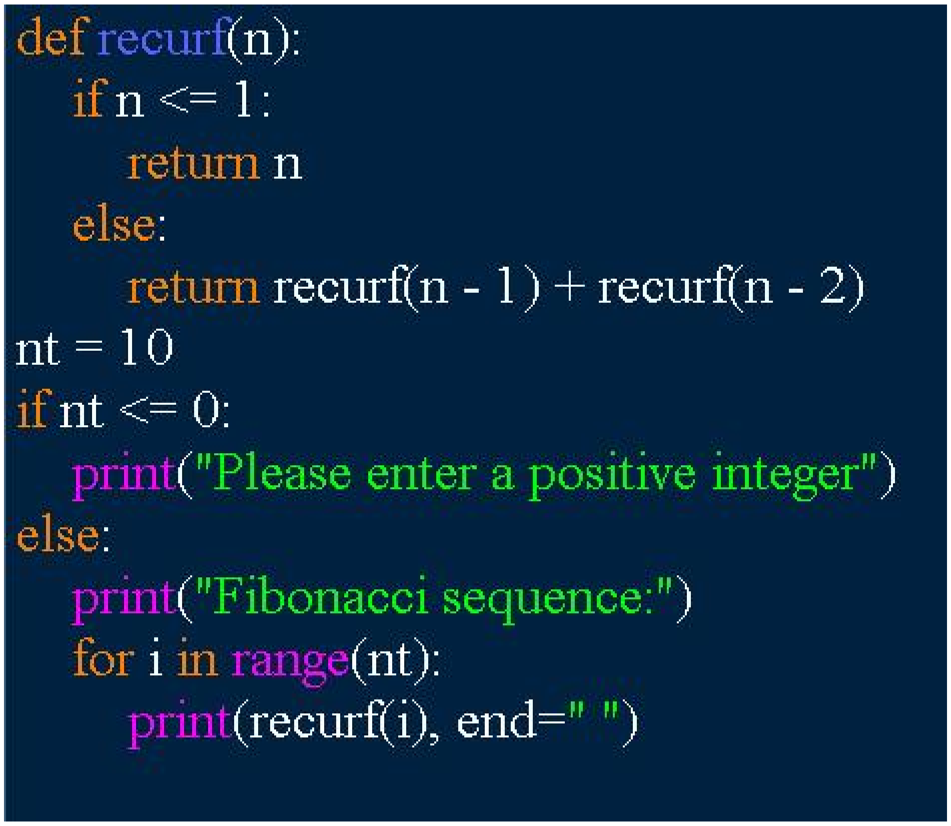
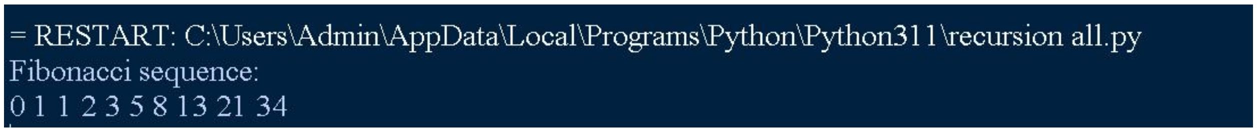
1.)Write a program to Print Fibonacci Series using recursion.

