Revision Class 4 (Dated: 23-Sep-2022)

1. Create class **Rope**. Rope has member's feet & inch. Inch must be less than 12. In case of a value greater than equal 12 for inch, add inch into feet to make the value of inches less than 12. Write following member functions:
   a. initialize function with two parameters, call set function inside
   b. set function with two parameters. Using integer division and remainder set value of feet and inch properly
   c. str function to return value of Rope with check value of feet and inches, the value can be zero for one or both members, return following strings accordingly:

      R1: Rope has 3 feet 5 inches
      R2: Rope has 6 feet **#in case of 0 inches**
      R3: Rope has 6 inches **#in case of 0 feet**
      R4: Rope has zero length **#in case both feet & inch are zero**

   d. write add function to add two rope objects into a new rope object and return new object. Again, you have to take care, if inch value exceeds 11
   e. write sub function to subtract two rope objects into a new rope object and return new object. Again, you have to take care, if inch value goes into negative. (We are assuming that rope 1 is greater than rope 2; however, it is possible that rope 2 has more inches than rope 1 than the subtraction will result into negative value; however, integer division and remainder operation can take care of that.

Finally, run following make to produce the given output:

```
import random as r
import roop


def main():
    r1 = roop.Rope(3, 5)
    r2 = roop.Rope(6, 7)
    r3 = roop.Rope(2, 9)
    print(f'R1: {r1}')
    print(f'R2: {r2}')
    print(f'R3: {r3}')
    print(f'R1+R2: {r1+r2}')
    print(f'R1+R3: {r1+r3}')
    print(f'R3+R2: {r3+r2}')
    print(f'R1-R3: {r1-r3}')
    print(f'R1-R1: {r1-r1}')
main()
```

```
R1: Rope has 3 feet 5 inches
R2: Rope has 6 feet 7 inches
R3: Rope has 2 feet 9 inches
R1+R2: Rope has 10 feet
R1+R3: Rope has 6 feet 2 inches
R3+R2: Rope has 9 feet 4 inches
R1-R3: Rope has 8 inches
R1-R1: Rope has zero length
```

2. Write class Bag. Bag may have multiple duplicate entries; however, we will keep a single copy in the list and we will maintain count of each element in another parallel list. Therefore, class has two data members of type list and following member functions:

a. initialization function to declare two empty lists

b. function to add element. We will check, if element already exist, we will increment the count, otherwise we will append element and 1 in the count

c. function to remove element. We will check, if element exists, and count is greater than 1, we will decrease count by 1. In case, count is 1, we will remove element from list and it's count as well

d. str function, check count of each element and add element count number of times like:

```
34  34  45  478  9  9  9  516  75  75  75  75  33
```