**Pseudo code:**

**SET 1:**

1. Given a list of non-negative integers, arrange them in such a manner that they form the largest number possible.  
   The result is going to be very large, hence return the result in the form of a string.  
   Example:  
   Input:   
   largest number([1, 4, 5, 2, 3]) → should return “54321”  
   largest number([30, 34, 5, 3, 9]) → should return “9534330”  
   largest number([54, 546, 548, 60]) → should return “6054854654”

Code:

def largest\_no(lst1):

        val, ans = [ ], ""

        l = len(str(max(lst1)))+ 1

        for i in lst1:

                temp = str(i) \* l

                val.append((temp[:l:], i))

                val.sort(reverse = True)

        for i in val:

                ans += str(i[1])

        if int(ans)==0:

                return "0"

        return ans

# a = [1, 34, 3, 98, 9]

a = [54, 546, 548, 60]

print(largest\_no(a))

1. We want to make a package of goal kilos of chocolate. We have small bars and big bars. Write a function **make\_chocolate**(as mentioned below), such that, it returns the number of small bars to use, assuming we always use big bars before small bars. **make\_chocolate(big\_bar\_weight, small\_bar\_weight, goal)** Following are example inputs and corresponding expected output: make\_chocolate(5, 2, 9) → 2 make\_chocolate(5, 1, 10) → 0 make\_chocolate(4, 1, 7) → 3

Code:

def make\_chocolate(big\_bar\_weight, small\_bar\_weight, goal):

        small\_bars\_to\_use = (goal % big\_bar\_weight) // small\_bar\_weight

        return small\_bars\_to\_use

print(make\_chocolate(5, 2, 9))

print(make\_chocolate(5, 1, 10))

print(make\_chocolate(4, 1, 7))