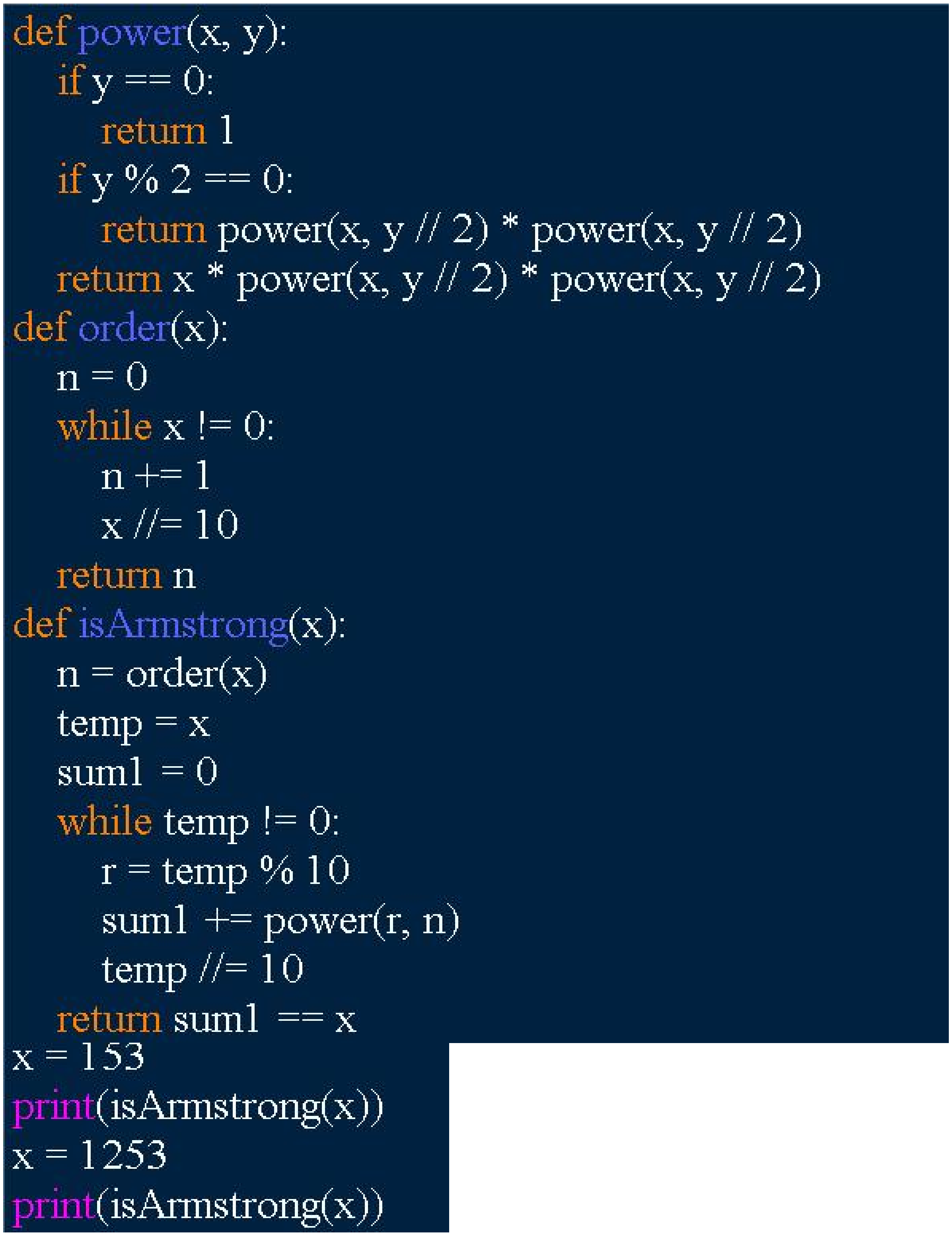


```
def recurf(n):
    if n <= 1:
        return n
    else:
        return recurf(n - 1) + recurf(n - 2)
nt = 10
if nt <= 0:
    print("Please enter a positive integer")
else:
    print("Fibonacci sequence:")
    for i in range(nt):
        print(recurf(i), end=" ")
```

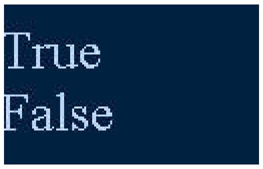


```
= RESTART: C:\Users\Admin\AppData\Local\Programs\Python\Python311\recursion all.py
Fibonacci sequence:
0 1 1 2 3 5 8 13 21 34
```

2.) Write a program to check the given no is Armstrong or not using recursive function.

