Palindrome.py Write a Python function that accepts a string \texttt{str} as argument, and makes a palindrome out of it by mirroring \texttt{str} and appending the mirrored string at the end of \texttt{str}. Test it on four different inputs.

Your submission will look like this:

```python
def Palindrome(s):
    return(s+s[::-1])
def run_Palindrome():
    Palindrome("abca13")
    Palindrome("(()()")
    Palindrome(""")
    Palindrome("z")
if __name__ == "__main__":
    run_Palindrome()
```

Prefix Closure:

```python
def prefixclosure(s):
    return { s[0:i:] for i in range(len(s)+1) }
def run_prefixclosure():
    prefixclosure(..your input 1..)
    prefixclosure(..your input 2..)
    prefixclosure(..your input 3..)
    prefixclosure(..your input 4..)
if __name__ == "__main__":
    run_prefixclosure()```

FacList.py Write a function that, given \texttt{N}, generates the list of factorials of all numbers from 1 to \texttt{N}. For example, given 5, your function must return

Your submission will look like this:

```python
def fac(N):
    return(reduce(lambda x,y: x*y, [i for i in range(1,N+1)]))
def facList(N):
    return [fac(j) for j in range(1,N+1)]```