 # PROGRAM 01 : Display greet message using the system time....
 # Programmer : Haysten D'costa...

```
#!/bin/bash

d=`date +"%H"` # following the 24 hour format....

if [ $d -ge 1 -a $d -lt 12 ]
then
   echo "Good morning Boss !"
elif [ $d -ge 12 -a $d -lt 16 ]
then
   echo "Good afternoon Boss !"
else
   echo "Good evening Boss !"
fi
```

OUTPUT :

```
student@cnc12-OptiPlex-3020:~/Desktop/21co56$ bash Program01.sh
Good evening Boss !
```

 # PROGRAM 02 : Find the factorial of a given number....
 # Programmer : Haysten D'costa

```
#!/bin/bash

fact=1
echo "Enter a number : " # prompt user to enter a number....
read num # reading input from user....

for(( i=num; i>0; i-- )) # while num > 0, keep reducing and multiply & store in fact....
do
        fact=`expr $fact \* $i`
done
echo "Factorial is : " $fact # displaying the factorial....
```

OUTPUT :

```
student@cnc12-OptiPlex-3020:~/Desktop/21co56$ bash Program02.sh
Enter a number : 5
Factorial is : 120
```

 # PROGRAM 03 : Generate the fibonacci series for the number of terms entered by the user....
 # Programmer : Haysten D'costa

```
#!/bin/bash

first_t=0  # 1st term....
second_t=1 # 2nd term....
echo "Enter number of terms : " # prompt user to enter no of terms in series....
read num # reading input from user....
echo "Fibonacii series is : "
for(( i=0; i<num; i++ ))
do
        echo $first_t" " # first display the 1st term....
        next_t=`expr $first_t + $second_t` # then compute the next term....
        first_t=$second_t # swap (to generete the new 1st and 2nd terms)....
        second_t=$next_t
done
echo
```

OUTPUT :

```
student@cnc12-OptiPlex-3020:~/Desktop/21co56$ bash Program03.sh
Enter number of terms :
5
Fibonacii series is :
0 1 1 2 3
```

 # PROGRAM 04 : To print the number table....
 # Programmer : Haysten D'costa

```
#!/bin/bash

limit=10 # range - upto which tables required....
echo "Enter a number : "
read num
echo "Tables of "$num" are : "
for(( i=0; i<=limit; i++ )) # displays upto 10 tables....
do
        echo " ->  "$num" x "$i = `expr $num \* $i` # displays in (num x i = result) format....
done
```

OUTPUT :

```
student@cnc12-OptiPlex-3020:~/Desktop/21co56$ bash Program04.sh
Enter a number : 10
Tables of 10 are :
 ->  10 x 0 = 0
 ->  10 x 1 = 10
 ->  10 x 2 = 20
 ->  10 x 3 = 30
 ->  10 x 4 = 40
 ->  10 x 5 = 50
 ->  10 x 6 = 60
 ->  10 x 7 = 70
 ->  10 x 8 = 80
 ->  10 x 9 = 90
 ->  10 x 10 = 100
```

 # <u>PROGRAM 05</u> : To display square of first 20 numbers....
 # <u>Programmer</u> : Haysten D'costa

```
#!/bin/bash

limit=20
echo "Displaying squares of first "$limit" numbers : "
for(( i=0; i<=limit; i++ ))
do
        echo "Square("$i") : " `expr $i \* $i`
done
```

<u>OUTPUT</u> :

```
student@cnc12-OptiPlex-3020:~/Desktop/21co56$ bash Program05.sh
Displaying squares of first 20 numbers :
Square(0) :  0
Square(1) :  1
Square(2) :  4
Square(3) :  9
Square(4) :  16
Square(5) :  25
Square(6) :  36
Square(7) :  49
Square(8) :  64
Square(9) :  81
Square(10) :  100
Square(11) :  121
Square(12) :  144
Square(13) :  169
Square(14) :  196
Square(15) :  225
Square(16) :  256
Square(17) :  289
Square(18) :  324
Square(19) :  361
Square(20) :  400
```