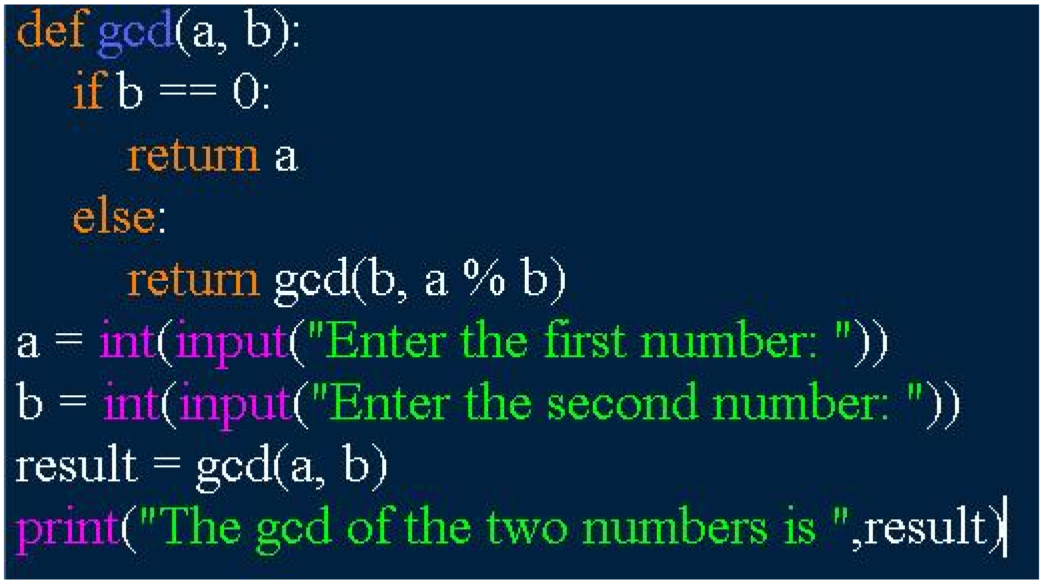


```
def power(x, y):
    if y == 0:
        return 1
    if y % 2 == 0:
        return power(x, y // 2) * power(x, y // 2)
    return x * power(x, y // 2) * power(x, y // 2)
def order(x):
    n = 0
    while x != 0:
        n += 1
        x //= 10
    return n
def isArmstrong(x):
    n = order(x)
    temp = x
    sum1 = 0
    while temp != 0:
        r = temp % 10
        sum1 += power(r, n)
        temp //= 10
    return sum1 == x
x = 153
print(isArmstrong(x))
x = 1253
print(isArmstrong(x))
```

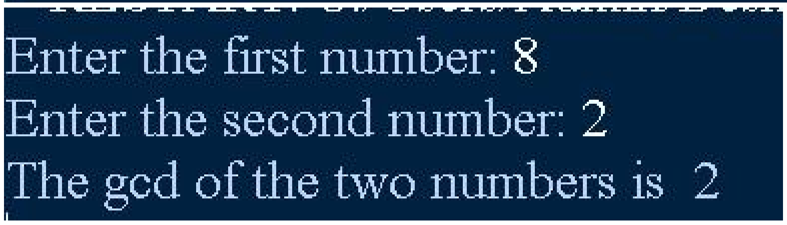


```
True
False
```

3.) GCD of two numbers using Recursion factorization

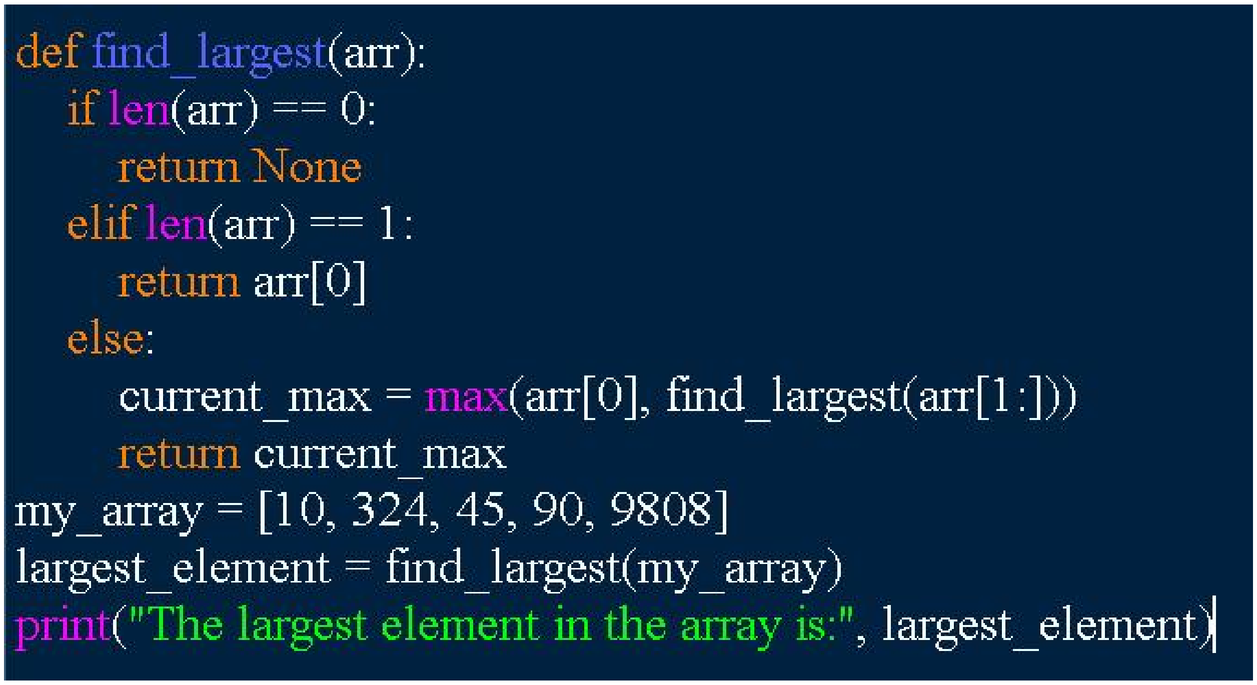


```python
def gcd(a, b):
    if b == 0:
        return a
    else:
        return gcd(b, a % b)
a = int(input("Enter the first number: "))
b = int(input("Enter the second number: "))
result = gcd(a, b)
print("The gcd of the two numbers is ",result)
```

