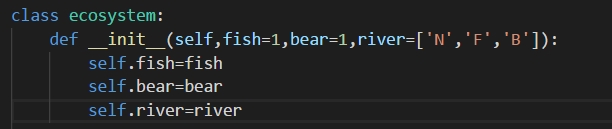
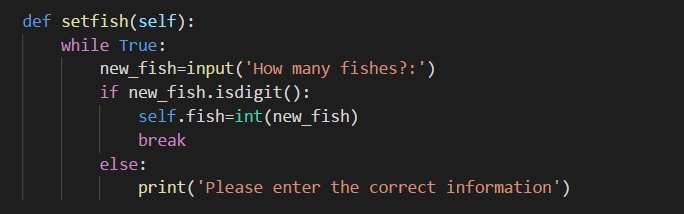
Q3:

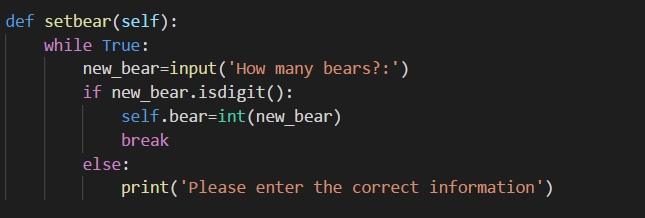
Step1: build a class ‘ecosystem’, containing instance ‘fish’, ‘bear’, and river, and set the default value for them.



Step2: ask the user to input the value of the number of the fish. The program can check whether the input is in correct format. If not, it will ask the user to input again.

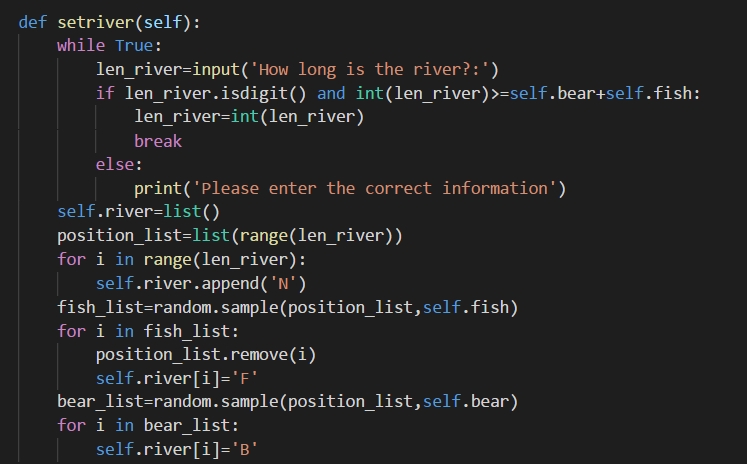


Step3: ask the user to input the value of the number of the bear. The program can check whether the input is in correct format. If not, it will ask the user to input again.

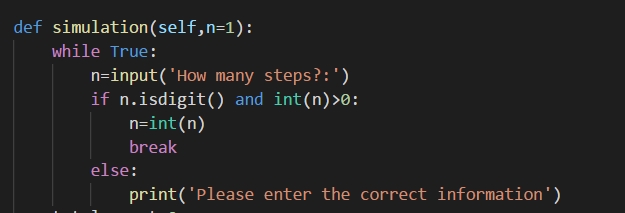


Step4: ask the ask the user to input the length of the number of the river. The program can check whether the input is in correct format. If not, it will ask the user to input again. What’s more, the program can automatically check that the length must be equal to or larger than the sum of the number of the fishes and bears.

After that, build a list meet the requirement randomly. self.river is equal to the list.



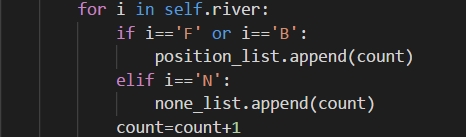
Step5: ask the user how many steps it want to execute.



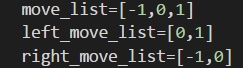
Step6: build a while loop, it will execute n times.



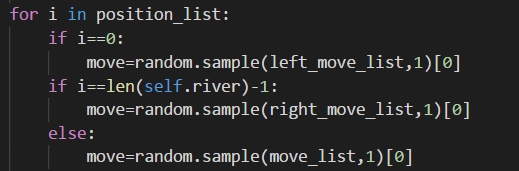
Step7: build 2 lists which recording the information of the position of ‘F’, ‘B’, and ‘N’.



Step8: build 3 list represents different possible movements of the item. The left\_move\_list is for the leftist item, the right\_move\_list is for the rightest item. move\_item is for the normal items.



Step9: randomly pick the possible movement according to the position of the fish or bear.



Step10: execute the movement from left to right one by one. Each if statement represent different situation. And it follow the requirement of the assignment: In each time step, based on a random process, each animal either attempts to move into an adjacent list location or stay where it is. If two animals of the same type are about to collide in the same cell, then they stay where they are, but they create a new instance of that type of animal, which is placed in a random empty (i.e., previously None) location in the list. If a bear and a fish collide, however, then the fish dies (i.e., it disappears).

if self.river[i]=='F' and not move==0:

if self.river[i+move]=='N':（鱼移动到空白处）

self.river[i]='N'

self.river[i+move]='F'

if self.river[i+move]=='B':（鱼移动到熊处，鱼死亡）

self.river[i]='N'

if self.river[i+move]=='F':（鱼移动到鱼处，生成新鱼）

if len(none\_list)>0:（有空白处才生成新鱼）

pick=random.sample(none\_list,1)[0]

self.river[pick]='F'

none\_list.remove(pick)

if self.river[i]=='B' and not move==0:

if self.river[i+move]=='N' or self.river[i+move]=='F':（熊移动到空白或鱼处）

self.river[i]='N'

self.river[i+move]='B'

if self.river[i+move]=='B':（熊有移动到熊处生成新熊）

if len(none\_list)>0: （有空白处才生成新熊）

pick=random.sample(none\_list,1)[0]

self.river[pick]='B'

none\_list.remove(pick)

Step11: let the user to begin:

a=ecosystem()

a.setfish()

a.setbear()

a.setriver()

print(a.river)

a.simulation()

print(a.river)