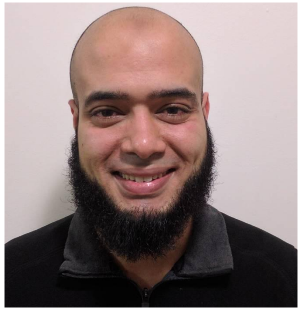
Python Programming

Recursive Functions

Homework 2

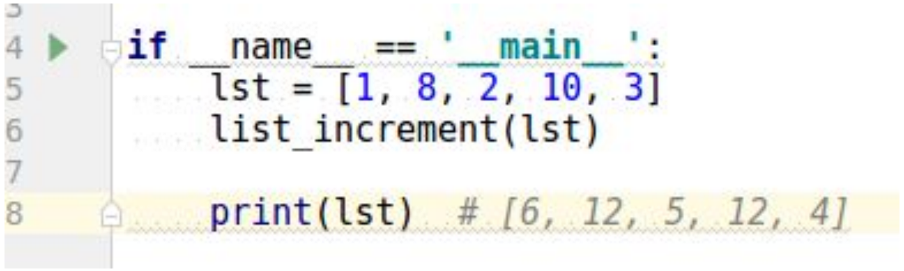
Mostafa S. Ibrahim
Teaching, Training and Coaching since more than a decade!

Artificial Intelligence & Computer Vision Researcher
PhD from Simon Fraser University - Canada
Bachelor / Msc from Cairo University - Egypt
Ex-(Software Engineer / ICPC World Finalist)



Problem #1: List Increment v2

- This is the same task as in homework 1, but this time the change is inplace
    - Hint: consider some default argument

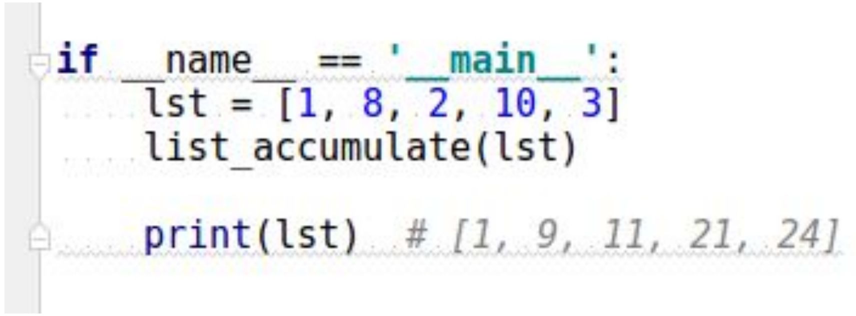


```
if __name__ == '__main__':
    lst = [1, 8, 2, 10, 3]
    list_increment(lst)

    print(lst)  # [6, 12, 5, 12, 4]
```

Problem #2: List Accumulation v2

- This is the same task as in homework 1, but this time the change is inplace

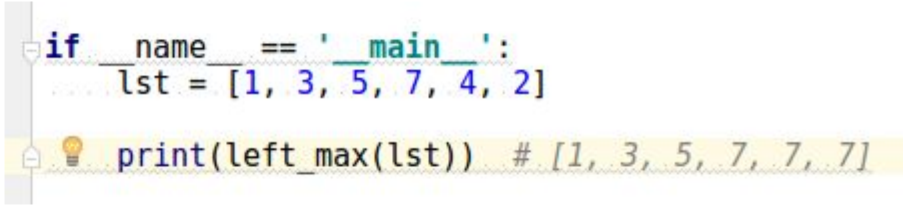


```
if __name__ == '__main__':
    lst = [1, 8, 2, 10, 3]
    list_accumulate(lst)

    print(lst)  # [1, 9, 11, 21, 24]
```

Problem #3: Left-Max

- Given list of numbers, return a list where each element at position i to be the maximum of numbers from 0 to index i
- E.g. input 1 3 5 7 4 2 ⇒ [1, 3, 5, 7, 7, 7]

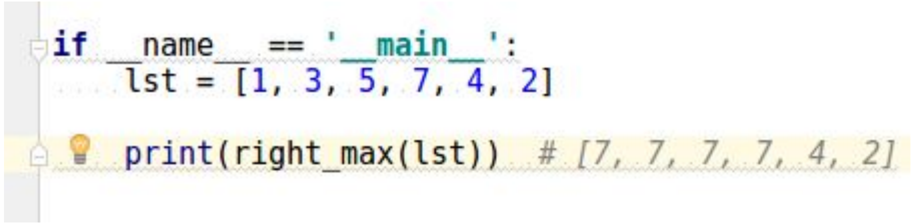


```
if __name__ == '__main__':
    lst = [1, 3, 5, 7, 4, 2]

    print(left_max(lst))  # [1, 3, 5, 7, 7, 7]
```

Problem #4: Right-Max

- Given list of numbers, return a list where each element at position i to be the maximum of numbers from index i to end of the list
- E.g. input 1 3 5 7 4 2 ⇒ [7, 7, 7, 7, 4, 2]

