***Shell Script Programs***

**1. Display greet message using the system time.**

#Experiment 02, Program 01: To display proper greetings based on time.  
#Adnan Ismail Shah Muzavor  
  
hours="`date +%H`"  
  
if [ $hours -ge 1 -a $hours -le 11 ]  
then  
  echo "Good Morning!!"  
elif [ $hours -ge 12 -a $hours -le 15 ]  
then  
  echo "Good Afternoon!!"  
else  
  echo "Good Evening!!"  
fi

**Output**



**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**2. Find the factorial of a given number.**

#Experiment 2, program 2: To print factorial of the given number

#Adnan Ismail Shah Muzavor

echo "Enter the Number whose factorial is to be found: "

read num

fact=1

for (( i=1 ; i<=$num ; i++ ))

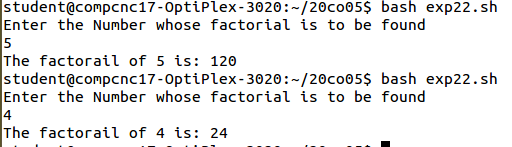
do

fact=`expr $fact \\* $i `

done

echo "The factorail of $num is: $fact"

**Output**

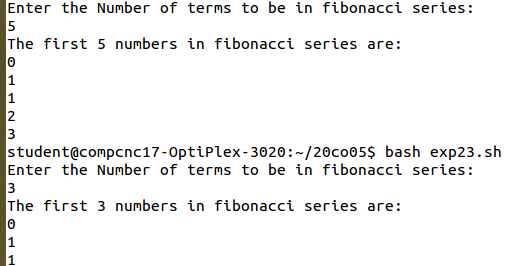


**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**3. Generate the fibonacci series for the number of terms entered by the user.**

#Experiment 2, program 3: Generate the fibonacci series for the number of terms entered by the user.  
#Adnan Ismail Shah Muzavor  
echo "Enter the Number of terms to be in fibonacci series: "  
read num  
a=0  
b=1  
for (( i=1 ; i<=$num ; i++ ))  
do  
echo "$a"  
c=` expr $a + $b `  
a=` expr $b + 0 `  
b=` expr $c + 0 `  
done

**Output**

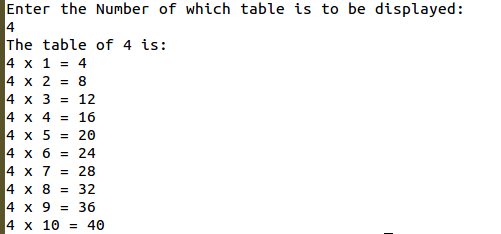


**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**4. To print the number table.**

#Experiment 2, program 4: To print the number table.  
#Adnan Ismail Shah Muzavor  
echo "Enter the Number of which table is to be displayed: "  
read num  
  
echo "The table of $num is: "  
for (( i=1 ; i<=10 ; i++ ))  
do  
table=`expr $num \\* $i `  
echo "$num x $i = $table"  
done

**Output**

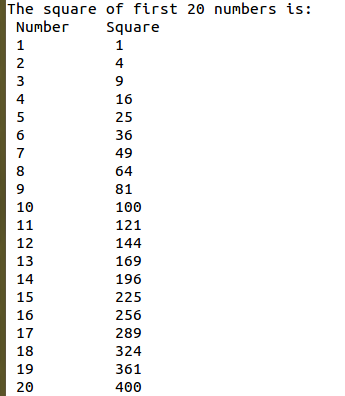


**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**5. To display square of first 20 numbers.**

#Experiment 2, program 4: To display square of first 20 numbers.  
#Adnan Ismail Shah Muzavor  
echo "The square of first 20 numbers is: "  
echo " Number    Square"  
for (( i=1 ; i<=20 ; i++ ))  
do  
if [ $i -le 9 ]  
then  
echo " $i          `expr $i \\* $i`"  
else  
echo " $i         `expr $i \\* $i`"  
fi  
done

