END

Items.delete\_item(“Dog”)  
print(val)

v

v

current\_item = self.tail  
 while current\_item:  
 val = current\_item.data  
 current\_item = current\_item.next

yield val

def delete\_item(self):

for i in reversed(items)

def insert\_sting\_middle(str, word):

return str[:2] + ”Elephant” + str[2:]

Class Node:

print(i)

Print.items  
print(\nTransverse the list in reverse order:”)

Items.append\_item(“Dog”)  
Items.append\_item(“Cat”)  
Items.append\_item(“Rabbit”)  
Items.append\_item(“Snake”)  
print(“Original list:”)  
items.print\_forward()  
print(“\nAppend item in front of the list:”)  
items.insert\_start(“Bird”)  
items.print\_forward

items.append\_item(“Dog”)  
items.append\_item(“Cat”)  
items.append\_item(“Rabbit”)  
items.append\_item(“Snake”)  
print(“Original list:”)  
items.print\_forward()

items = singly\_linked\_list()

items = singly\_linked\_list()