 # PROGRAM 06 : To print the largest of 3 numbers entered by the user....
 # Programmer : Haysten D'costa

```
#!/bin/bash

echo -n "Enter first number  : "; read num1 # reading 3 numbers....
echo -n "Enter second number : "; read num2
echo -n "Enter third number  : "; read num3

if [ $num1 -eq $num2 -a $num2 -eq $num3 ] # if all numbers are equal, exit....
then
        echo "All three numbers are equal...."
        exit
fi

if [ $num1 -gt $num2 ] # compare num1 and num2....
then
        if [ $num1 -gt $num3 ] # compare num1 and num3....
        then
                echo "First number is GREATEST !"
        else
                echo "Second number is GREATEST !"
        fi
else
        if [ $num2 -gt $num3 ] # compare num2 and num3....
        then
                echo "Second number is GREATEST !"
        else
                echo "Third number is GREATEST !"
        fi
fi
```

OUTPUT :

```
student@cnc12-OptiPlex-3020:~/Desktop/21co56$ bash Program06.sh
Enter first number  : 10
Enter second number : 0
Enter third number  : 5
First number is GREATEST !
student@cnc12-OptiPlex-3020:~/Desktop/21co56$ bash Program06.sh
Enter first number  : 1
Enter second number : 2
Enter third number  : 3
Third number is GREATEST !
student@cnc12-OptiPlex-3020:~/Desktop/21co56$ bash Program06.sh
Enter first number  : 5
Enter second number : 5
Enter third number  : 5
All three numbers are equal....
```

 # PROGRAM 07 : To reverse a number....
 # Programmer : Haysten D'costa

```
#!/bin/bash

echo -n "Enter a number : "
read num

echo # 'echo' is used to print a new line....
echo -n "Reversed number is : " # 'printf' can be used to display content without a new line....
while [ $num -gt 0 ] # while number greater than 0, remove and display the last digit....
do
   remainder=`expr $num % 10`
   num=`expr $num / 10`
   printf $remainder
done
echo # to display a new line…
```

OUTPUT :

```
student@cnc12-OptiPlex-3020:~/Desktop/21co56$ bash Program07.sh
Enter a number : 12345

Reversed number is : 54321
```

 # PROGRAM 08 : To reverse a string....
 # Programmer : Haysten D'costa

```
#!/bin/bash

echo -n "Enter a string : "
read string

len=${#string} # gives the length of the string....
for (( i=$len-1; i>=0; i-- )); do # from last to first char, store in reverse....
   reverse="$reverse${string:$i:1}"
done
echo $reverse
```

OUTPUT :

```
student@cnc12-OptiPlex-3020:~/Desktop/21co56$ bash Program08.sh
Enter a string : HAYSTEN DCOSTA
ATSOCD NETSYAH
```

 # PROGRAM 09 : To check whether entered number is even or odd....
 # Programmer : Haysten D'costa

```
#!/bin/bash

echo "Enter a number : "
read num

remainder=`expr $num % 2` # computing remainder....
if [ $remainder -eq 0 ] # if remainder is equal to 0, then even else odd....
then
   echo "Number is even !"
else
   echo "Number is odd !"
fi
```

OUTPUT :

```
student@cnc12-OptiPlex-3020:~/Desktop/21co56$ bash Program09.sh
Enter a number :
5
Number is odd !
student@cnc12-OptiPlex-3020:~/Desktop/21co56$ bash Program09.sh
Enter a number :
10
Number is even !
```

 # PROGRAM 10 : Check whether given year is leap year or not...
 # Programmer : Haysten D'costa

```
#!/bin/bash

year=`date +%Y` # gives the current year....

if [ `expr $year % 4` -eq 0 ] # if year is divisible by 4, leap year else not a leap year....
then
   echo "$year is a leap year."
else
   echo "$year is not a leap year."
fi
```

OUTPUT :

```
student@cnc12-OptiPlex-3020:~/Desktop/21co56$ bash Program10.sh
2023 is not a leap year.
```

 # PROGRAM 11 : To print all the prime numbers less than 20....
 # Programmer : Haysten D'costa

```
#!/bin/bash

limit=20
echo "Prime numbers are : "

for (( i=2; i<limit; i++ )); do
   is_prime=1
   for (( j=2; j<i; j++ )); do
      if [ `expr $i % $j` -eq 0 ]
      then
         is_prime=0
      fi
   done
   if [ $is_prime -eq 1 ]
   then
      echo -n $i" "
   fi
done
```

