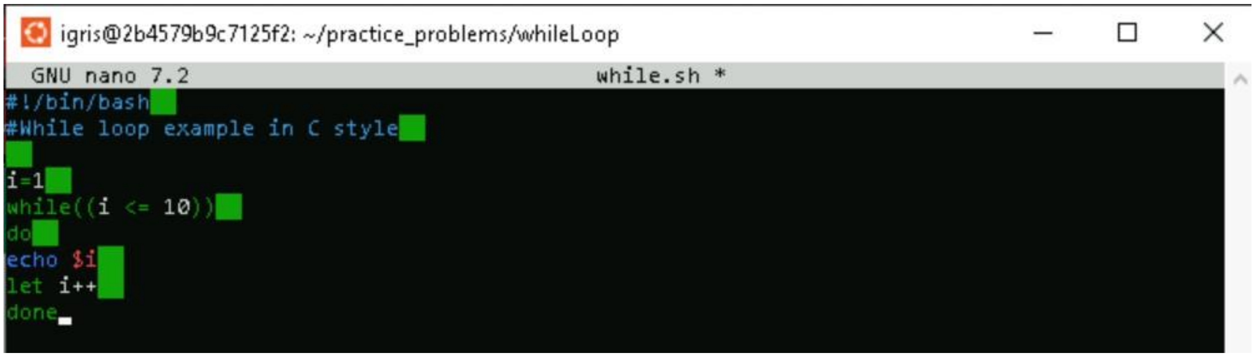
Pratheek U B
289226



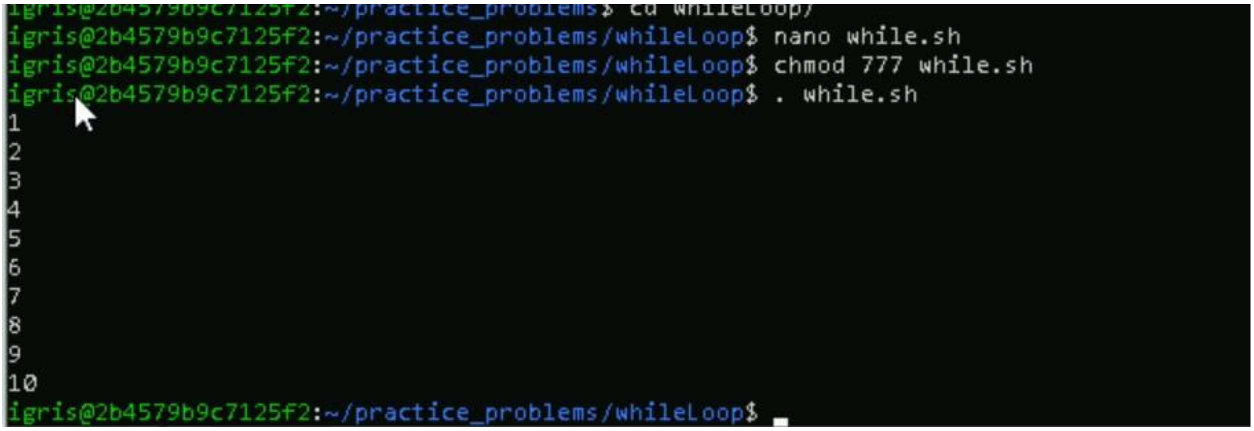
```
igris@2b4579b9c7125f2:~/practice_problems$ nano forloop4.sh
igris@2b4579b9c7125f2:~/practice_problems$ chmod 777 forloop4.sh
igris@2b4579b9c7125f2:~/practice_problems$ . forloop4.sh
2
4
6
8
10
12
14
16
18
20
igris@2b4579b9c7125f2:~/practice_problems$ _
```

Bash WHILE LOOP

Script 1: While loop with C-Style



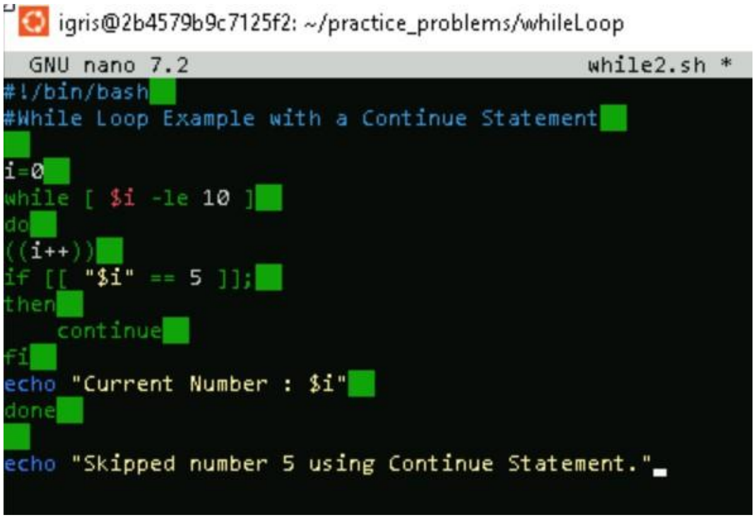
```
  GNU nano 7.2                           while.sh *
#!/bin/bash
#While loop example in C style

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



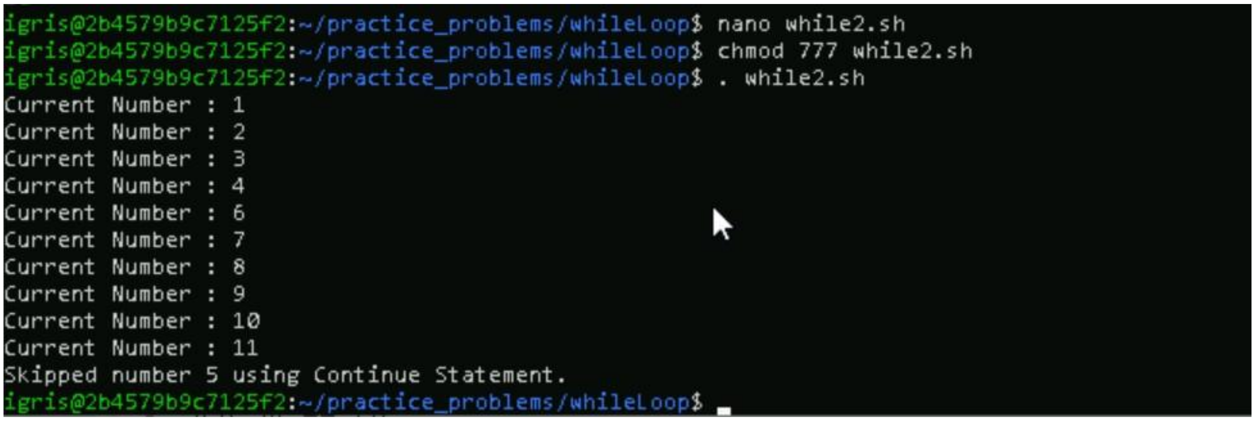
```
igris@2b4579b9c7125f2:~/practice_problems$ cd whileLoop/
igris@2b4579b9c7125f2:~/practice_problems/whileLoop$ nano while.sh
igris@2b4579b9c7125f2:~/practice_problems/whileLoop$ chmod 777 while.sh
igris@2b4579b9c7125f2:~/practice_problems/whileLoop$ . while.sh
1
2
3
4
5
6
7
8
9
10
igris@2b4579b9c7125f2:~/practice_problems/whileLoop$ _
```

Pratheek U B
289226

Script 2: While loop with continue statement

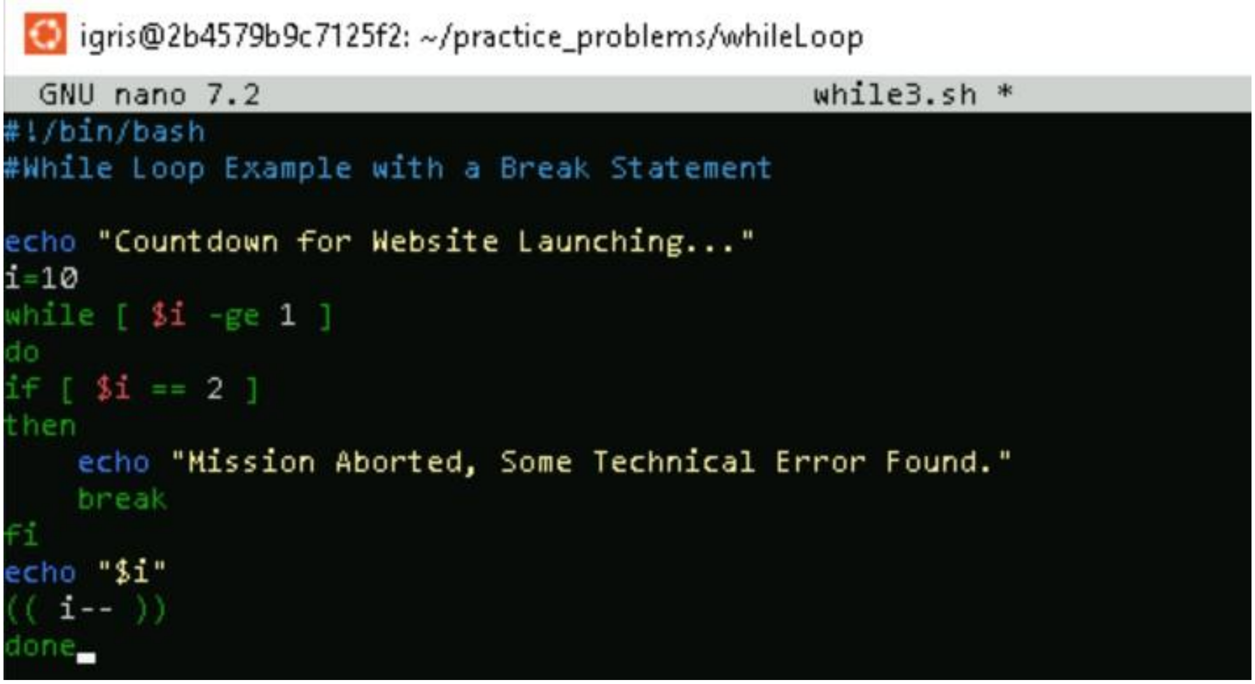


```
igris@2b4579b9c7125f2: ~/practice_problems/whileLoop
  GNU nano 7.2                                   while2.sh *
#!/bin/bash
#While Loop Example with a Continue Statement

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."
```

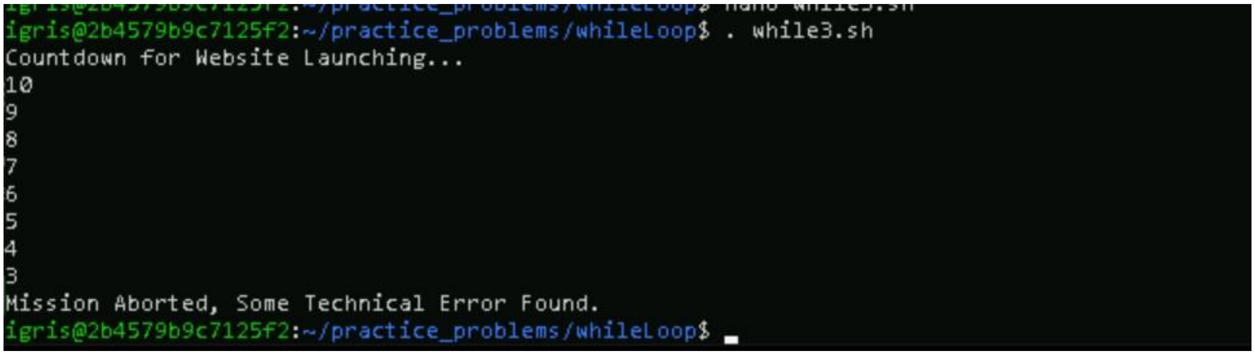


```
igris@2b4579b9c7125f2:~/practice_problems/whileLoop$ nano while2.sh
igris@2b4579b9c7125f2:~/practice_problems/whileLoop$ chmod 777 while2.sh
igris@2b4579b9c7125f2:~/practice_problems/whileLoop$ . while2.sh
Current Number : 1
Current Number : 2
Current Number : 3
Current Number : 4
Current Number : 6
Current Number : 7
Current Number : 8
Current Number : 9
Current Number : 10
Current Number : 11
Skipped number 5 using Continue Statement.
igris@2b4579b9c7125f2:~/practice_problems/whileLoop$
```

Script 3: While loop with break statement



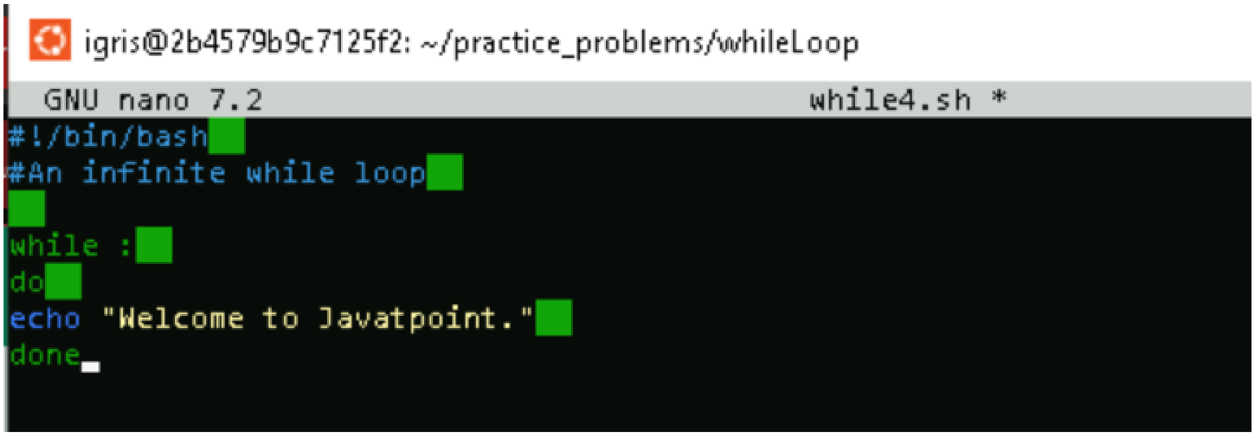
```
igris@2b4579b9c7125f2: ~/practice_problems/whileLoop
  GNU nano 7.2                                   while3.sh *
#!/bin/bash
#While Loop Example with a Break Statement