**Output**

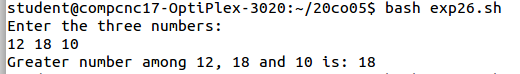


**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**6. To print the largest of 3 numbers entered by the user.**

#Experiment 2, Program 6: To print largest of 3 numbers entered by the user  
#Adnan Ismail Shah Muzavor  
echo "Enter the three numbers: "  
read num1 num2 num3  
grtr=0  
if [ $num1 -ge $num2 -a $num1 -ge $num3 ]  
then  
grtr=$num1  
elif [ $num2 -ge $num1 -a $num2 -ge $num3 ]  
then  
grtr=$num2  
else  
grtr=$num3  
fi

**Output**

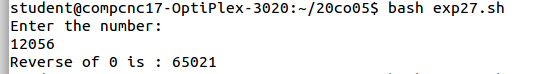


**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**7. To reverse a number.**

#Experiment 2, Program 7: To print reverse of number entered by user  
#Adnan Ismail Shah Muzavor  
echo "Enter the number: "  
read num  
rev=0  
temp=0  
digit=0  
while [ $num -ne 0 ]  
do  
digit=`expr $num % 10`   #get the digit  
num=`expr $num / 10`     #update the number  
rev=`expr $rev \\* 10`    #Multiply prev digit by 10  
rev=`expr $rev + $digit` #Append current digit to it  
  
done

**Output**



**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**8. To reverse a string.**

#Experiment 2, Program 8: To reverse the string entered by the user

#Adnan Ismail Shah Muzavor

echo -n "Enter a string: "

read string

length=`expr length $string`

while [ $length -gt 0 ]

do

# cut -c index will display/return character from index value

# hence for first iteration nth (last\_char) will be returned

# hence for second iteration n-1th (second\_last) will be returned

#Append length (indexed) character into the string

rev=$rev`echo $string | cut -c $length` #cut the string from length-index

#character and append to rev

length=`expr $length - 1`

done

echo "Reverse of $string is: $rev"

**Output**

student@compcnc17-optiPlex-3020:~/20co05$ bash exp28.sh

Enter a string: HelloReverseMe

Reverse of HelloReverseMe is: eMesreveRolleH

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

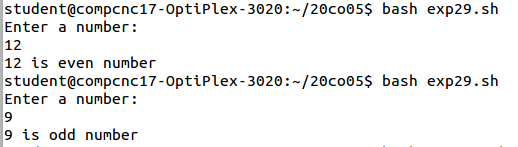
**9. Check whether the entered number is odd or even.**

#Experiment 02, program 9:Check whether entered number is even or not

#Adnan Ismail Shah Muzavor.

echo "Enter a number: ";  
read num  
rem=`expr $num % 2`  
if [ $rem -eq 0 ]  
then  
echo "$num is even number"  
else  
echo "$num is odd number"  
fi

**Output**

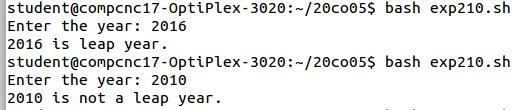


**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**10. Check Whether given year is leap year or not.**

#Experiment 02, program 10:Check whether given year is leap year or not.  
#Adnan Ismail Shah Muzavor.\_echo -n "Enter a year: "  
  
echo -n "Enter the year: "  
read year  
  
#div by 400 then its leap year  
#not div by 400 then should be div by 4 and not by 100  
check0=`expr $year % 100`  
check1=`expr $year % 400`  
check2=`expr $year % 4`  
  
#div by 400 then its leap year  
if [ $check1 -eq 0 ]  
then  
echo "$year is leap year."  
#not div by 400 then should be div by 4 and not by 100  
elif [ $check2 -eq 0 -a $check0 -ne 0 ]  
then  
echo "$year is leap year."  
else  
echo "$year is not a leap year."  
fi

**Output**



**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**11. To print all the prime numbers less than 20.**

#Experiment 02, program 11:Print all prime numbers less then 20

#Adnan Ismail Shah Muzavor.