OUTPUT :

```
student@cnc12-OptiPlex-3020:~/Desktop/21co56$ bash Program11.sh
Prime numbers are :
2 3 5 7 11 13 17 19
```

 # PROGRAM 12 : To check if the entered string is palindromic or not....
 # Programmer : Haysten D'costa

```
#!/bin/bash

echo -n "Enter a string : "
read string

len=${#string} # gives the length of the string....
for (( i=$len-1; i>=0; i-- )); do # from last to first char, store in reverse....
   reverse="$reverse${string:$i:1}"
done

if [ $reverse == $string ]
then
   echo "String is a palindrome"
else
   echo "String not a palindrome"
fi
```

OUTPUT :

```
student@cnc12-OptiPlex-3020:~/Desktop/21co56$ bash Program12.sh
Enter a string : MOM
String is a palindrome
student@cnc12-OptiPlex-3020:~/Desktop/21co56$ bash Program12.sh
Enter a string : MOMMY
String not a palindrome
```

 # PROGRAM 13 : Using switch case, print the capitals given the state name....
 # Programmer : Haysten D'costa

```
#!/bin/bash

echo "Enter a state : "; read state

case "$state" in
   "Goa") echo "Panaji";;
   "Uttar Pradesh") echo "Lucknow";;
   "Haryana") echo "Chandigarh";;
   "Andhra Pradesh") echo "Amaravati";;
   "Rajasthan") echo "Jaipur";;
   "Tamil Nadu") echo "Chennai";;
   *) echo "State not valid !"
esac
```

OUTPUT :

```
student@cnc12-OptiPlex-3020:~/Desktop/21co56$ bash Program13.sh
Enter a state :
Goa
Panaji
student@cnc12-OptiPlex-3020:~/Desktop/21co56$ bash Program13.sh
Enter a state :
AAA
State not valid !
```

 # PROGRAM 14 : Write shell script to generate following series : 20, 22, 18, 20, 16, 18, 14, 16....
 # Programmer : Haysten D'costa

```
#!/bin/bash

i=1
num=20
echo -n "Enter the no. of terms : " # reading the no. of terms in series....
read n
while [ $i -le `expr $n` ]
do
   printf $num" " # displaying the term....
   if [ `expr $i % 2` -ne 0 ]
   then
      num=`expr $num + 2`
   else
      num=`expr $num - 4`
   fi
   i=`expr $i + 1` # to increment i....
done
echo
```

OUTPUT :

```
student@cnc12-OptiPlex-3020:~/Desktop/21co56$ bash Program14.sh
Enter the no. of terms : 10
20 22 18 20 16 18 14 16 12 14
```

 # PROGRAM 15 : Append content to a file provided file name exists and it has write permission....
 # Programmer : Haysten D'costa

```
#!/bin/bash

echo -n "Enter the file name : "
read file

if [ -f "$file" ] # -f -> to check if file exists....
then
   echo
   echo -n "$file exist "
   if [ -w "$file" ] # -w to check if file has write permission....
   then
      echo "and has write permissions"

      echo # to leave a line....
      echo "Enter Contents to add to file : "
      read contents
      echo $contents >> $file

      echo
      echo "Contents added : "
      cat $file
   else
      echo " but does not have write permissions"
   fi
else
   echo "File Doesn't Exist"
fi
```

OUTPUT :

```
student@cnc12-OptiPlex-3020:~/Desktop/21co56$ bash Program15.sh
Enter the file name : test.txt

test.txt exist and has write permissions

Enter Contents to add to file :
This is a file, i am attending sample text to this file.........

Contents added :
Haysten D'costa
This is a file, i am attending sample text to this file.........
```

```
student@cnc12-OptiPlex-3020:~/Desktop/21co56$ bash Program15.sh
Enter the file name : no_such_file.txt
File Doesn't Exist
```