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
```



```
igris@2b4579b9c7125f2:~/practice_problems/whileLoop$ nano while3.sh
igris@2b4579b9c7125f2:~/practice_problems/whileLoop$ . while3.sh
Countdown for Website Launching...
10
9
8
7
6
5
4
3
Mission Aborted, Some Technical Error Found.
igris@2b4579b9c7125f2:~/practice_problems/whileLoop$
```

Pratheek U B
289226

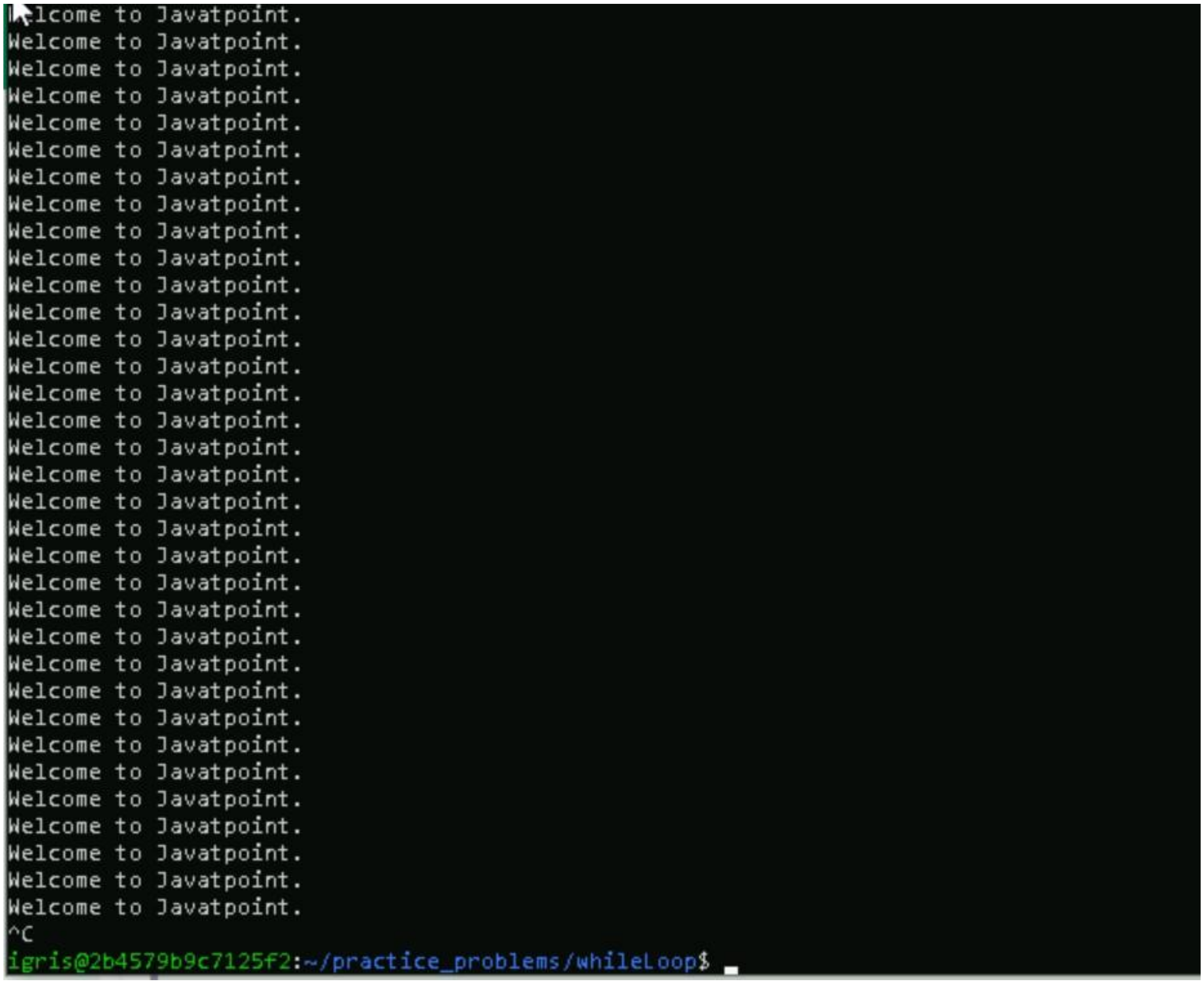
Script 4: Infinite while loop



igris@2b4579b9c7125f2: ~/practice_problems/whileLoop

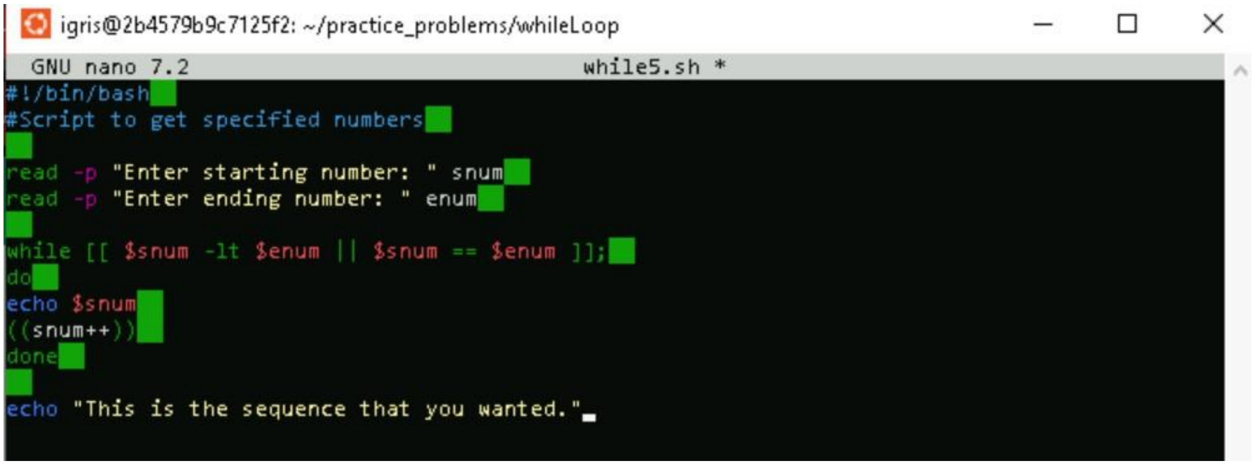
```
 GNU nano 7.2                              while4.sh *
#!/bin/bash
#An infinite while loop

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

Pratheek U B
289226



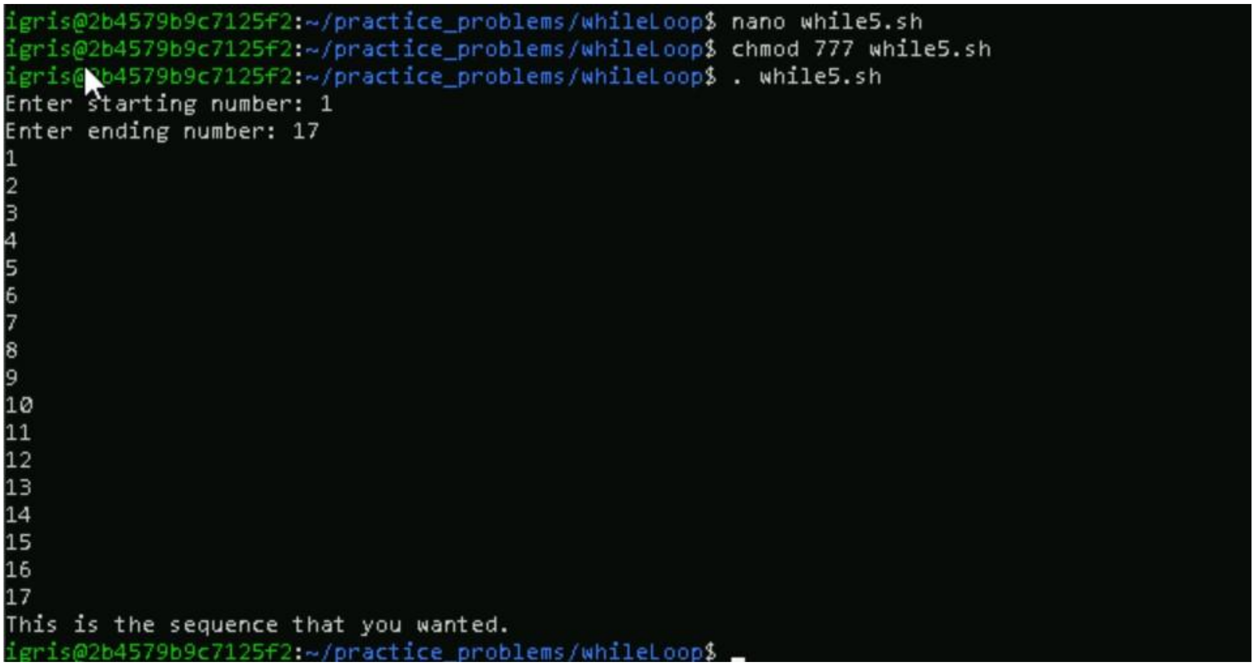
```
Welcome to Javatpoint.
Welcome to Javatpoint.
Welcome to Javatpoint.
Welcome to Javatpoint.
Welcome to Javatpoint.
Welcome to Javatpoint.
Welcome to Javatpoint.
Welcome to Javatpoint.
Welcome to Javatpoint.
Welcome to Javatpoint.
Welcome to Javatpoint.
Welcome to Javatpoint.
Welcome to Javatpoint.
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.
Welcome to Javatpoint.
Welcome to Javatpoint.
Welcome to Javatpoint.
Welcome to Javatpoint.
^C
igris@2b4579b9c7125f2:~/practice_problems/whileLoop$
```

Script 5: While loop with multiple conditions



```
  GNU nano 7.2                              while5.sh *
#!/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."
```

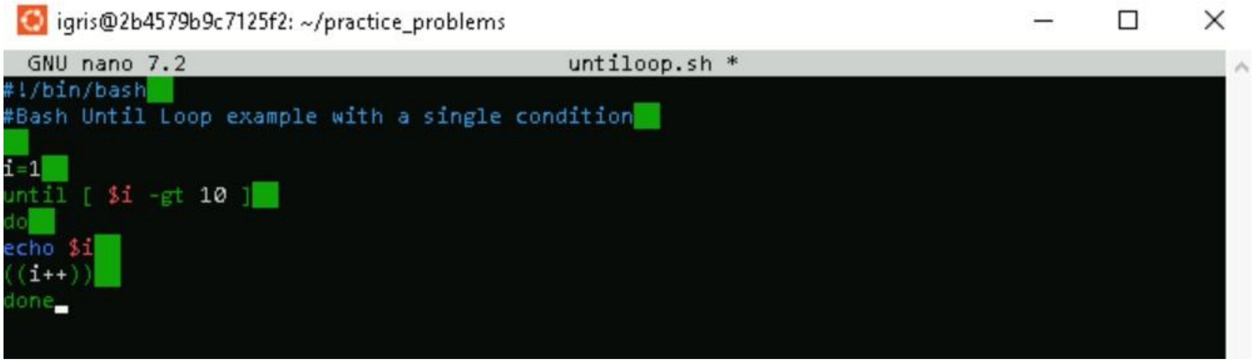
Pratheek U B
289226



```
igris@2b4579b9c7125f2:~/practice_problems/whileLoop$ nano while5.sh
igris@2b4579b9c7125f2:~/practice_problems/whileLoop$ chmod 777 while5.sh
igris@2b4579b9c7125f2:~/practice_problems/whileLoop$ . while5.sh
Enter starting number: 1
Enter ending number: 17
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
This is the sequence that you wanted.
igris@2b4579b9c7125f2:~/practice_problems/whileLoop$ _
```