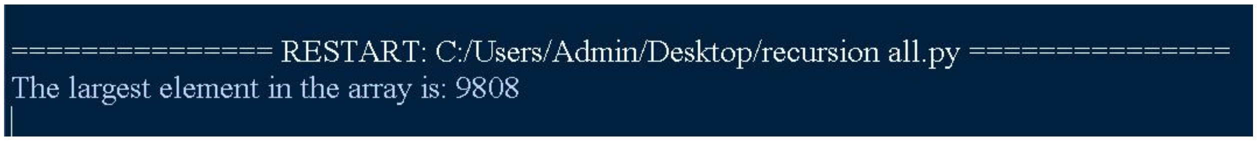


```
Enter the first number: 8
Enter the second number: 2
The gcd of the two numbers is  2
```

4.) finding largest number in an array using recursion.

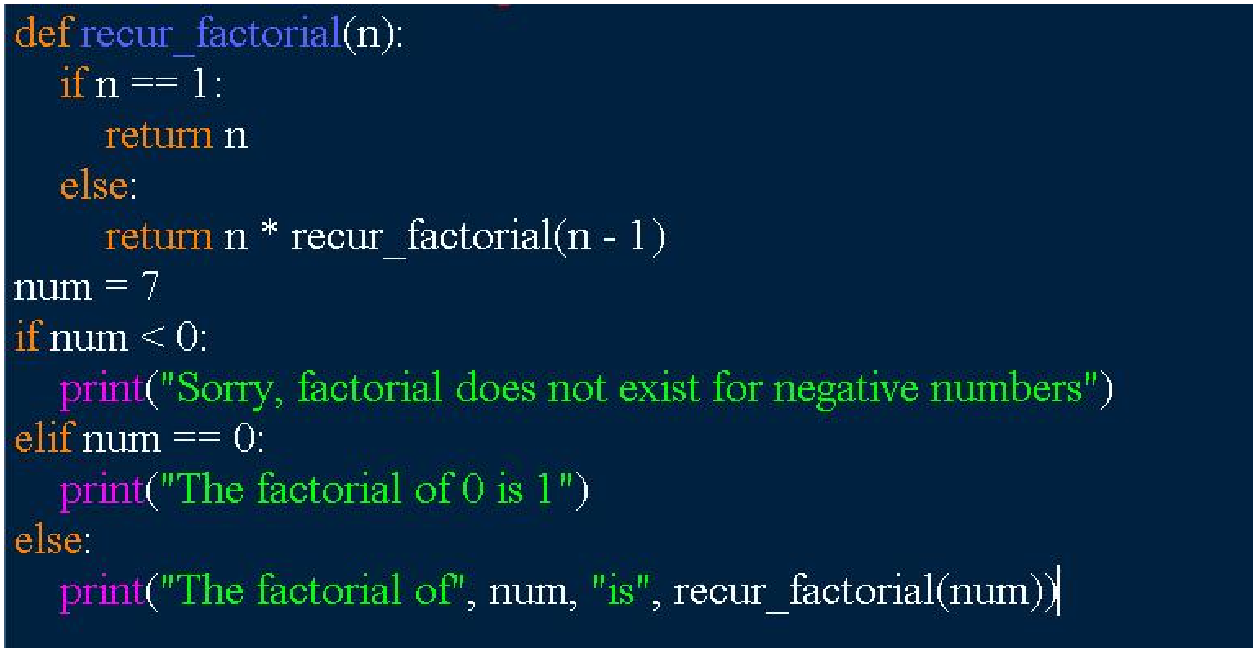


```python
def find_largest(arr):
    if len(arr) == 0:
        return None
    elif len(arr) == 1:
        return arr[0]
    else:
        current_max = max(arr[0], find_largest(arr[1:]))
        return current_max
my_array = [10, 324, 45, 90, 9808]
largest_element = find_largest(my_array)
print("The largest element in the array is:", largest_element)
```

