```
#task 1
def generic(func, test_list):
    for i in range(len(test_list)):
        for j in range(i+1, len(test_list)):
            if func(test_list[i], test_list[j]) > 0:
                test_list[i], test_list[j] = test_list[j], test_list[i]
    return test_list

#1.1
test_list = [5, 2, 12, 4, 9, 13, 22, 1, 6, 17]
sorted = generic(lambda x, y: y - x, test_list)
print("answer 1.1 is: ",sorted)

#1.2
test_list = ["Kate", "Sam", "Kate", "Jolly", "Alp", "Beta", "Alpine", "Samuel", "Bob", "Joy"]
sorted = generic(lambda x, y: ord(x[0])-ord(y[0]), test_list)
print("answer 1.2 is: ",sorted)


#1.3
test_list = [("Kate", 3), ("Sam", 2), ("Kate", 5), ("Jolly", 1), ("Alp", 2), ("Beta", 3), ("Alp", 1),
("Alpine", 2), ("Sam", 4), ("Bob", 2), ("Sam", 3)]
sorted = generic(lambda x, y: x[1] - y[1] if x[0] == y[0] else ord(x[0][0]) - ord(y[0][0]),
test_list)
print("answer 1.3 is: ",sorted)

#task 2
from math import sqrt

def isPrime(n):
    if n <= 1:
        return False
    for fac in range(2, int(sqrt(n))+1):
        if n % fac == 0:
            return False
    return True

def main():
    L = [i for i in range(1, 101)]
    print("L: ", L)

    doubled = list(map(lambda x: x * 2, L))
    print("answer 2.1, Doubled: ", doubled)

    squared_odd = [x**2 for x in L if x%2 != 0]
    print("answer 2.2, Squared odd: ", squared_odd)

    primes = list(filter(isPrime, L))
    print("answer 2.3, Primes: ", primes)

main()

"""
answer 1.1 is:  [22, 17, 13, 12, 9, 6, 5, 4, 2, 1]
answer 1.2 is:  ['Alp', 'Alpine', 'Beta', 'Bob', 'Jolly', 'Joy', 'Kate', 'Kate', 'Sam', 'Samuel']
answer 1.3 is:  [('Alp', 1), ('Alp', 2), ('Alpine', 2), ('Beta', 3), ('Bob', 2), ('Jolly', 1),
('Kate', 3), ('Kate', 5), ('Sam', 2), ('Sam', 3), ('Sam', 4)]

L:  [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26,
27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51,
52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76,
77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100]
answer 2.1, Doubled:  [2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40,
42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64, 66, 68, 70, 72, 74, 76, 78, 80, 82, 84, 86, 88, 90,
92, 94, 96, 98, 100, 102, 104, 106, 108, 110, 112, 114, 116, 118, 120, 122, 124, 126, 128, 130, 132,
134, 136, 138, 140, 142, 144, 146, 148, 150, 152, 154, 156, 158, 160, 162, 164, 166, 168, 170, 172,
174, 176, 178, 180, 182, 184, 186, 188, 190, 192, 194, 196, 198, 200]
answer 2.2, Squared odd:  [1, 9, 25, 49, 81, 121, 169, 225, 289, 361, 441, 529, 625, 729, 841, 961,
1089, 1225, 1369, 1521, 1681, 1849, 2025, 2209, 2401, 2601, 2809, 3025, 3249, 3481, 3721, 3969, 4225,
4489, 4761, 5041, 5329, 5625, 5929, 6241, 6561, 6889, 7225, 7569, 7921, 8281, 8649, 9025, 9409, 9801]
answer 2.3, Primes:  [2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73,
79, 83, 89, 97]
"""
```