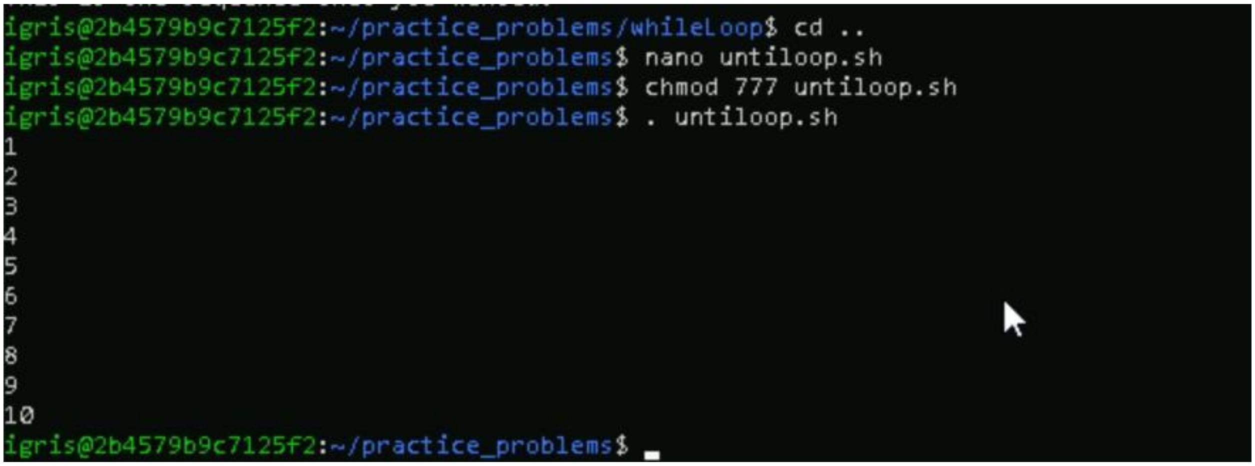
Bash UNTIL LOOP

Script 1: Until loop with single condition



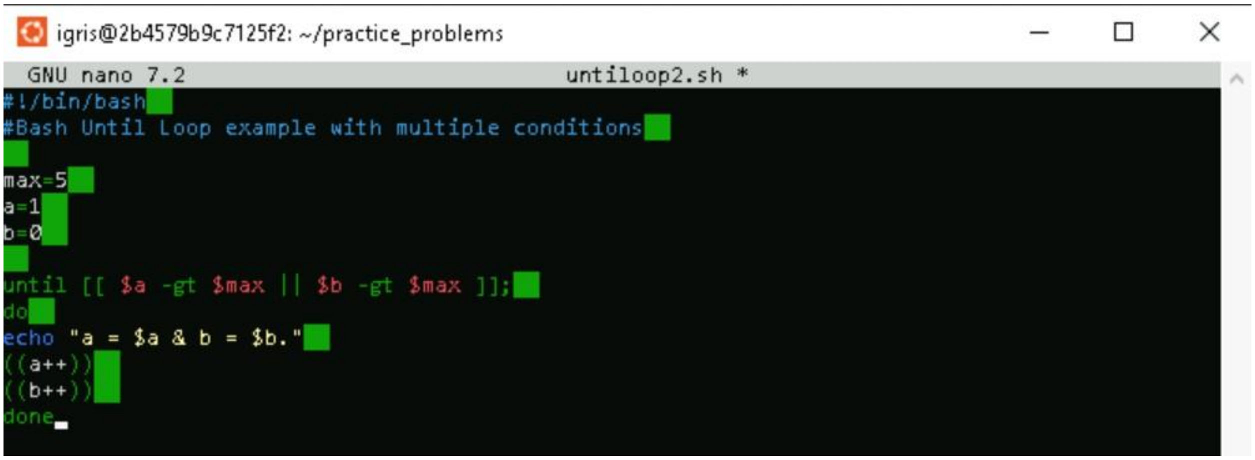
```
  GNU nano 7.2                          untiloop.sh *
#!/bin/bash
#Bash Until Loop example with a single condition

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

Pratheek U B
289226

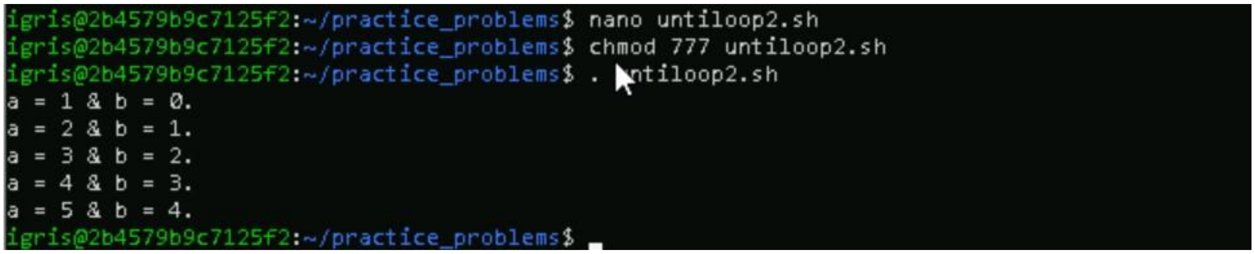


```
igris@2b4579b9c7125f2:~/practice_problems/whileLoop$ cd ..
igris@2b4579b9c7125f2:~/practice_problems$ nano untiloop.sh
igris@2b4579b9c7125f2:~/practice_problems$ chmod 777 untiloop.sh
igris@2b4579b9c7125f2:~/practice_problems$ . untiloop.sh
1
2
3
4
5
6
7
8
9
10
igris@2b4579b9c7125f2:~/practice_problems$ _
```

Script 2: Until loop with multiple statement



```
  GNU nano 7.2                          untiloop2.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
```

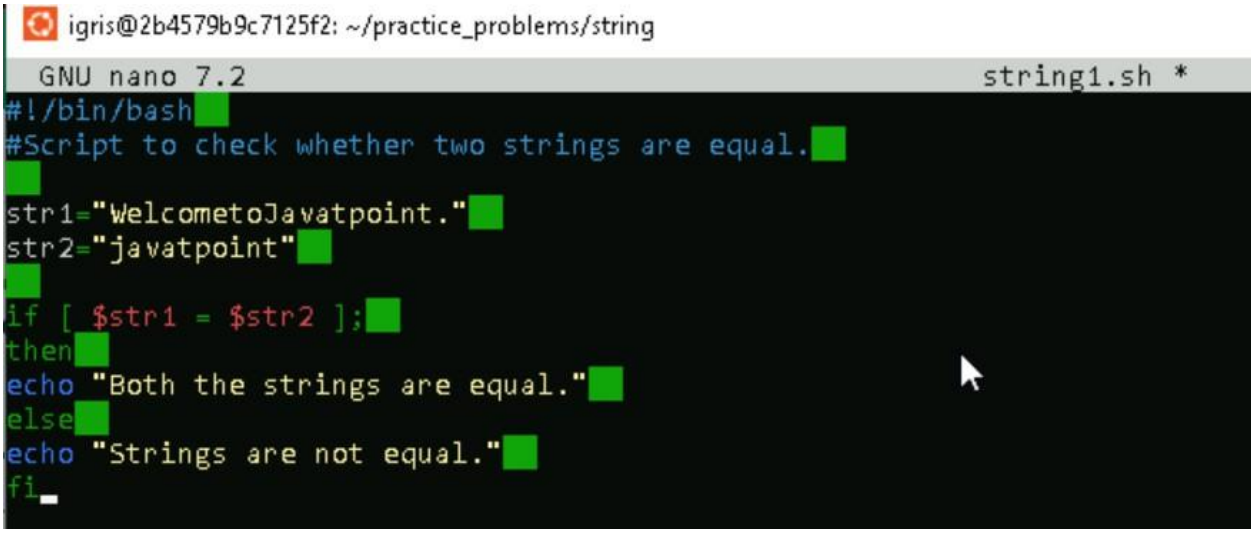


```
igris@2b4579b9c7125f2:~/practice_problems$ nano untiloop2.sh
igris@2b4579b9c7125f2:~/practice_problems$ chmod 777 untiloop2.sh
igris@2b4579b9c7125f2:~/practice_problems$ . untiloop2.sh
a = 1 & b = 0.
a = 2 & b = 1.
a = 3 & b = 2.
a = 4 & b = 3.
a = 5 & b = 4.
igris@2b4579b9c7125f2:~/practice_problems$ _
```

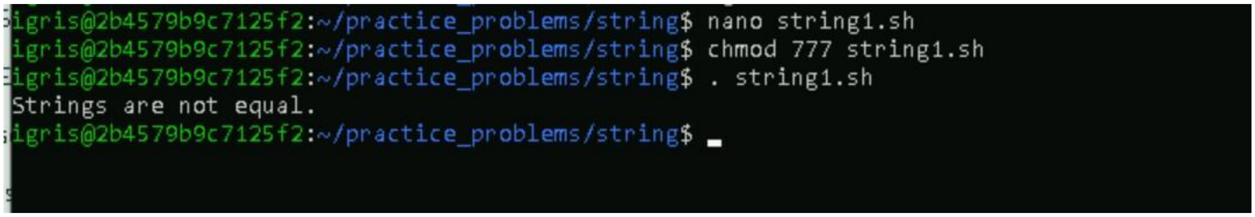
Pratheek U B
289226

Bash STRING

Script 1: Equal operator

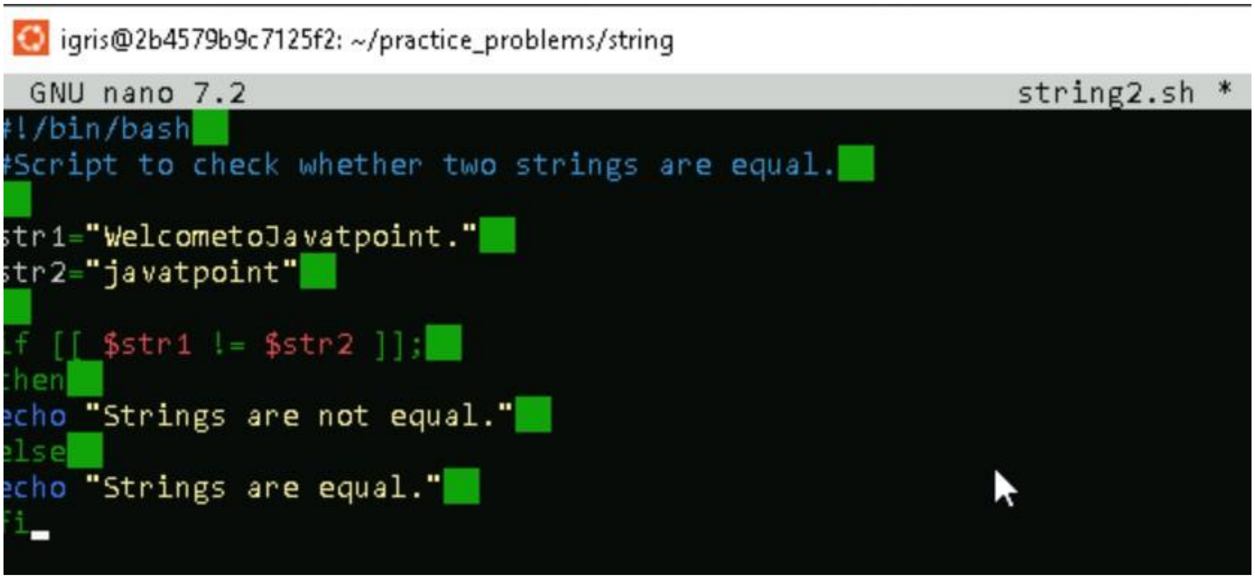


```
#!/bin/bash
#Script to check whether two strings are equal.

str1="WelcometoJavatpoint."
str2="javatpoint"

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

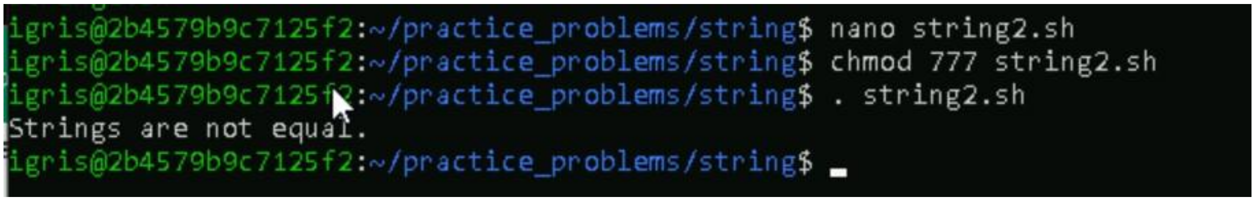


```
igris@2b4579b9c7125f2:~/practice_problems/string$ nano string1.sh
igris@2b4579b9c7125f2:~/practice_problems/string$ chmod 777 string1.sh
igris@2b4579b9c7125f2:~/practice_problems/string$ . string1.sh
Strings are not equal.
igris@2b4579b9c7125f2:~/practice_problems/string$
```

Script 2: Not equal operator

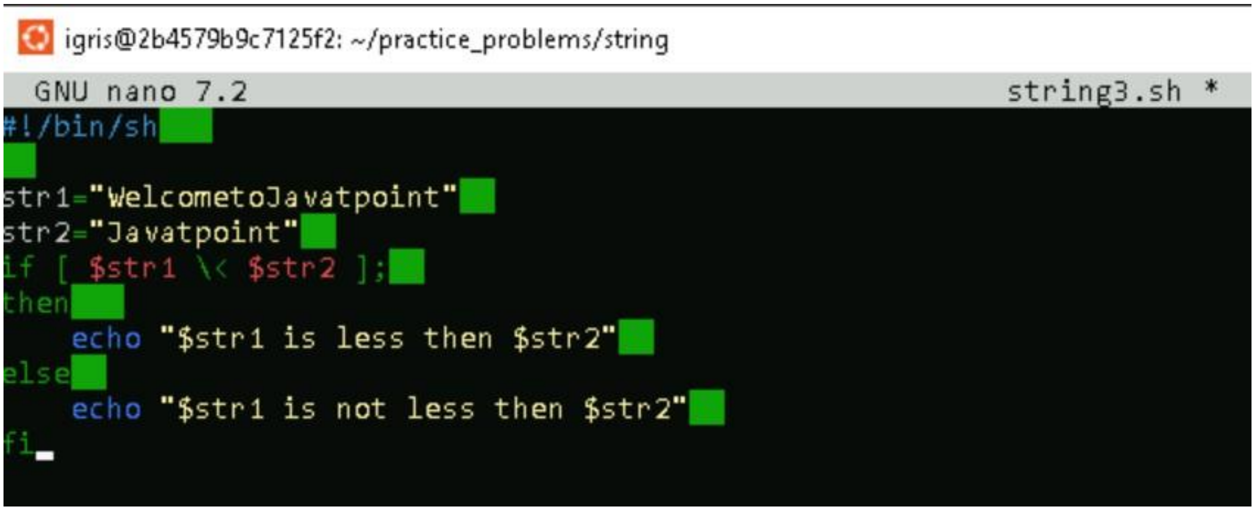


```
#!/bin/bash
#Script to check whether two strings are equal.

str1="WelcometoJavatpoint."
str2="javatpoint"

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

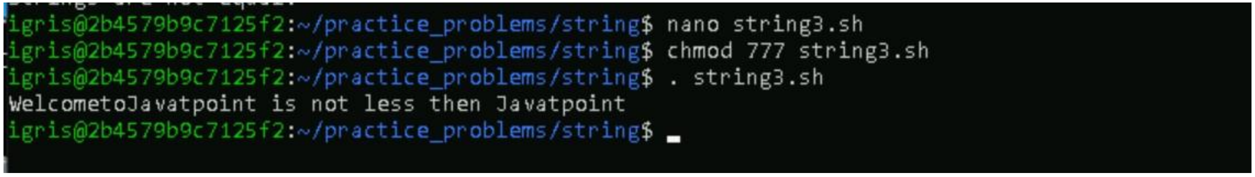