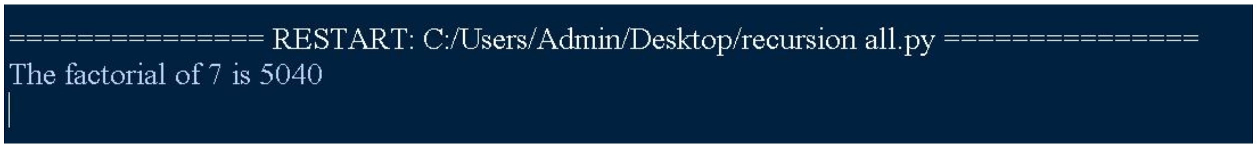


```
=============== RESTART: C:/Users/Admin/Desktop/recursion all.py ===============
The largest element in the array is: 9808
```

5.)factorial of a number using recursion

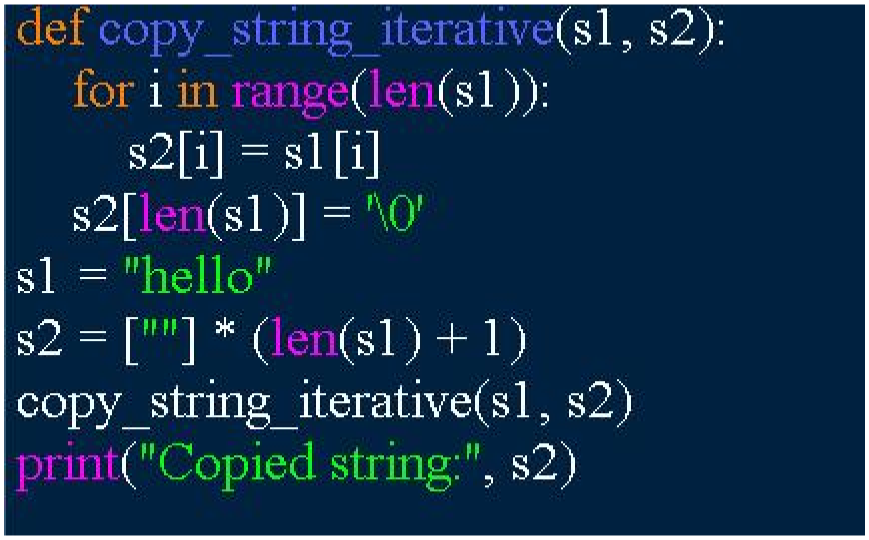


```python
def recur_factorial(n):
   if n == 1:
       return n
   else:
       return n * recur_factorial(n - 1)
num = 7
if num < 0:
   print("Sorry, factorial does not exist for negative numbers")
elif num == 0:
   print("The factorial of 0 is 1")
else:
   print("The factorial of", num, "is", recur_factorial(num))
```

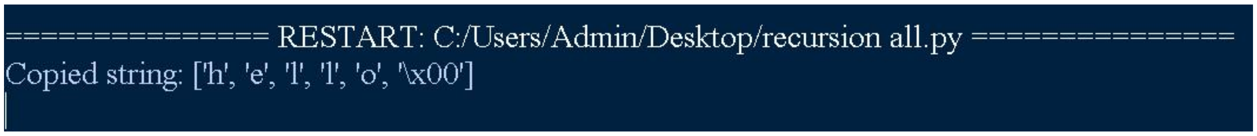


```
=============== RESTART: C:/Users/Admin/Desktop/recursion all.py ===============
The factorial of 7 is 5040
```