#Experiment 02, program 11:Print all prime numbers less then 20  
#Adnan Ismail Shah Muzavor.  
  
echo -n "Prime numbers less then 20 are: "  
for (( i=2 ; i<=20 ; i++ ))  
do  
  temp=0  
  for (( j=2 ; j<$i ; j++ ))  
  do  
    #If any number preceding to i divides i then i is not a prime number  
    if [ `expr $i % $j` -eq 0 ]  
    then  
       temp=1  #As we know that i is not prime, no need to loop ahead  
       break  
    fi  
  done  
  if [ $temp -eq 0 ]  
  then  
    echo -n "$i "  
  fi  
done  
echo ""

**Output**



**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**12. To check if the entered string is palindrome or not.**

#Experiment 2, Program 12: To check if string is palindromic or not

#Adnan Ismail Shah Muzavor

echo -n "Enter a string: "

read string

length=`expr length $string`

while [ $length -gt 0 ]

do

# cut -c index will display/return character from index value

# hence for first iteration nth (last\_char) will be returned

# hence for second iteration n-1th (second\_last) will be returned

#Append length (indexed) character into the string

rev=$rev`echo $string | cut -c $length` #cut the string from length-index

#character and append to rev

length=`expr $length - 1`

done

echo "Reverse of $string is: $rev"

if [ $rev == $string ]

then

echo "$string is palindromic"

else

echo "since $string != $rev, $string is not palindromic"

fi

**Output**

student@compcnc17-optiPlex-3020:~/20co05$ bash exp212.sh

Enter a string: reverse

Reverse of reverse is: esrever

since reverse != esrever, reverse is not palindromic

student@compcnc17-optiPlex-3020:~/20co05$ bash exp212.sh

Enter a string: abcdcba

Reverse of abcdcba is: abcdcba

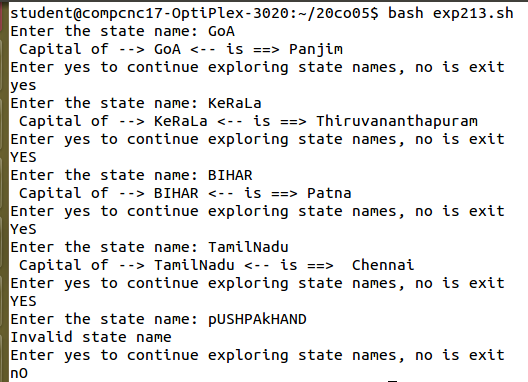
abcdcba is palindromic

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

1. **Using switch case print the capitals given the state name.**

#Experiment 02, program 17:Given the state name, print the capitals  
#Adnan Ismail Shah Muzavor.  
cont=1  
while [ $cont -eq 1 ]  
do  
echo -n "Enter the state name: "  
read sname  
  
case "$sname" in  
[Gg][Oo][Aa]) echo " Capital of --> $sname <-- is ==> Panjim " ;;  
[Tt][Aa][Mm][Ii][Ll][Nn][Aa][Dd][Uu]) echo " Capital of --> $sname <-- is ==>  Chennai " ;;  
[Bb][Ii][Hh][Aa][Rr]) echo " Capital of --> $sname <-- is ==> Patna " ;;  
[Kk][Ee][Rr][Aa][Ll][Aa]) echo " Capital of --> $sname <-- is ==> Thiruvananthapuram " ;;  
  \*) echo "Invalid state name"  
esac  
  
echo "Enter yes to continue exploring state names, no is exit"  
read ch2  
case "$ch2" in  
[Yy][Ee][Ss]) cont=1 ;;  
[Nn][Oo]) cont=0 ;;  
        \*) echo "Invalid choice, so choosing to continue";;  
esac  
done

**Output**



**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

1. **Write shell script to generate following series: 20, 22, 18, 20, 16, 18, 14, 16.**

#Experiment 02, program 14:Print the given series (20,22,18,20,16,18,14,16)

#Adnan Ismail Shah Muzavor.

echo -n "Series is:  " # -n  helps in avoiding newline  
num=20  
for (( i=1 ; i<=8 ; i++ ))  
do  
echo -n "$num " #print current number  
newi=`expr $i % 2`  
  
#perform operations alternatively  
if [ $newi -eq 1 ]  
then  
num=`expr $num + 2`  
else  
num=`expr $num - 4`  
fi  
done

**Output**



**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

1. **Append content to a file provided file name exists and it has write permission.**

#Experiment 02: Program 15: To append a content to file whose name is read from user.