```
igris@2b4579b9c7125f2:~/practice_problems/string$ nano string2.sh
igris@2b4579b9c7125f2:~/practice_problems/string$ chmod 777 string2.sh
igris@2b4579b9c7125f2:~/practice_problems/string$ . string2.sh
Strings are not equal.
igris@2b4579b9c7125f2:~/practice_problems/string$
```

Script 3: Less than operator

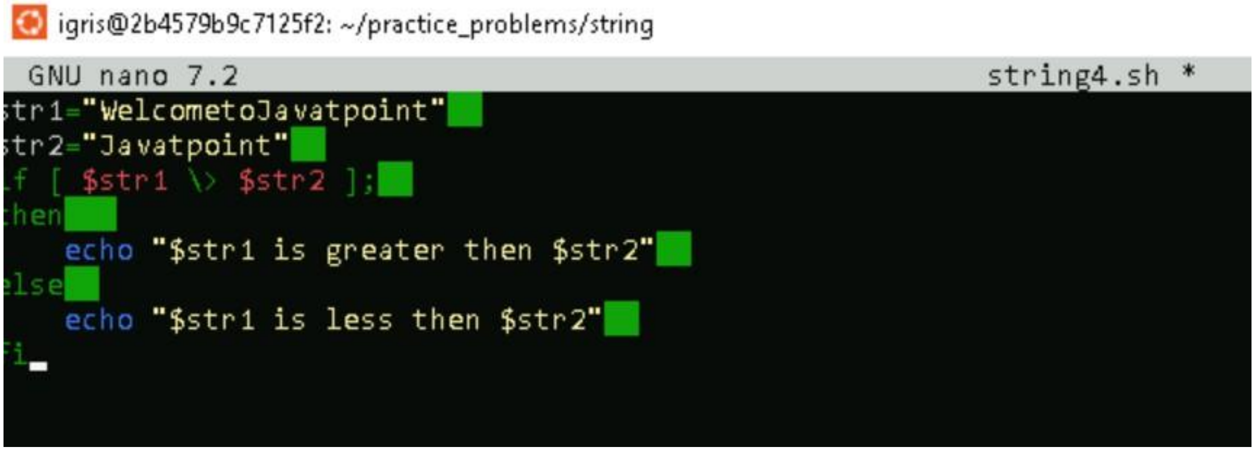


```
#!/bin/sh

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

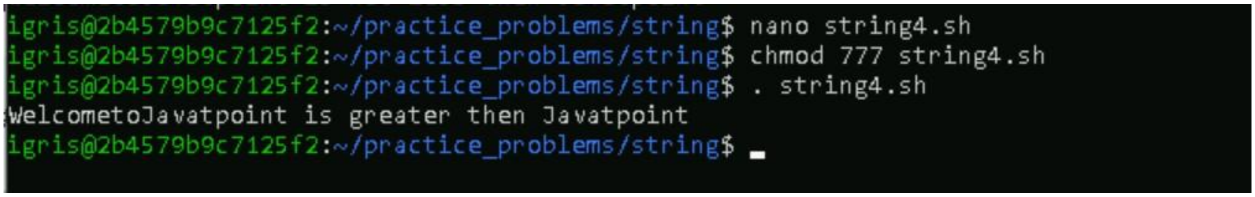


```
igris@2b4579b9c7125f2:~/practice_problems/string$ nano string3.sh
igris@2b4579b9c7125f2:~/practice_problems/string$ chmod 777 string3.sh
igris@2b4579b9c7125f2:~/practice_problems/string$ . string3.sh
WelcometoJavatpoint is not less then Javatpoint
igris@2b4579b9c7125f2:~/practice_problems/string$
```

Script 4: Greater than operator

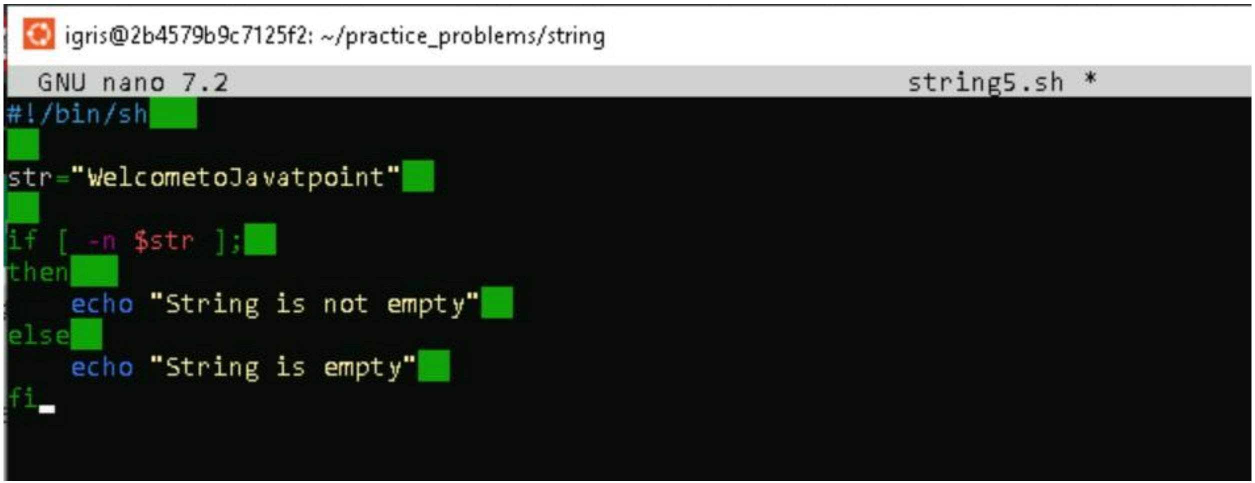


```
str1="WelcometoJavatpoint"
str2="Javatpoint"
if [ $str1 \> $str2 ];
then
    echo "$str1 is greater then $str2"
else
    echo "$str1 is less then $str2"
fi
```



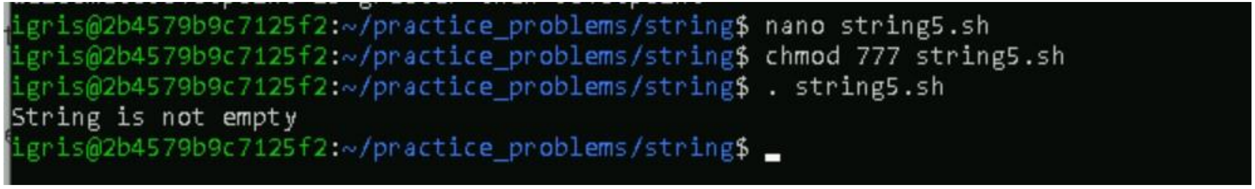
```
igris@2b4579b9c7125f2:~/practice_problems/string$ nano string4.sh
igris@2b4579b9c7125f2:~/practice_problems/string$ chmod 777 string4.sh
igris@2b4579b9c7125f2:~/practice_problems/string$ . string4.sh
WelcometoJavatpoint is greater then Javatpoint
igris@2b4579b9c7125f2:~/practice_problems/string$ 
```

Script 5: To check if the string length is greater than Zero



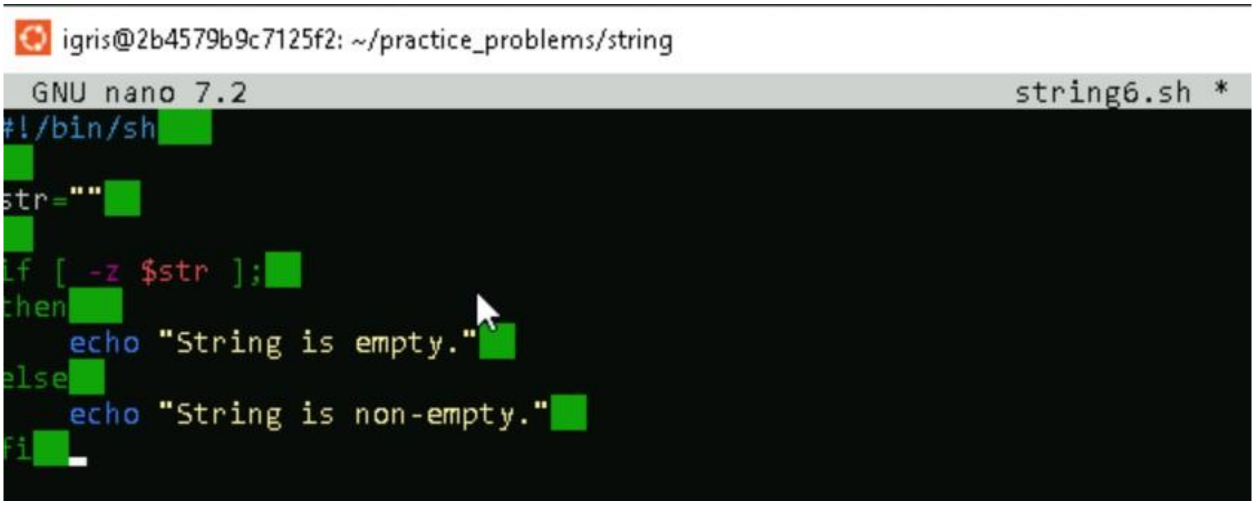
igris@2b4579b9c7125f2: ~/practice_problems/string

```
  GNU nano 7.2                                              string5.sh *
#!/bin/sh

str="WelcometoJavatpoint"

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

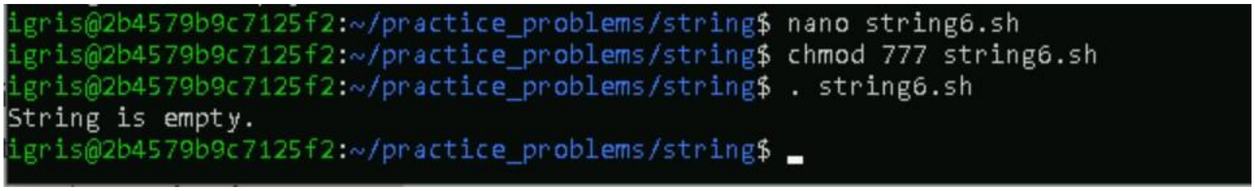


```
igris@2b4579b9c7125f2:~/practice_problems/string$ nano string5.sh
igris@2b4579b9c7125f2:~/practice_problems/string$ chmod 777 string5.sh
igris@2b4579b9c7125f2:~/practice_problems/string$ . string5.sh
String is not empty
igris@2b4579b9c7125f2:~/practice_problems/string$ 
```

Script 6: To check if the string length is equal to Zero



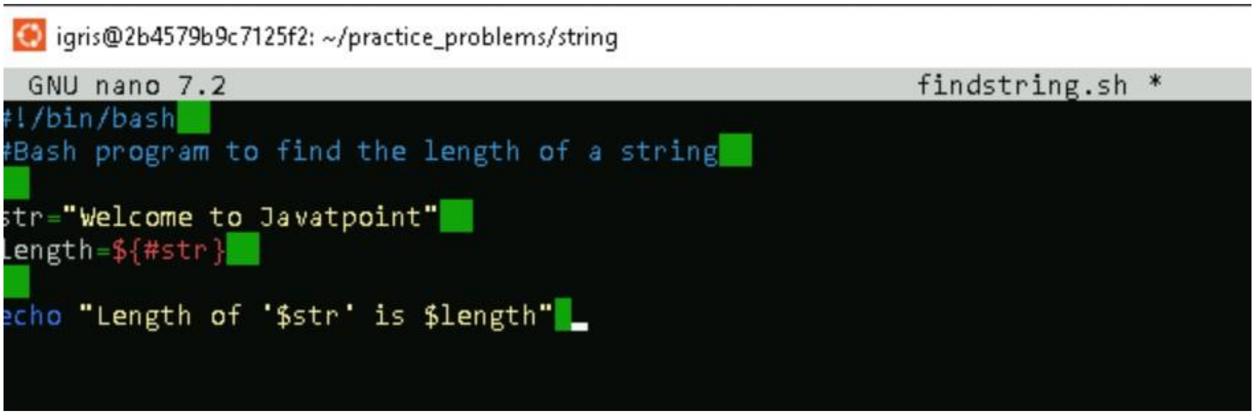
```
igris@2b4579b9c7125f2: ~/practice_problems/string
 GNU nano 7.2                                                    string6.sh *
#!/bin/sh

str=""

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



```
igris@2b4579b9c7125f2:~/practice_problems/string$ nano string6.sh
igris@2b4579b9c7125f2:~/practice_problems/string$ chmod 777 string6.sh
igris@2b4579b9c7125f2:~/practice_problems/string$ . string6.sh
String is empty.
igris@2b4579b9c7125f2:~/practice_problems/string$
```

