return max(List)

Driven code

a=10
b=14
c=12

Print(maximum(a, b, c))

output:

6

Python Assignment 2

write a python function to find the max of three numbers

```
def maximum(a, b, c):
 if (a>=b) and (a>=c):
   largest=a
elif (b>=a) and (b>=c):
    largest = b
else:
   largest=c
return largest
```

3. write a
number i
Ans:- nur
go tak
num=
prime
if nu
ch
for i
if (n
prin
Print
b
else:
Pr
if
is n
else:
Print

2. write a python program to reverse a string.

Ans:- txt =
Reverse the String "Hello world".

```
txt = "Hello world"[::-1]
print(txt)
```

3. write a python function to check whether the number is prime or not.

Ans:- num = 407

```
# To take input from the user
# num = int (input (" Enter a number: "))
#prime numbers are greater than 1
if num>1:
  # check for factors.
for i in range (2, num):
  if (num % i)==0:
  print(num, "is not a prime numbea")
  Print(i," times", num//i, "is",num)
  break
else:
  Print (num," is a prime number")
# if input number is less than or equal to 1, it
is not prime
else:
Print(num,"is not a prime number")
```

4. use try, except, else and finally block to whether the number is palindrome or not.
(Raise error when input is not proper).

A:-
```
num = int(input("Enter a number:"))
temp = num
rev = 0
while (num > 0):
    dig = num % 10
    rev = rev * 10 + dig
    num = num // 10
if (temp == rev):
    print("The number is palindrome!")
else:
    print("Not a palindrome!")
```

5. write a python function to find Sum of Squares of first n natural numbers.

```
# python 3 program to find sum of square
# of first n natural numbers.

# Return the Sum of square of first n
natural numbers.
def Square Sum(n):
# Iterate i from 1
# and n finding
```

```
# square of i
# add to Sum
Sm = 0
for i in rang
    Sm = Sm +
return Sm
# Driven prog
n = 4
print (square S
```

lly block to whether
not.
proper).
umber:"))
indrome!u")
!")
d Sum of Squares
of square

```
# square of i and
# add to Sum.
 Sm=0
 for i in range(1,n+1):
     Sm = Sm+(i*i)

 return Sm

# Driven program
 n=4
print (square Sum(n))
```