

In [27]: *# Program to find factorial of a number using recursion*

```
def factFor(n):
    assert n>=0, 'Number should be positive.'
    fact = 1
    for i in range(n):
        if i == 0:
            fact = 1
        else:
            fact = fact*(i+1)
    return fact

def factRecursion(n):
    assert n>=0, 'Number should be positive.'
    fact = 1
    if n == 0 or n == 1:
        fact = 1
    else:
        fact = n*factRecursion(n-1)
    return fact

num = int(input('Enter number to find factorial:'))

print('Using for loop:')
print(factFor(num))

print('Using recursion:')
print(factRecursion(num))
```

```
Enter number to find factorial:6
Using for loop:
720
Using recursion:
720
```

In [51]: *# Program to find nth Fibonacci Number*

```
def fiboFor(n):
    assert n>0, 'Number should be GT 0.'
    fiboSecondLast = 0
    fiboLast = 1

    if n == 1:
        fiboCurrent = fiboSecondLast
    elif n == 2:
        fiboCurrent = fiboLast
    for i in range(3,n+1):
        fiboCurrent = fiboSecondLast + fiboLast
        fiboSecondLast = fiboLast
        fiboLast = fiboCurrent
    return fiboCurrent

def fiboRecursion(n):
```

```

    assert n>0, 'Number should be GT 0.'
    if n == 1:
        return 0
    elif n == 2:
        return 1
    else:
        return fiboRecursion(n-1)+fiboRecursion(n-2)

num = int(input('Enter number to find Fibonacci number:'))

print('Using for loop:')
print(fiboFor(num))

print('Using recursion:')
print(fiboRecursion(num))

```

Enter number to find Fibonacci number:9
Using for loop:
21
Using recursion:
21

```

In [69]: # Program to reverse a string using recursion
         # Program to find nth Fibonacci Number

def strReverseFor1(inputString):
    reverseString = ''
    L = len(inputString)
    for i in range(L):
        reverseString = reverseString + inputString[L-i-1]
    return reverseString

def strReverseFor2(inputString):
    reverseString = ''
    for i in inputString:
        reverseString = i + reverseString
    return reverseString

def strReverseRecursion(inputString):
    reverseString = ''
    if inputString == '':
        return ''
    else:
        return inputString[-1]+strReverseRecursion(inputString[:-1])

str00 = input('Enter string to reverse:')

print('Using for loop:')
print(strReverseFor1(str00))
print(strReverseFor2(str00))

print('Using recursion:')
print(strReverseRecursion(str00))

```

```
Enter string to reverse:Gyana Ranjan Patra
Using for loop:
artaP najnaR anayG
artaP najnaR anayG
Using recursion:
artaP najnaR anayG
```

```
In [70]: # Program to find fibonacci number using a dictionary

Dict = {1:0, 2:1}

def fibDict(n):
    assert n>0
    if n not in Dict:
        Dict[n] = fibDict(n-1) + fibDict(n-2)
    return Dict[n]

num = int(input('Enter number to find Fibonacci number:'))

print(fibDict(num))
print(Dict)
```

```
Enter number to find Fibonacci number:6
5
{1: 0, 2: 1, 3: 1, 4: 2, 5: 3, 6: 5}
```

```
In [74]: # Program to find factorial of number using a dictionary

Dict = {0:1, 1:1}

def factDict(n):
    assert n>=0
    if n not in Dict:
        Dict[n] = factDict(n-1) * n
    return Dict[n]

num = int(input('Enter number to find factorial:'))

print(factDict(num))
print(Dict)
```

```
Enter number to find factorial:6
720
{0: 1, 1: 1, 2: 2, 3: 6, 4: 24, 5: 120, 6: 720}
```

```
In [1]: # program to find if a string is palindrome

def palindromeFor(inputStr):
    outputStr = ''
    inputStr = inputStr.lower()
    for i in inputStr:
        outputStr = i + outputStr
    return inputStr == outputStr

def palindromeRecursion(inputStr):
    if len(inputStr) <= 1:
        return True
```

```

        elif inputStr[0] == inputStr[-1] and palindromeRecursion(inputStr[1:-1])
            return True
        else:
            return False

str00 = 'yyeyy1'
print('Using for loop:')
print(palindromeFor(str00))

print('Using recursion:')
print(palindromeFor(str00))

```

Using for loop:
False
Using recursion:
False

Program to flatten a list

Psuedocode:

1. function flatten(lst1, lst2 = [])
2. for every element of lst1
3. if element is not a list
4. append in lst2
5. else
6. flatten(element, lst2)
7. end if
8. end for
9. end function

In [3]: *# Program to flatten a list*

```

def flatten(lst1, lst2=[]):
    for element in lst1:
        if type(element) != list:
            lst2.append(element)
        else:
            flatten(element, lst2)
    return lst2

lst1 = [1,2,[3,4,5,[6,7,8]],9]

print(flatten(lst1))

```

[1, 2, 3, 4, 5, 6, 7, 8, 9]