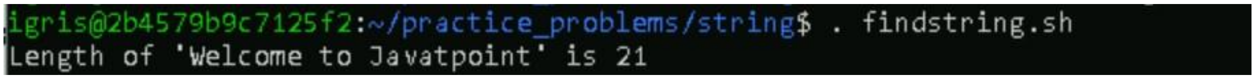
Bash FIND STRING

Script 1: To find the string length in bash



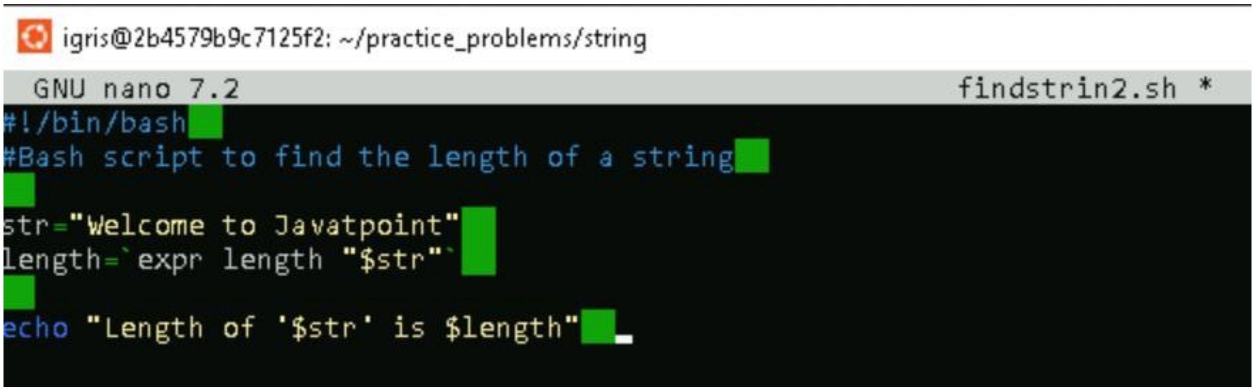
```
igris@2b4579b9c7125f2: ~/practice_problems/string
 GNU nano 7.2                                               findstring.sh *
#!/bin/bash
#Bash program to find the length of a string

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

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

Pratheek U B
289226



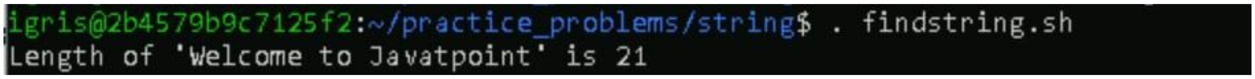
```
igris@2b4579b9c7125f2:~/practice_problems/string$ . findstring.sh
Length of 'Welcome to Javatpoint' is 21
```

Script 2:



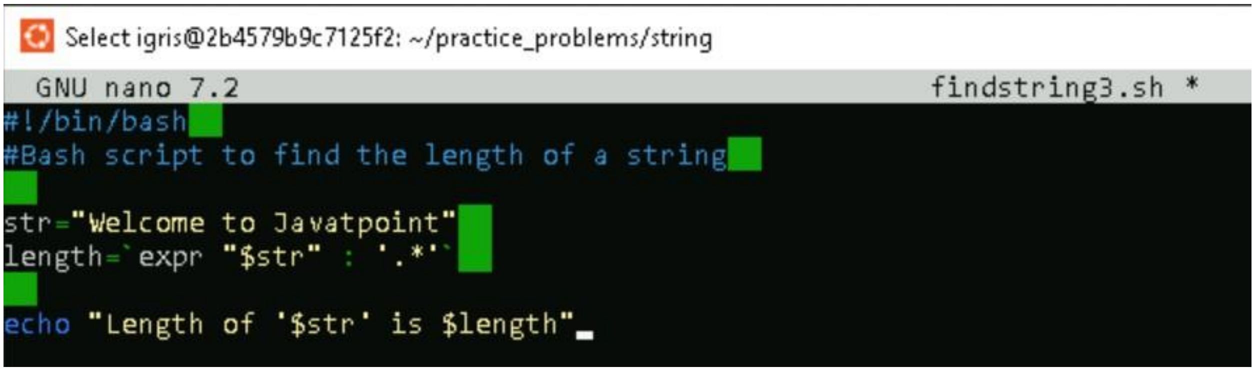
igris@2b4579b9c7125f2: ~/practice_problems/string

```
  GNU nano 7.2                                              findstrin2.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"
```



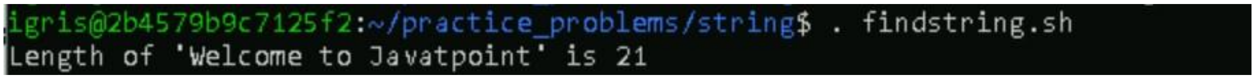
```
igris@2b4579b9c7125f2:~/practice_problems/string$ . findstring.sh
Length of 'Welcome to Javatpoint' is 21
```

Script 3:



Select igris@2b4579b9c7125f2: ~/practice_problems/string

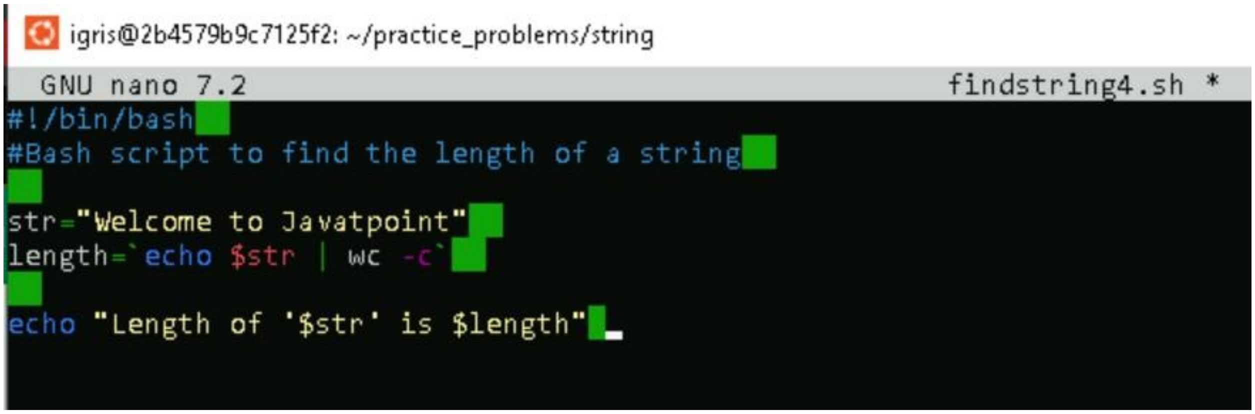
```
  GNU nano 7.2                                            findstring3.sh *
#!/bin/bash
#Bash script to find the length of a string

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

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



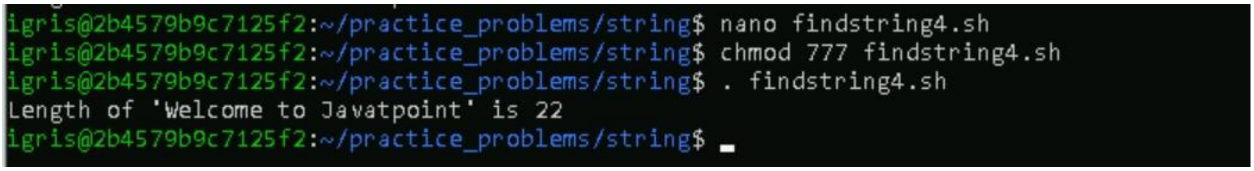
```
igris@2b4579b9c7125f2:~/practice_problems/string$ . findstring.sh
Length of 'Welcome to Javatpoint' is 21
```

Pratheek U B
289226

Script 4:

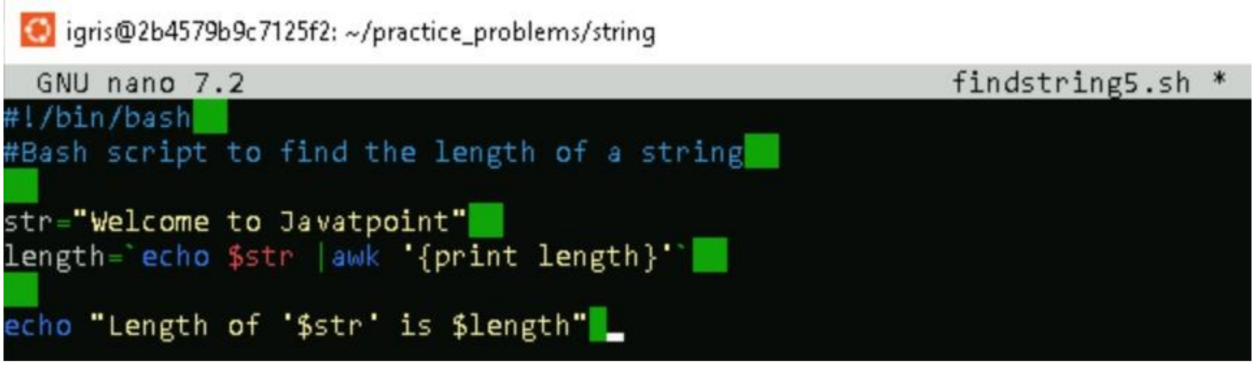


```
igris@2b4579b9c7125f2: ~/practice_problems/string
 GNU nano 7.2                                          findstring4.sh *
#!/bin/bash
#Bash script to find the length of a string

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

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

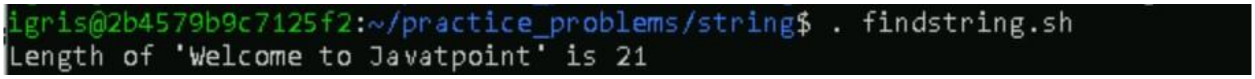


```
igris@2b4579b9c7125f2:~/practice_problems/string$ nano findstring4.sh
igris@2b4579b9c7125f2:~/practice_problems/string$ chmod 777 findstring4.sh
igris@2b4579b9c7125f2:~/practice_problems/string$ . findstring4.sh
Length of 'Welcome to Javatpoint' is 22
igris@2b4579b9c7125f2:~/practice_problems/string$
```

Script 5:



```
igris@2b4579b9c7125f2: ~/practice_problems/string
 GNU nano 7.2                                          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"
```

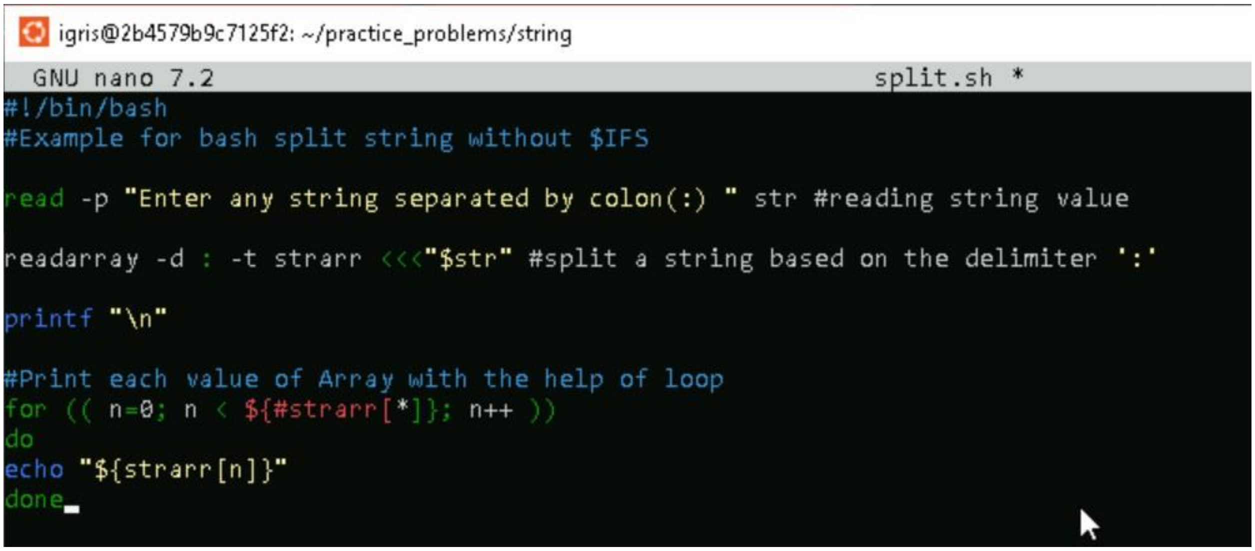


```
igris@2b4579b9c7125f2:~/practice_problems/string$ . findstring.sh
Length of 'Welcome to Javatpoint' is 21
```

Pratheek U B
289226

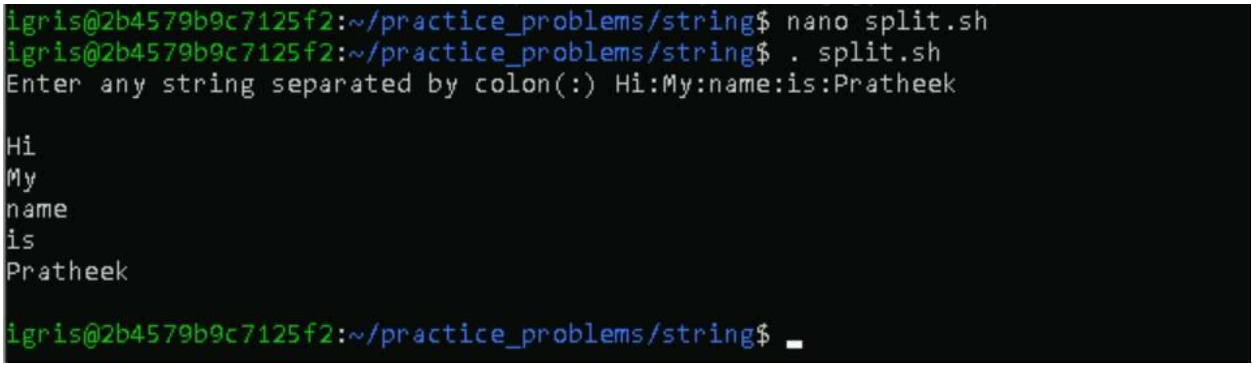
Bash SPLIT STRING

Script 1: Bash Split String by Space



igris@2b4579b9c7125f2: ~/practice_problems/string

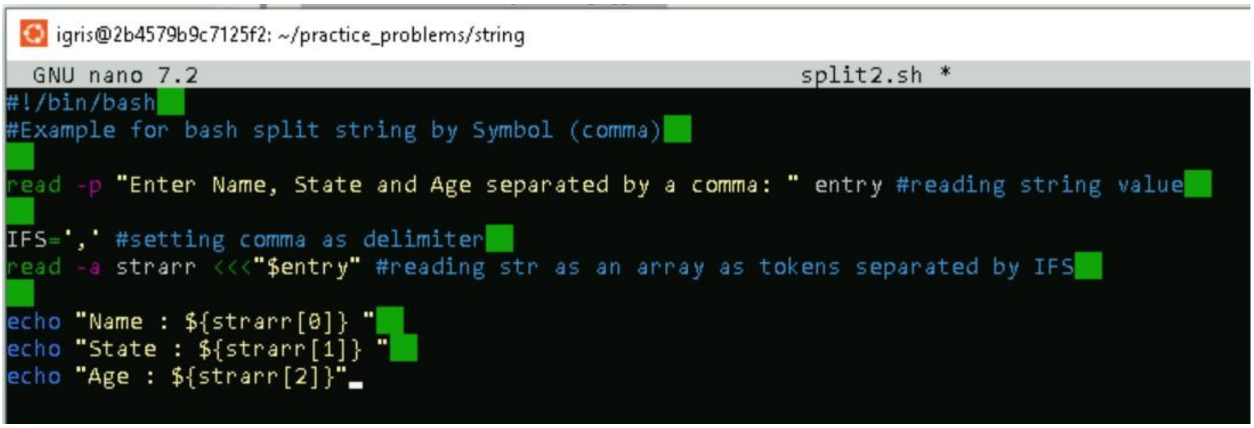
```
  GNU nano 7.2                                              split.sh *
#!/bin/bash
#Example for bash split string without $IFS