 # PROGRAM 16 : Generate the series : 1, 3, 2, 4, 3, 5, 4, 6....
 # Programmer : Haysten D'costa

```
#!/bin/bash

term=1
echo -n "Enter no. of terms in series : "; read n
echo "Series is : "
for (( i=1; i<=n; i++ ))
do
        echo -n $term" "
        if [ `expr $i % 2` -ne 0 ]
        then
        term=`expr $term + 2`
        else
        term=`expr $term - 1`
        fi
done
```

OUTPUT :

```
student@cnc12-OptiPlex-3020:~/Desktop/21co56$ bash Program16.sh
Enter no. of terms in series : 10
Series is :
1 3 2 4 3 5 4 6 5 7 student@cnc12-OptiPlex-3020:~/Desktop/21co56$
```

 # PROGRAM 17 : Write a shell script to prompt user to enter 2 numbers and perform various arithmetic operations using switch statement....
 # Programmer : Haysten D'costa

```
#!/bin/bash

echo "Enter two numbers : "
read num1
read num2

echo -n "Enter the operator : "
read operator

case "$operator" in
        "+") echo "Result : $(($num1 + $num2))";;
        "-") echo "Result : $(($num1 - $num2))";;
        "*") echo "Result : $(($num1 * $num2))";;
        "/") echo "Result : $(($num1 / $num2))";;
        "%") echo "Result : $(($num1 % $num2))";;
        *) echo "Invalid operator !"
esac
```

OUTPUT :

```
student@cnc12-OptiPlex-3020:~/Desktop/21co56$ bash Program17.sh
Enter two numbers :
5
5
Enter the operator : +
Result : 10
student@cnc12-OptiPlex-3020:~/Desktop/21co56$ bash Program17.sh
Enter two numbers :
5
5
Enter the operator : -
Result : 0
```

 # PROGRAM 18 : To sort given numbers in descending order....
 # Programmer : Haysten D'costa

```
#!/bin/bash

echo -n "Enter the array elements : "
read -ra array
size=${#array[@]}

#Before Sort
echo -n "Before Sort : "
for element in ${array[@]}; do
        echo -n $element ""
done

# Bubble Sort
for((i=0; i<size-1; i++)); do
        for((j=0; j<size-i-1; j++)); do
                if [ ${array[$j]} -lt ${array[$j + 1]} ]; then
                        temp=${array[$j]}
                        array[$j]=${array[$j+1]}
                        array[$j+1]=$temp
                fi
        done
done

#After Sort
echo -e "\nAfter Sort : "
for element in ${array[@]}; do
        echo -n $element ""
done
echo
```

OUTPUT :

```
student@cnc12-OptiPlex-3020:~/Desktop/21co56$ bash Program18.sh
Enter the array elements : 1 2 3 4 5
Before Sort : 1 2 3 4 5
After Sort :
5 4 3 2 1
```

 # PROGRAM 19(a) : To display the following pattern....
 # Programmer : Haysten D'costa

```
#!/bin/bash

rows=5
for (( i=1; i<=rows; i++ )); do
        for (( j=1; j<=rows-i+1; j++ )); do
                printf "* "
                done
                echo
done
```

OUTPUT :

```
student@cnc12-OptiPlex-3020:~/Desktop/21co56$ bash Program19-a.sh
* * * * *
* * * *
* * *
* *
*
```

 # PROGRAM 19(b) : To display the following pattern....
 # Programmer : Haysten D'costa

```
#!/bin/bash

n=4
for (( i=0; i<n; i++ )); do
        for (( j=i+1; j<=n; j++ )); do
                printf $j" "
                done
                echo
done
```

OUTPUT :

```
student@cnc12-OptiPlex-3020:~/Desktop/21co56$ bash Program19-b.sh
1 2 3 4
2 3 4
3 4
4
```

 # PROGRAM 19(c) : To display the following pattern....
 # Programmer : Haysten D'costa

```
#!/bin/bash

n=4
for (( i=1; i<=n; i++ )); do
        for (( j=1; j<=i; j++ )); do
        printf $i" "
        done
        echo
done
```

OUTPUT :

```
student@cnc12-OptiPlex-3020:~/Desktop/21co56$ bash Program19-c.sh
1
2 2
3 3 3
4 4 4 4
```