Module 4 Assignment 1

January 20, 2018

In [3]:
```
# Question 1
# find the square of given list

num_list = [4,6,8,12,45,78]
# print the square of num_list

sq_list = [i**2 for i in num_list]
print(sq_list)
```

```
[16, 36, 64, 144, 2025, 6084]
```

In [39]:
```
# Question 2
# replace all the elements in even indexes with squares and odd indexes with cubes

num_list = list(range(20))
print(num_list)
idx = 0
for i in num_list:
    if idx % 2:
        num_list[idx] = i**3
    else:
        num_list[idx] = i**2
    idx += 1
print(num_list)
```

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19]
[0, 1, 4, 27, 16, 125, 36, 343, 64, 729, 100, 1331, 144, 2197, 196, 3375, 256, 4913, 324, 6859]
```

In [48]:
```
# Question 3
# Using negative indexing, reverse the given list

num_list = list(range(20))
print(num_list)
for i in num_list[::-1]:
    print(i)
```

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19]
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
0
```

```
In [51]: # Quesiton 4
         # Given a tuple, convert it to a list and remove the last element using remove()

         num_tup = (1,2,3,4,5,6)
         # output should look like [1,2,3,4,5]
         num_list = list(num_tup)
         print(num_list)
         num_list.remove(len(num_list))
         print(num_list)
```

```
[1, 2, 3, 4, 5, 6]
[1, 2, 3, 4, 5]
```

```
In [54]: # Question 5
         # Given a list, find two sublists: list with even indexed elements and other with odd
         # return a new list which is formed after concatenating the even indexed list with od

         # Ex: given [1,2,3,4,5], the output should be: [1,3,5,2,4]
         num_list = list(range(1,6))
         # num_list = [1,2,3,4,5]
         evens = []
         odds = []

         for n in num_list:
```

```
            if num_list.index(n) % 2:
                evens.append(n)
            else:
                odds.append(n)

        print(odds + evens)
```

```
[1, 3, 5, 2, 4]
```

In [59]:
```
# Question 6
# Using for-loop, print the remainder of all the elements in a given list when divide

num_list = list(range(10))
remainders = []
print(num_list)
for n in num_list:
    remainders.append(n % 3)
print(remainders)
```

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
[0, 1, 2, 0, 1, 2, 0, 1, 2, 0]
```

In [64]:
```
# Question 7
# You are designer for the new login system for you class project.
# To save the usernames and passwords, you want to use a dictionary as the data struc
# Username will be the key and the value will be the password.
# For simplicity lets assume we store plaintext passwords
# write program which asks a new user for username and password
# call a function called register_user which prints "Registration complete!" and
# returns True after successfully inserting the username and password key-value pair
# If the selected username already exists, the function should print "Username alread
# Based on the return value the main program should decide to exit or ask for new use

def register_user(login_dict, uname, passwd):
    if uname not in login_dict:
        login_dict[uname] = passwd
        print("Registration complete!")
        return True
    else:
        print("Username already exists.  Try again!")
        return False

login_dict = {'jdoe':'23!@56'}

# hint: use of break statement is required
while True:
    uname = input("Enter user name: ")
```

```
            passwd = input("Enter password: ")

            if register_user(login_dict, uname, passwd):
                break
            else: # This else statement may not really be needed.  If register_user is false
                pass
```

```
Enter user name: jdoe
Enter password: 23!@56
Username already exists.  Try again!
Enter user name: jsmith
Enter password: pass
Registration complete!
```

```
In [73]: # Question 8
         # Create a generator function to return numbers from given number 'n' in decreasing o
         # if n=20, generate numbers from 20 to 0

         def reverse_num_generator(n):

             while n >= 0 :
                 yield n
                 n = n - 1

         # print numbers from 10 to 0 (n=10)
         n = 10
         rev = reverse_num_generator(n)

         for num in rev:
             print(num)

         print("-" * 30)

         n = 20
         rev = reverse_num_generator(n)

         for num in rev:
             print(num)
```

```
10
9
8
7
6
5
4
3
```

```
2
1
0
------------------------------
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
0
```