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



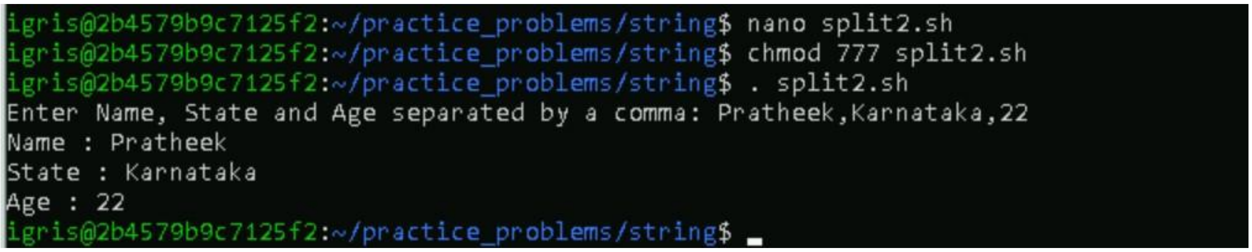
```
igris@2b4579b9c7125f2:~/practice_problems/string$ nano split.sh
igris@2b4579b9c7125f2:~/practice_problems/string$ . split.sh
Enter any string separated by colon(:) Hi:My:name:is:Pratheek

Hi
My
name
is
Pratheek

igris@2b4579b9c7125f2:~/practice_problems/string$
```

Pratheek U B
289226

Script 2: Bash split string by symbol

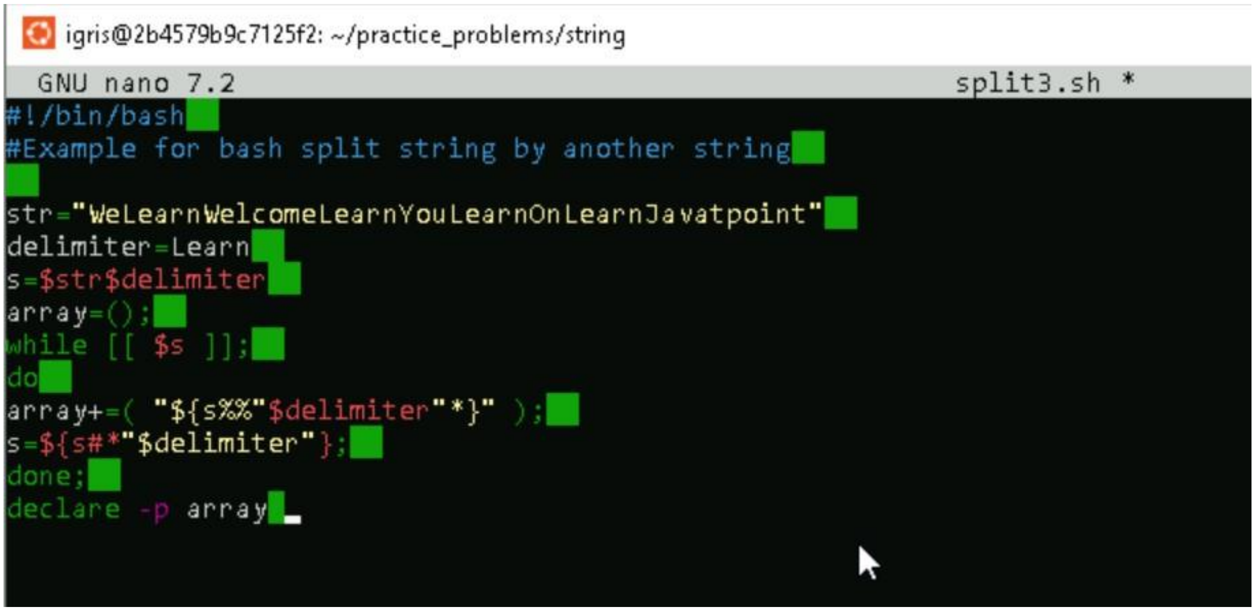


```
#!/bin/bash
#Example for bash split string by Symbol (comma)

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]}"
```



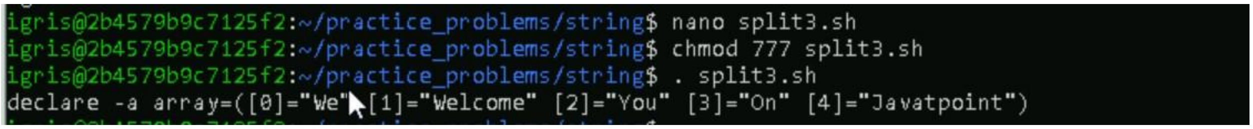
```
igris@2b4579b9c7125f2:~/practice_problems/string$ nano split2.sh
igris@2b4579b9c7125f2:~/practice_problems/string$ chmod 777 split2.sh
igris@2b4579b9c7125f2:~/practice_problems/string$ . split2.sh
Enter Name, State and Age separated by a comma: Pratheek,Karnataka,22
Name : Pratheek
State : Karnataka
Age : 22
igris@2b4579b9c7125f2:~/practice_problems/string$
```

Script 3: Bash split string by another string



```
#!/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
```

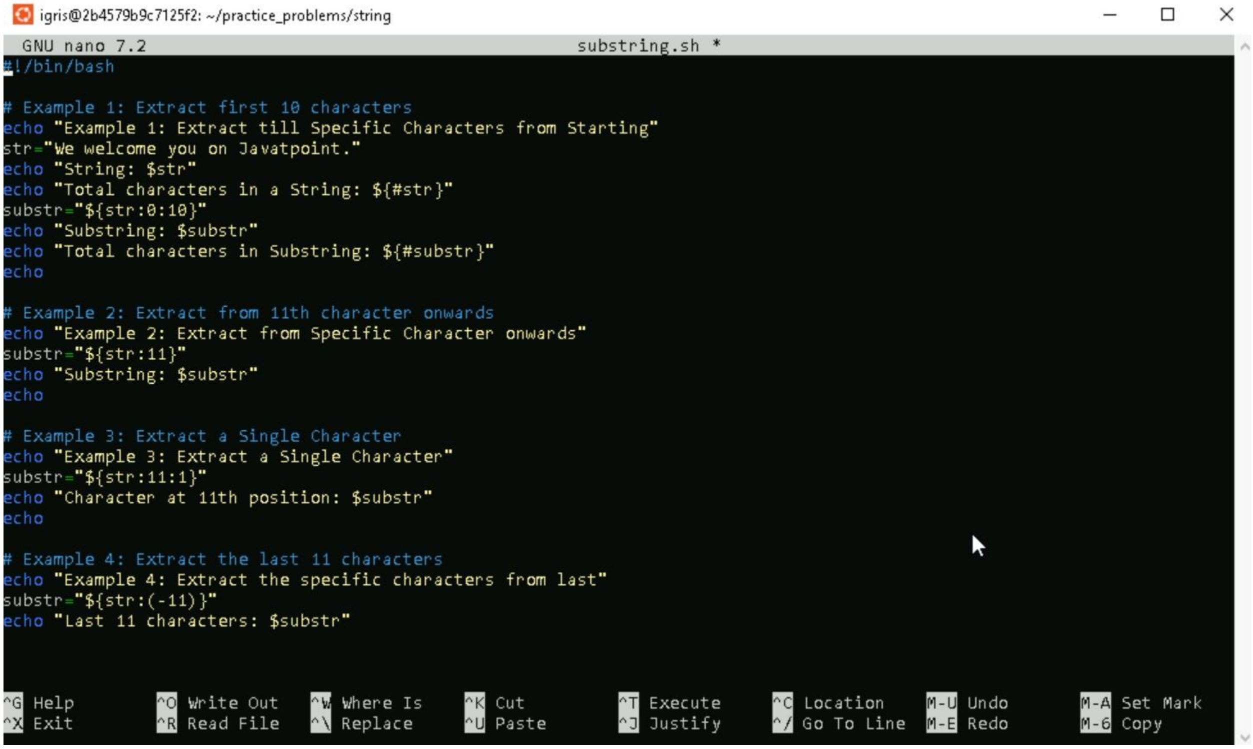
Pratheek U B
289226



```
igris@2b4579b9c7125f2:~/practice_problems/string$ nano split3.sh
igris@2b4579b9c7125f2:~/practice_problems/string$ chmod 777 split3.sh
igris@2b4579b9c7125f2:~/practice_problems/string$ . split3.sh
declare -a array=([0]="We" [1]="Welcome" [2]="You" [3]="On" [4]="Javatpoint")
```

Bash SUBSTRING

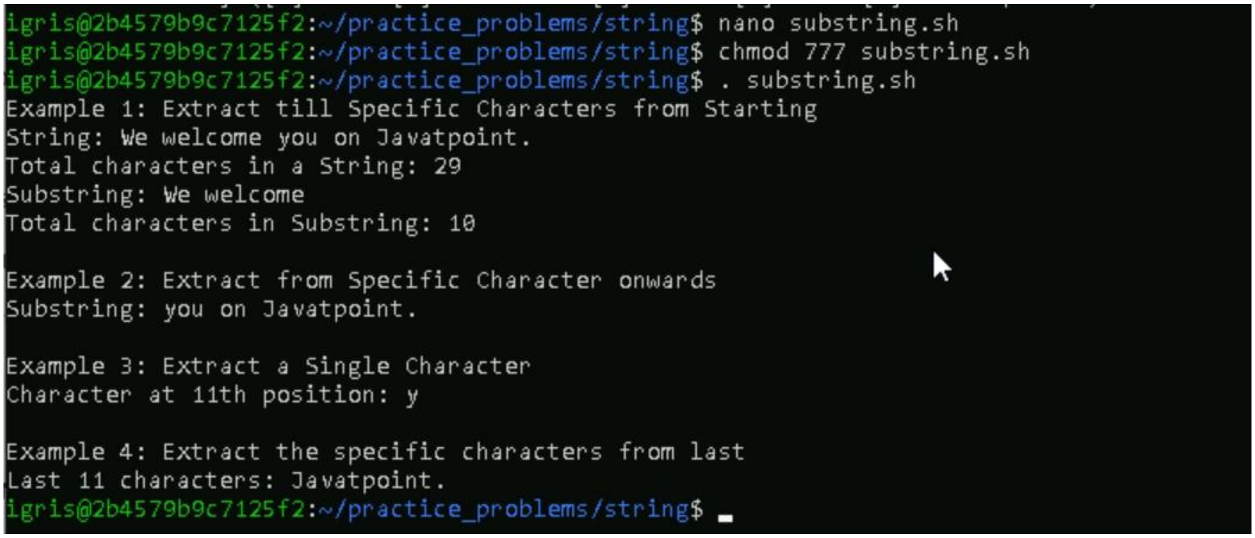
Script 1:



igris@2b4579b9c7125f2: ~/practice_problems/string

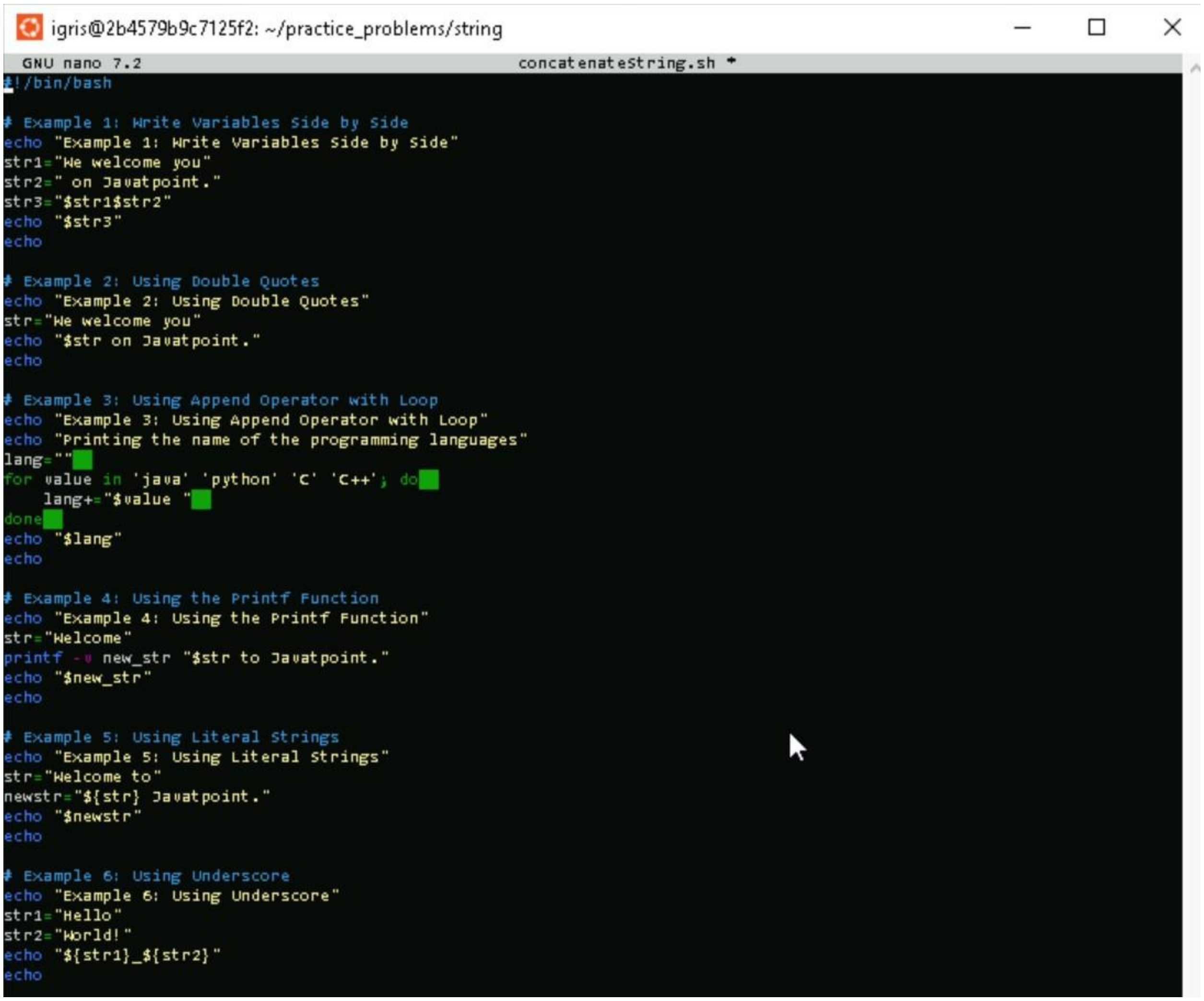
```
  GNU nano 7.2                                 substring.sh *
#!/bin/bash

# Example 1: Extract first 10 characters
echo "Example 1: Extract till Specific Characters from Starting"
str="We welcome you on Javatpoint."
echo "String: $str"
echo "Total characters in a String: ${#str}"
substr="${str:0:10}"
echo "Substring: $substr"
echo "Total characters in Substring: ${#substr}"
echo

# Example 2: Extract from 11th character onwards
echo "Example 2: Extract from Specific Character onwards"
substr="${str:11}"
echo "Substring: $substr"
echo

# Example 3: Extract a Single Character
echo "Example 3: Extract a Single Character"
substr="${str:11:1}"
echo "Character at 11th position: $substr"
echo

# Example 4: Extract the last 11 characters
echo "Example 4: Extract the specific characters from last"
substr="${str:(-11)}"
echo "Last 11 characters: $substr"


^G Help       ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo      M-A Set Mark
^X Exit       ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line M-E Redo      M-6 Copy
```

