import ctypes

class DynamicArray(object):

def \_\_init\_\_(self):

self.n = 0

self.capacity = 1

self.A = self.make\_array(self.capacity)

def \_\_len\_\_(self):

return self.n

def \_\_getitem\_\_(self, k):

if not 0 <= k <self.n:

return IndexError('K is out of bounds !')

return self.A[k]

def append(self, ele):

if self.n == self.capacity:

self.\_resize(2 \* self.capacity)

self.A[self.n] = ele

self.n += 1

def insertAt(self,item,index):

if index<0 or index>self.n:

print("please enter appropriate index..")

return

if self.n==self.capacity:

self.\_resize(2\*self.capacity)

for i in range(self.n-1,index-1,-1):

self.A[i+1]=self.A[i]

self.A[index]=item

self.n+=1

def delete(self):

if self.n==0:

print("Array is empty deletion not Possible")

return

self.A[self.n-1]=0

self.n-=1

def removeAt(self,index):

if self.n==0:

print("Array is empty deletion not Possible")

return

if index<0 or index>=self.n:

return IndexError("Index out of bound....deletion not possible")

if index==self.n-1:

self.A[index]=0

self.n-=1

return

for i in range(index,self.n-1):

self.A[i]=self.A[i+1]

self.A[self.n-1]=0

self.n-=1

def \_resize(self, new\_cap):

B = self.make\_array(new\_cap)

for k in range(self.n):

B[k] = self.A[k]

self.A = B

self.capacity = new\_cap

def make\_array(self, new\_cap):

return (new\_cap \* ctypes.py\_object)()

………………………………………………………………………………………………………………………………………………………………………………

**Call function**   
# Instantiate

arr = DynamicArray()

# Append new element

arr.append(1)

len(arr)