#Adnan Ismail Shah Muzavor

echo -n "Enter the file name: "

read fname

#if file exists

if [ -f $fname ]

then

#and has write permisisions

if [ -w $fname ]

then

echo "Enter content to be appended: "

cat>>$fname

echo ""

echo "Content of $fname is: `cat $fname`"

#file has no write permisions

else

echo "$fname exists, but has no write permisisions"

fi

#file doesnt exists

else

echo "No file with name $fname"

fi

**Output**

student@compcnc17-optiPlex-3020:~/20co05$ bash exp215.sh

Enter the file name: hello.txt

No file with name hello.txt

student@compcnc17-optiPlex-3020:~/20co05$ cat > addme.txt

hello i am file student@compcnc17-optiPlex-3020:~/20co05$ bash exp215.sh

Enter the file name: addme.txt

Enter content to be appended:

I am appending this to file

Content of addme.txt is: hello i am file I am appending this to file

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

1. **Generate the series: 1,3,2,4,3,5,4,6,.....**

#Experiment 02, program 16:Print the given series (1,3,2,4,3,5,4,6.....)

#Adnan Ismail Shah Muzavor.

echo -n "Enter the numbe rof elements you want to have in series: "  
read nval  
echo -n "Series is:  " # -n  helps in avoiding newline  
  
num=1  
  
for (( i=1 ; i<=$nval ; i++ ))  
do  
echo -n "$num " #print current number  
newi=`expr $i % 2`  
#perform operations alternatively  
if [ $newi -eq 1 ]  
then  
num=`expr $num + 2`  
else  
num=`expr $num - 1`  
fi  
done

**Output**

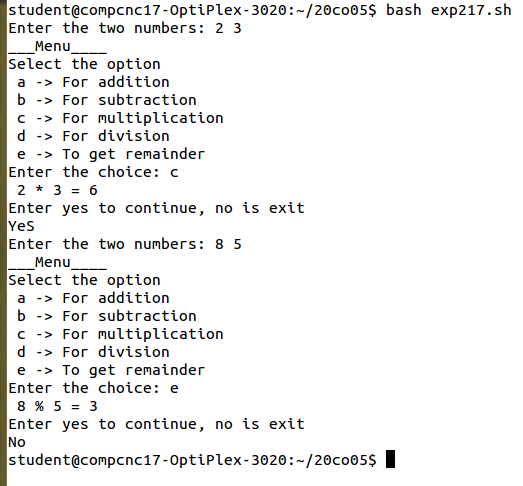


**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

1. **Write a shell script to prompt user to enter 2 numbers and perform various arithmetic operations using switch statement.**

#Experiment 02, program 17:To take two numbers as input from user and perform arithmatic operations chosen by the user  
#Adnan Ismail Shah Muzavor.  
cont=1  
while [ $cont -eq 1 ]  
do  
echo -n "Enter the two numbers: "  
read num1 num2  
echo "\_\_\_Menu\_\_\_\_"  
echo "Select the option"  
echo " a -> For addition"  
echo " b -> For subtraction"  
echo " c -> For multiplication"  
echo " d -> For division"  
echo " e -> To get remainder"  
echo -n "Enter the choice: "  
read ch  
case "$ch" in  
'a') echo " $num1 + $num2 = ` expr $num1 + $num2`" ;;  
'b') echo " $num1 - $num2 = ` expr $num1 - $num2`" ;;  
'c') echo " $num1 \* $num2 = ` expr $num1 \\* $num2`" ;;  
'd') echo " $num1 / $num2 = ` expr $num1 / $num2`" ;;  
'e') echo " $num1 % $num2 = ` expr $num1 % $num2`" ;;  
  \*) echo "Invalid option chosen"  
esac  
  
echo "Enter yes to continue, no is exit"  
read ch2  
case "$ch2" in  
[Yy][Ee][Ss]) cont=1 ;;  
[Nn][Oo]) cont=0 ;;  
        \*) echo "Invalid choice, so choosing to continue";;  
esac  
done