Pratheek U B
289226

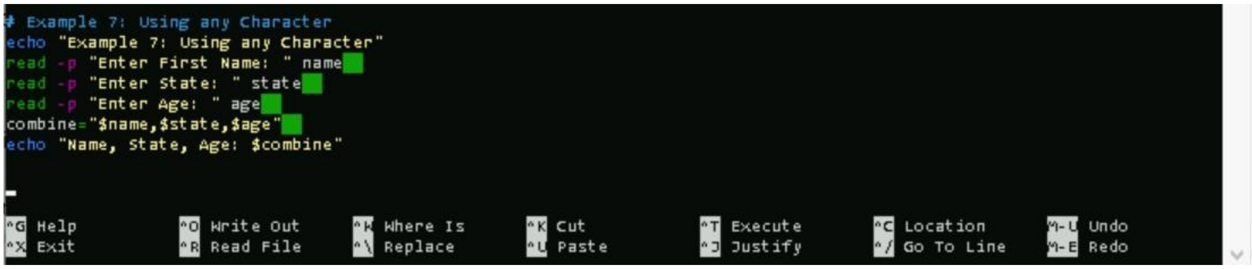


```
igris@2b4579b9c7125f2:~/practice_problems/string$ nano substring.sh
igris@2b4579b9c7125f2:~/practice_problems/string$ chmod 777 substring.sh
igris@2b4579b9c7125f2:~/practice_problems/string$ . substring.sh
Example 1: Extract till Specific Characters from Starting
String: We welcome you on Javatpoint.
Total characters in a String: 29
Substring: We welcome
Total characters in Substring: 10

Example 2: Extract from Specific Character onwards
Substring: you on Javatpoint.

Example 3: Extract a Single Character
Character at 11th position: y

Example 4: Extract the specific characters from last
Last 11 characters: Javatpoint.
igris@2b4579b9c7125f2:~/practice_problems/string$ _
```

Bash CONCATENATE STRING



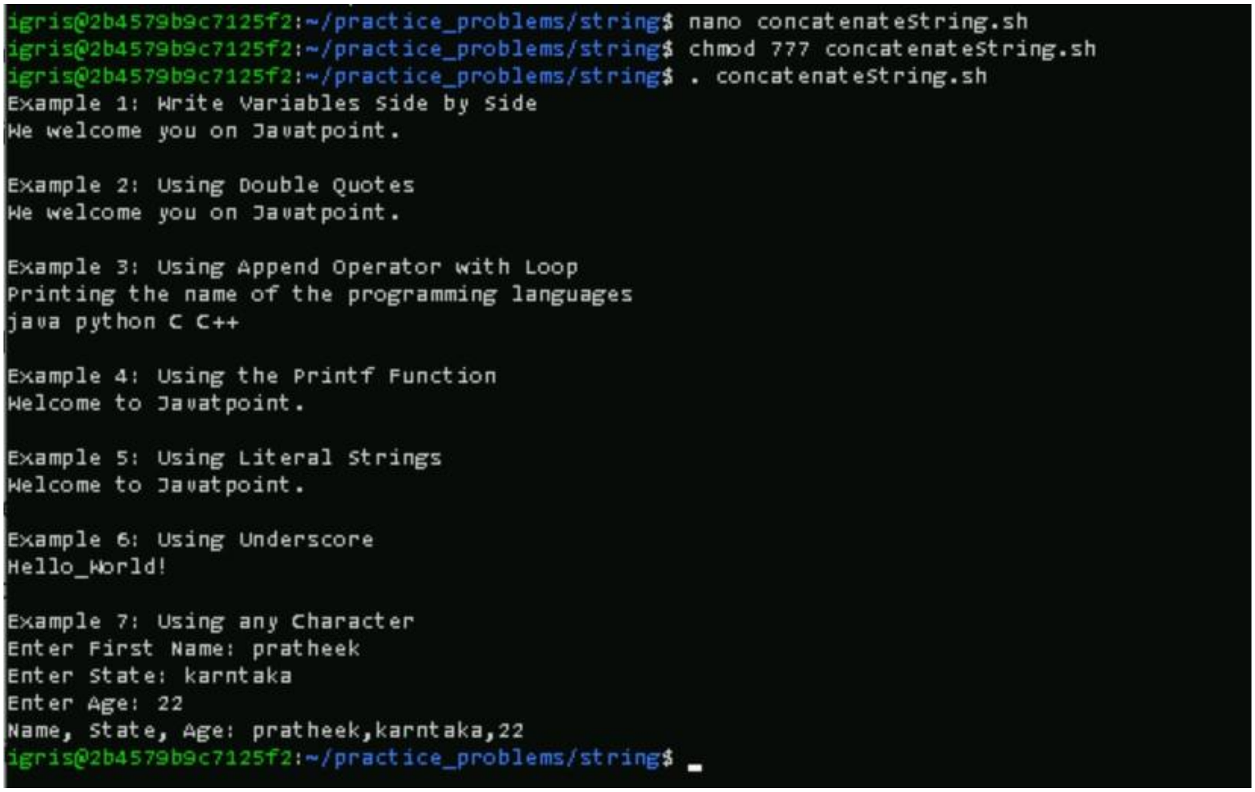
```
#!/bin/bash

# Example 1: Write Variables Side by Side
echo "Example 1: Write Variables Side by Side"
str1="We welcome you"
str2=" on Javatpoint."
str3="$str1$str2"
echo "$str3"
echo

# Example 2: Using Double Quotes
echo "Example 2: Using Double Quotes"
str="We welcome you"
echo "$str on Javatpoint."
echo

# Example 3: Using Append Operator with Loop
echo "Example 3: Using Append Operator with Loop"
echo "Printing the name of the programming languages"
lang=""
for value in 'java' 'python' 'C' 'C++'; do
    lang+="$value "
done
echo "$lang"
echo

# Example 4: Using the Printf Function
echo "Example 4: Using the Printf Function"
str="Welcome"
printf -v new_str "$str to Javatpoint."
echo "$new_str"
echo

# Example 5: Using Literal Strings
echo "Example 5: Using Literal Strings"
str="Welcome to"
newstr="${str} Javatpoint."
echo "$newstr"
echo

# Example 6: Using Underscore
echo "Example 6: Using Underscore"
str1="Hello"
str2="World!"
echo "${str1}_${str2}"
echo
```



```
# Example 7: Using any Character
echo "Example 7: Using any Character"
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"
```

Pratheek U B
289226

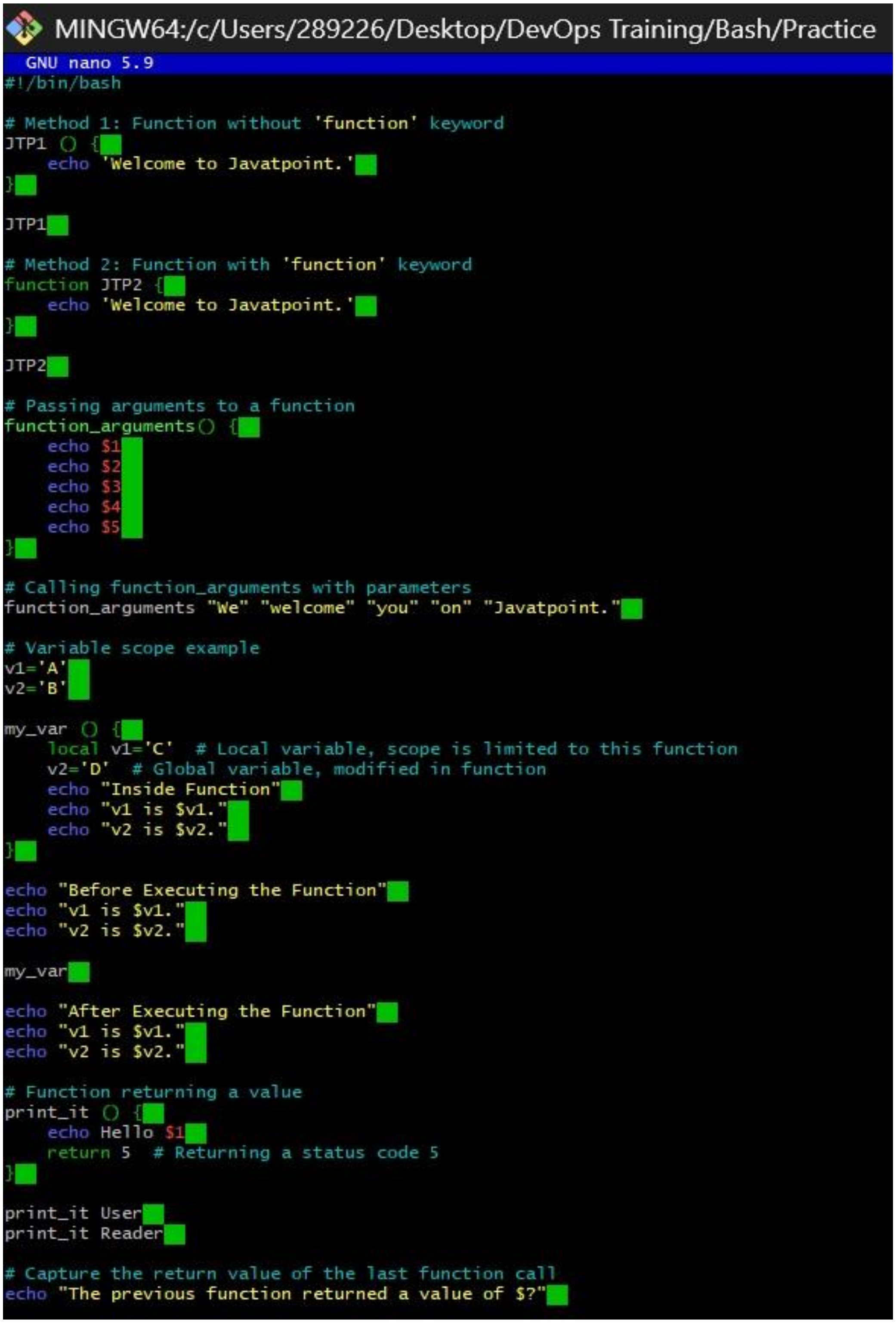
OUTPUT:



```
igris@2b4579b9c7125f2:~/practice_problems/string$ nano concatenateString.sh
igris@2b4579b9c7125f2:~/practice_problems/string$ chmod 777 concatenateString.sh
igris@2b4579b9c7125f2:~/practice_problems/string$ . concatenateString.sh
Example 1: Write Variables Side by Side
We welcome you on Javatpoint.

Example 2: Using Double Quotes
We welcome you on Javatpoint.

Example 3: Using Append Operator with Loop
Printing the name of the programming languages
java python C C++

Example 4: Using the Printf Function
Welcome to Javatpoint.

Example 5: Using Literal Strings
Welcome to Javatpoint.

Example 6: Using Underscore
Hello_World!

Example 7: Using any Character
Enter First Name: pratheek
Enter State: karntaka
Enter Age: 22
Name, State, Age: pratheek,karntaka,22
igris@2b4579b9c7125f2:~/practice_problems/string$ _
```