6.) Write a program for to copy one string to another using recursion

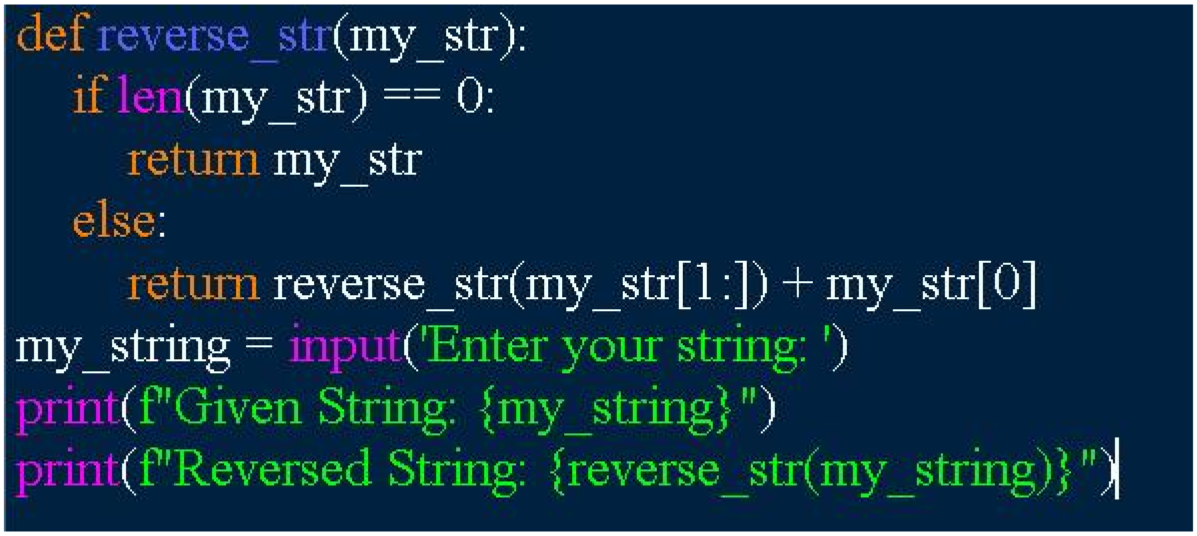


```
def copy_string_iterative(s1, s2):
    for i in range(len(s1)):
        s2[i] = s1[i]
    s2[len(s1)] = '\0'
s1 = "hello"
s2 = [""] * (len(s1) + 1)
copy_string_iterative(s1, s2)
print("Copied string:", s2)
```

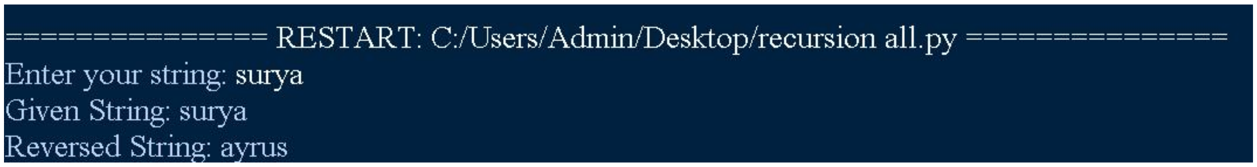


```
=============== RESTART: C:/Users/Admin/Desktop/recursion all.py ===============
Copied string: ['h', 'e', 'l', 'l', 'o', '\x00']
```