**Output**



**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

1. **To sort given numbers in descending order.**

#Experiment 02: Program 18: To sort the array in descending order

#Adnan Ismail Shah Muzavor

echo -n "Enter the size of array: "

read n

echo "Enter the array elements: "

#take input pf array elemnts

#like c,c++ here also it starts from 0

for (( i=0 ; i<$n ; i++ ))

do

read arr[$i]

done

#display elemnts before sorting in descending order

echo -n "Array elemnts before sorting are: "

for (( i=0 ; i<$n ; i++ ))

do

echo -n "${arr[$i]} "

done

#sort the array using bubble sort (using \_ opposite \_ condition)

for (( i=0 ; i<$n ; i++ ))

do

for (( j=0 ; j<$n-1-$i ; j++ ))

do

#we have used ooposite condition here to get result

#in descending order

if [ ${arr[$j]} -le ${arr[$j+1]} ]

then

temp=${arr[$j]}

arr[$j]=${arr[$j+1]}

arr[$j+1]=$temp

fi

done

done

echo "" #leave a line

#output we wanted

echo -n "Array elemnts after sorting in descending order are: "

for (( i=0 ; i<$n ; i++ ))

do

echo -n "${arr[$i]} "

done

**Output**

student@compcnc17-optiPlex-3020:~/20co05$ bash filee.sh

Enter the size of array: 5

Enter the array elements:

44

65

0

11

23

Array elemnts before sorting are: 44 65 0 11 23

Array elemnts after sorting in descending order are: 65 44 23 11 0 student@compcnc17-optiPlex-3020:~/20co05$

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

19. Display the following patterns :

**\* \* \* \* \***

**\* \* \* \***

**\* \* \***

**\* \***

**\***

**1 2 3 4**

**2 3 4**

**3 4**

**4**

**1**

**2 2**

**3 3 3**

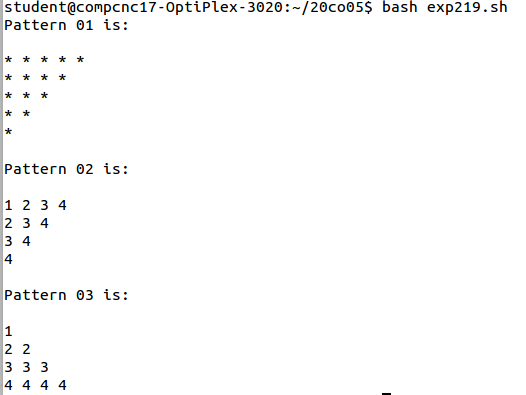
**4 4 4 4**

#Experiment 02, program 19:Print the following patterns

#Adnan Ismail Shah Muzavor.

echo "Pattern 01 is: "  
echo ""  
for(( i=5 ; i>=0 ; i-- ))  
do  
for(( j=0 ; j<i ; j++ ))  
do  
echo -n "\* "  
done  
echo ""  
done  
echo "Pattern 02 is: "  
echo ""  
jval=1 #a value from whcih each row has to be started  
       #first row start with 1 and go upto 4  
       #second row start with 2 and go upto 4  
       #third row start from 3 and go upto 4  
       #so on..  
         
for(( i=4 ; i>=0 ; i-- ))  #number of rows  
do  
for(( j=$jval ; j<=4 ; j++ )) #printing upto 4 at each row  
do  
echo -n "$j "  
done  
jval=`expr $jval + 1`  
echo ""  
done  
echo "Pattern 03 is: "  
echo ""  
for(( i=1 ; i<=4 ; i++ ))  
do  
for(( j=1 ; j<=i ; j++ )) #Print value of i, j times  
do  
echo -n "$i "  
done  
echo ""  
done

**Output**



**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**