Pratheek U B
289226

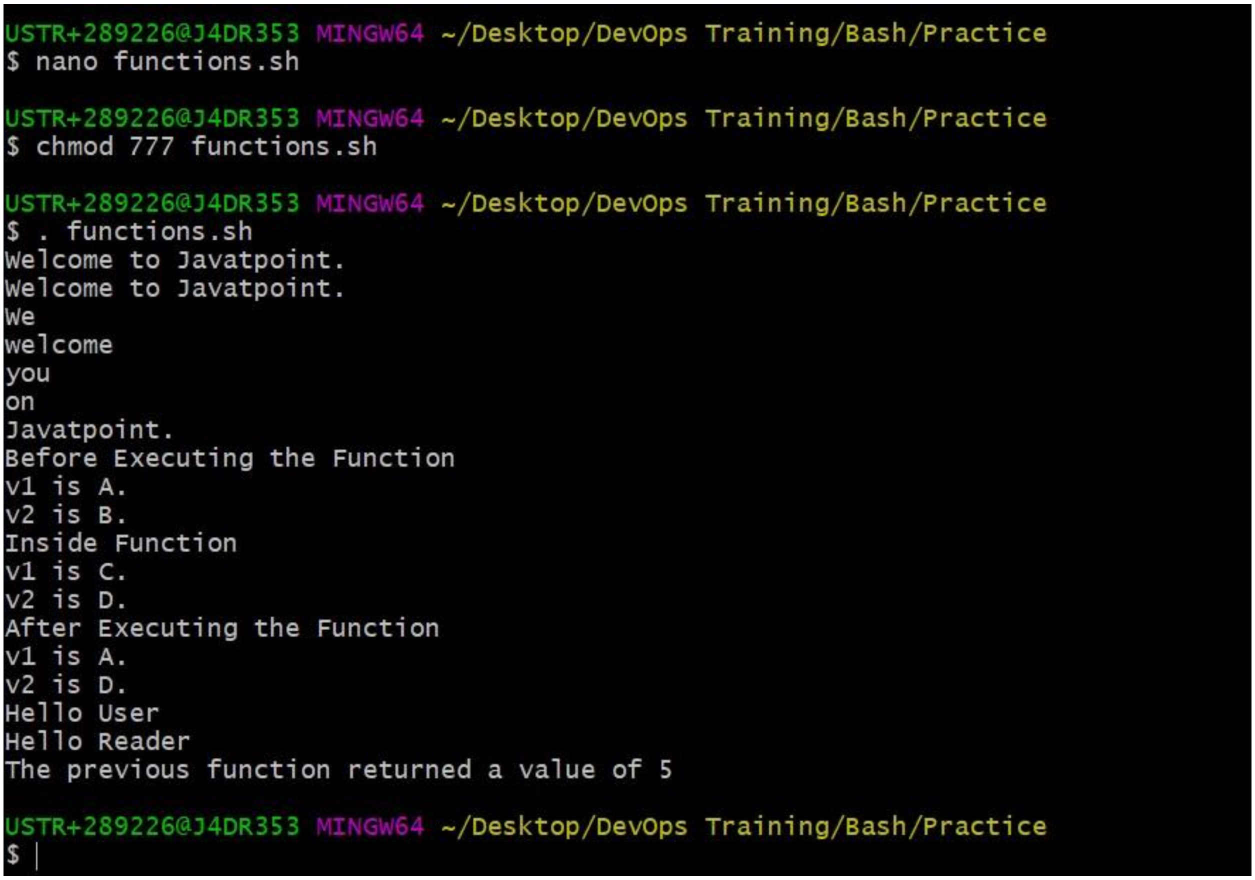
Bash FUNCTIONS



MINGW64:/c/Users/289226/Desktop/DevOps Training/Bash/Practice

```
  GNU nano 5.9
#!/bin/bash

# Method 1: Function without 'function' keyword
JTP1 () {
    echo 'Welcome to Javatpoint.'
}

JTP1

# Method 2: Function with 'function' keyword
function JTP2 {
    echo 'Welcome to Javatpoint.'
}

JTP2

# Passing arguments to a function
function_arguments() {
    echo $1
    echo $2
    echo $3
    echo $4
    echo $5
}

# Calling function_arguments with parameters
function_arguments "We" "welcome" "you" "on" "Javatpoint."

# Variable scope example
v1='A'
v2='B'

my_var () {
    local v1='C'  # Local variable, scope is limited to this function
    v2='D'  # Global variable, modified in function
    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."

# Function returning a value
print_it () {
    echo Hello $1
    return 5  # Returning a status code 5
}

print_it User
print_it Reader

# Capture the return value of the last function call
echo "The previous function returned a value of $?"
```

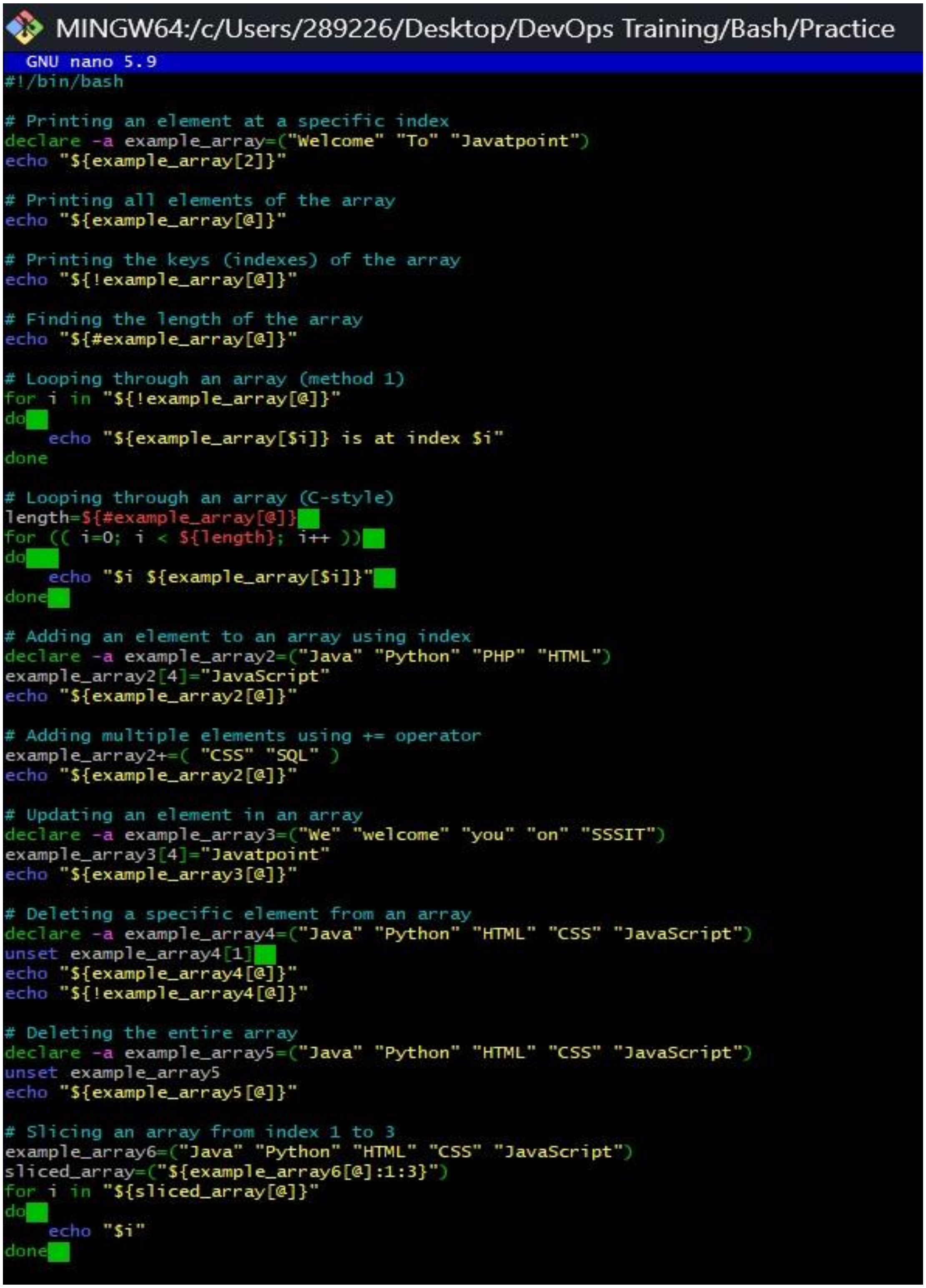
OUTPUT:



```
USTR+289226@J4DR353 MINGW64 ~/Desktop/DevOps Training/Bash/Practice
$ nano functions.sh

USTR+289226@J4DR353 MINGW64 ~/Desktop/DevOps Training/Bash/Practice
$ chmod 777 functions.sh

USTR+289226@J4DR353 MINGW64 ~/Desktop/DevOps Training/Bash/Practice
$ . functions.sh
Welcome to Javatpoint.
Welcome to Javatpoint.
We
welcome
you
on
Javatpoint.
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.
Hello User
Hello Reader
The previous function returned a value of 5

USTR+289226@J4DR353 MINGW64 ~/Desktop/DevOps Training/Bash/Practice
$ 
```

Pratheek U B
289226

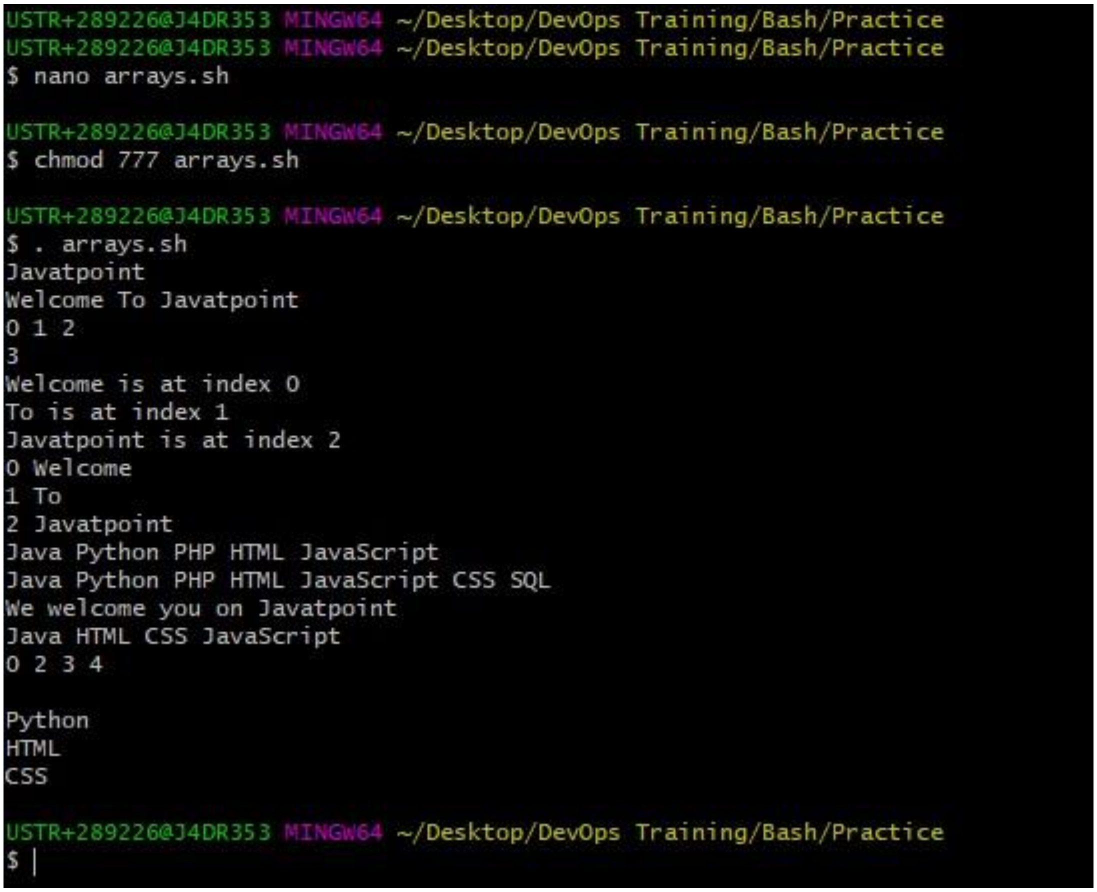
Bash ARRAYS



MINGW64:/c/Users/289226/Desktop/DevOps Training/Bash/Practice
GNU nano 5.9

```
#!/bin/bash

# Printing an element at a specific index
declare -a example_array=("Welcome" "To" "Javatpoint")
echo "${example_array[2]}"

# Printing all elements of the array
echo "${example_array[@]}"

# Printing the keys (indexes) of the array
echo "${!example_array[@]}"

# Finding the length of the array
echo "${#example_array[@]}"

# Looping through an array (method 1)
for i in "${!example_array[@]}"
do
    echo "${example_array[$i]} is at index $i"
done

# Looping through an array (C-style)
length=${#example_array[@]}
for (( i=0; i < ${length}; i++ ))
do
    echo "$i ${example_array[$i]}"
done

# Adding an element to an array using index
declare -a example_array2=("Java" "Python" "PHP" "HTML")
example_array2[4]="JavaScript"
echo "${example_array2[@]}"

# Adding multiple elements using += operator
example_array2+=( "CSS" "SQL" )
echo "${example_array2[@]}"

# Updating an element in an array
declare -a example_array3=("We" "welcome" "you" "on" "SSSIT")
example_array3[4]="Javatpoint"
echo "${example_array3[@]}"

# Deleting a specific element from an array
declare -a example_array4=("Java" "Python" "HTML" "CSS" "JavaScript")
unset example_array4[1]
echo "${example_array4[@]}"
echo "${!example_array4[@]}"

# Deleting the entire array
declare -a example_array5=("Java" "Python" "HTML" "CSS" "JavaScript")
unset example_array5
echo "${example_array5[@]}"

# Slicing an array from index 1 to 3
example_array6=("Java" "Python" "HTML" "CSS" "JavaScript")
sliced_array=("${example_array6[@]:1:3}")
for i in "${sliced_array[@]}"
do
    echo "$i"
done
```

OUTPUT:



```
USTR+289226@J4DR353 MINGW64 ~/Desktop/DevOps Training/Bash/Practice
USTR+289226@J4DR353 MINGW64 ~/Desktop/DevOps Training/Bash/Practice
$ nano arrays.sh

USTR+289226@J4DR353 MINGW64 ~/Desktop/DevOps Training/Bash/Practice
$ chmod 777 arrays.sh

USTR+289226@J4DR353 MINGW64 ~/Desktop/DevOps Training/Bash/Practice
$ . arrays.sh
Javatpoint
Welcome To Javatpoint
0 1 2
3
Welcome is at index 0
To is at index 1
Javatpoint is at index 2
0 Welcome
1 To
2 Javatpoint
Java Python PHP HTML JavaScript
Java Python PHP HTML JavaScript CSS SQL
We welcome you on Javatpoint
Java HTML CSS JavaScript
0 2 3 4

Python
HTML
CSS

USTR+289226@J4DR353 MINGW64 ~/Desktop/DevOps Training/Bash/Practice
$ |
```