7.) Write a program   to print the reverse of a string using recursion

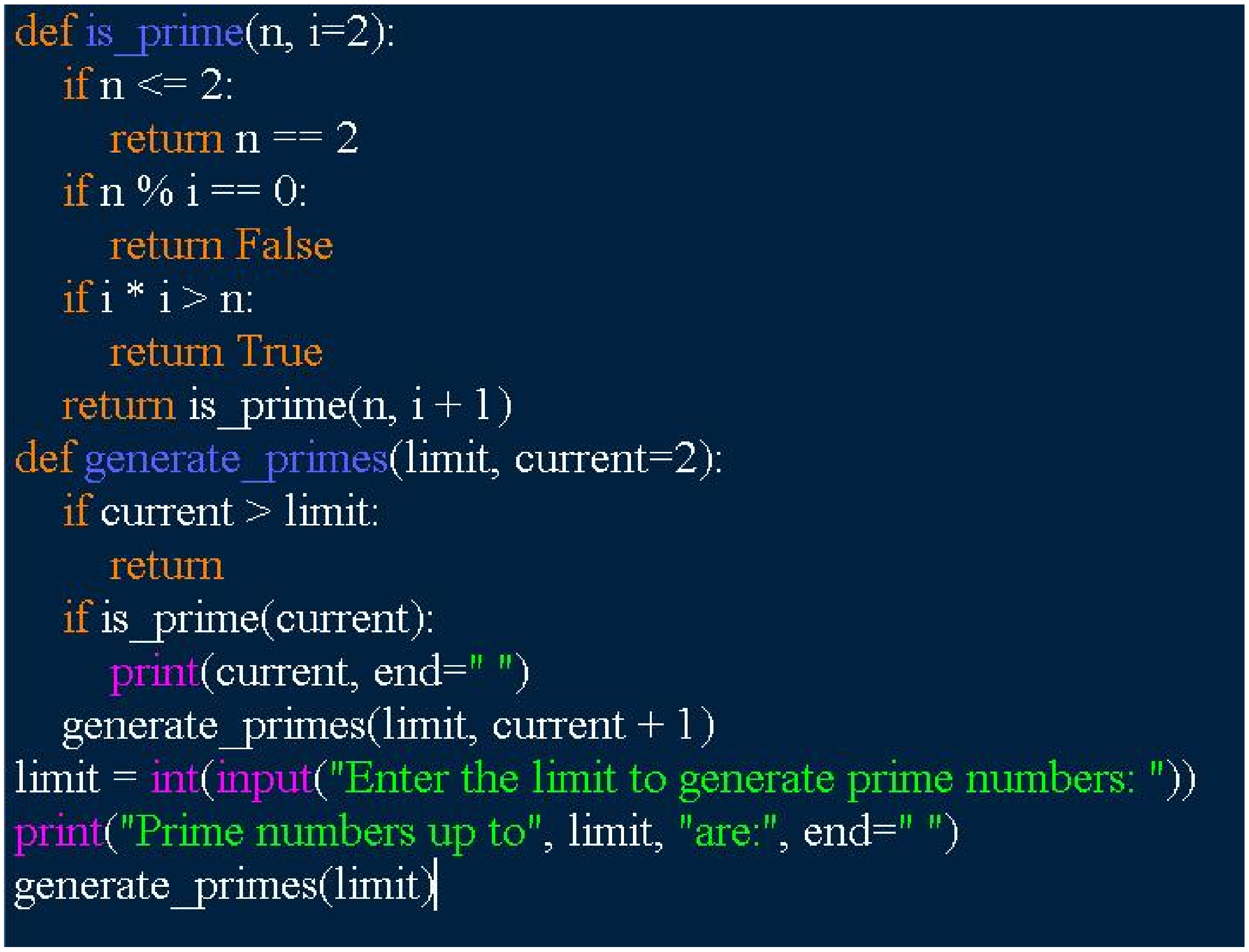


```
def reverse_str(my_str):
    if len(my_str) == 0:
        return my_str
    else:
        return reverse_str(my_str[1:]) + my_str[0]
my_string = input('Enter your string: ')
print(f"Given String: {my_string}")
print(f"Reversed String: {reverse_str(my_string)}")
```

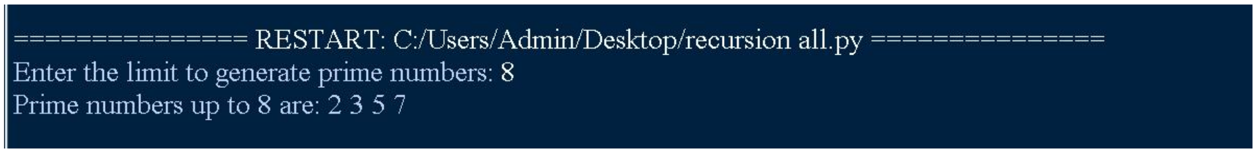


```
=============== RESTART: C:/Users/Admin/Desktop/recursion all.py ===============
Enter your string: surya
Given String: surya
Reversed String: ayrus
```