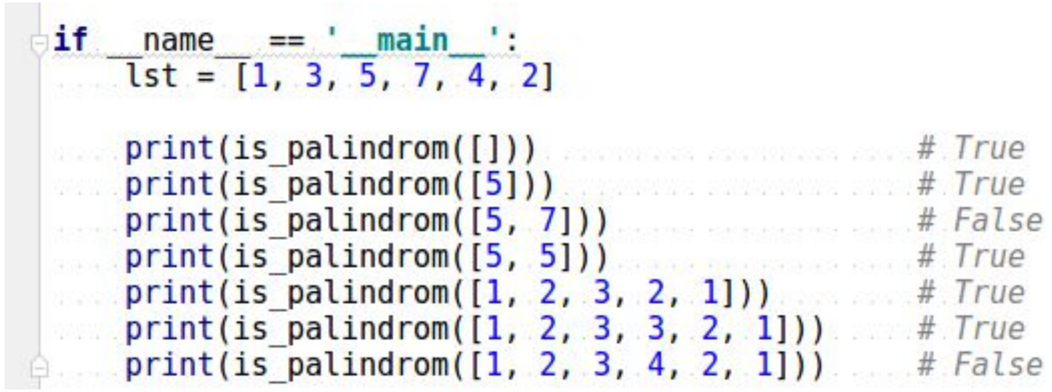


```
if __name__ == '__main__':
    lst = [1, 3, 5, 7, 4, 2]

    print(right_max(lst))  # [7, 7, 7, 7, 4, 2]
```

Problem #5: Is Palindrome

- Given a list of items, check recursively if it is a palindrome or not
    - We can read it the same from both directions

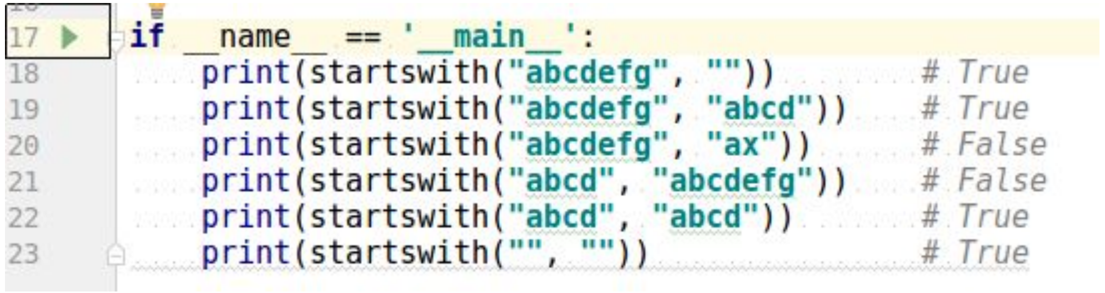


```python
if __name__ == '__main__':
    lst = [1, 3, 5, 7, 4, 2]

    print(is_palindrom([]))                          # True
    print(is_palindrom([5]))                         # True
    print(is_palindrom([5, 7]))                      # False
    print(is_palindrom([5, 5]))                      # True
    print(is_palindrom([1, 2, 3, 2, 1]))             # True
    print(is_palindrom([1, 2, 3, 3, 2, 1]))          # True
    print(is_palindrom([1, 2, 3, 4, 2, 1]))          # False
```

Problem #6: startswith

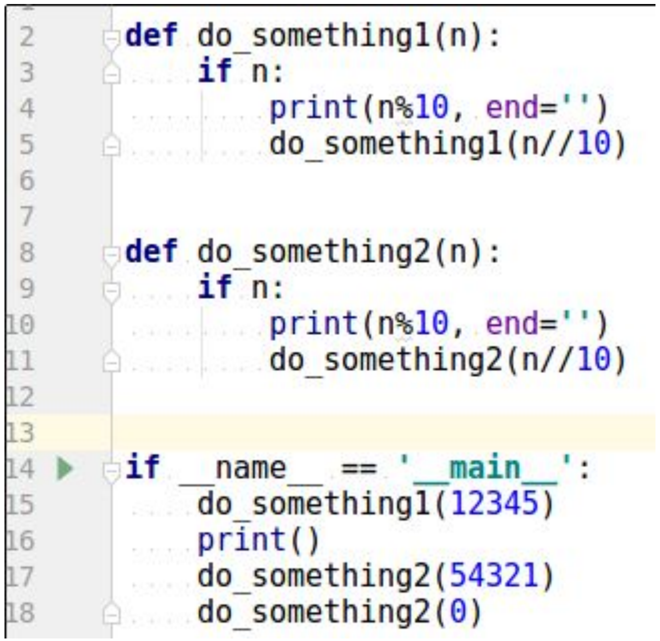
- The startswith() function returns True if a string starts with the specified prefix(string). If not, it returns False.



```
if __name__ == '__main__':
    print(startswith("abcdefg", ""))          # True
    print(startswith("abcdefg", "abcd"))      # True
    print(startswith("abcdefg", "ax"))        # False
    print(startswith("abcd", "abcdefg"))      # False
    print(startswith("abcd", "abcd"))         # True
    print(startswith("", ""))                 # True
```

Problem #7: Trace

- Without running code on the right
- Guess the output
- What are these methods doing



```
def do_something1(n):
    if n:
        print(n%10, end='')
        do_something1(n//10)


def do_something2(n):
    if n:
        print(n%10, end='')
        do_something2(n//10)


if __name__ == '__main__':
    do_something1(12345)
    print()
    do_something2(54321)
    do_something2(0)
```

"Acquire knowledge and impart it to the people."

"Seek knowledge from the Cradle to the Grave."