8.) Write a program   to generate all the prime numbers using recursion

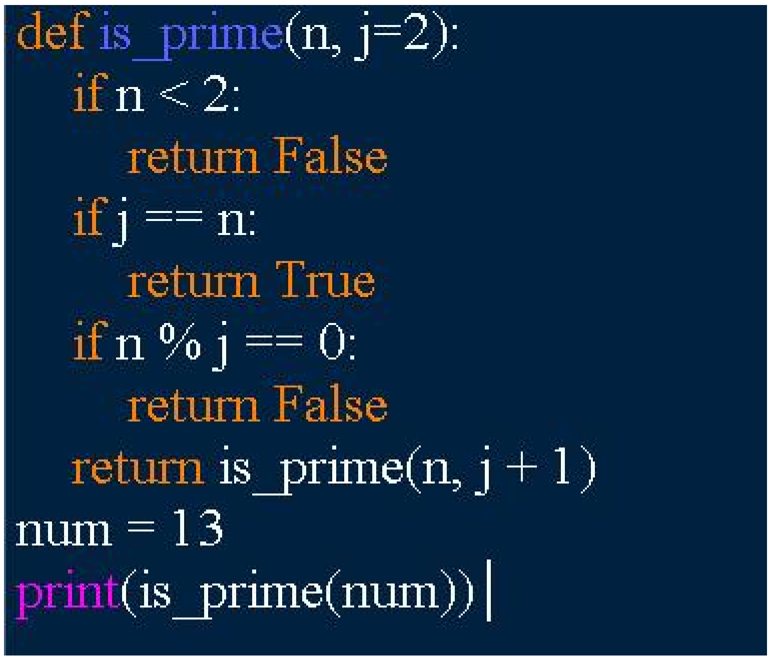


```
def is_prime(n, i=2):
    if n <= 2:
        return n == 2
    if n % i == 0:
        return False
    if i * i > n:
        return True
    return is_prime(n, i + 1)
def generate_primes(limit, current=2):
    if current > limit:
        return
    if is_prime(current):
        print(current, end=" ")
    generate_primes(limit, current + 1)
limit = int(input("Enter the limit to generate prime numbers: "))
print("Prime numbers up to", limit, "are:", end=" ")
generate_primes(limit)
```

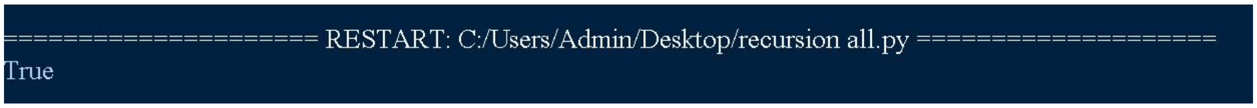


```
=============== RESTART: C:/Users/Admin/Desktop/recursion all.py ===============
Enter the limit to generate prime numbers: 8
Prime numbers up to 8 are: 2 3 5 7
```

9.) Write a program to check a number is a prime number or not using recursion.

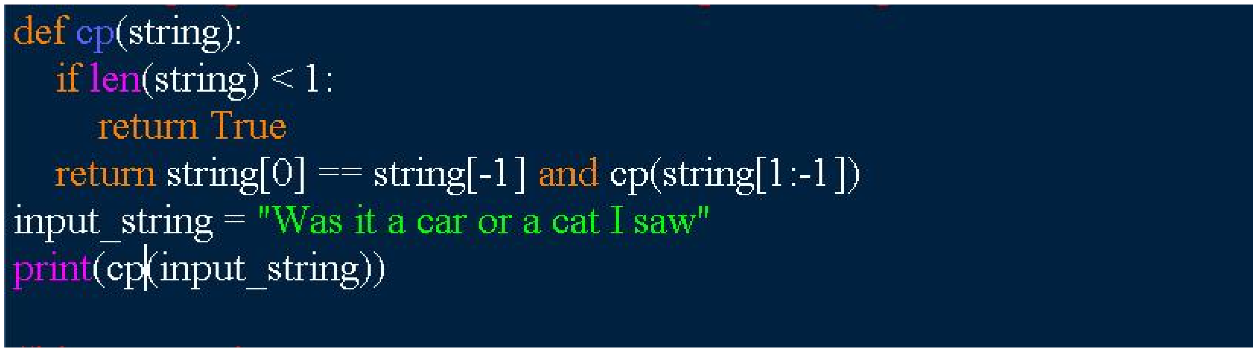


```python
def is_prime(n, j=2):
    if n < 2:
        return False
    if j == n:
        return True
    if n % j == 0:
        return False
    return is_prime(n, j + 1 )
num = 13
print(is_prime(num))
```

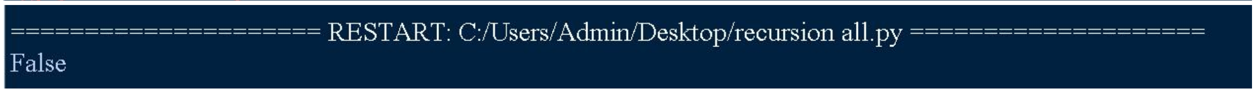


```
==================== RESTART: C:/Users/Admin/Desktop/recursion all.py ====================
True
```

10.) Write a program for to check whether a given String is Palindrome or not using recursion



```python
def cp(string):
    if len(string) < 1:
        return True
    return string[0] == string[-1] and cp(string[1:-1])
input_string = "Was it a car or a cat I saw"
print(cp(input_string))
```



```
==================== RESTART: C:/Users/Admin/Desktop/recursion all.py ====================
False
```