2. Array insertion

```
from array import array

# before going into insertion, understand difference between
assigning a value to an array at some positon and inserting a
value into an array
arr=array('i',[1,2,3,4,5])
print("initial array-->",arr)
# let me assign a value to particular index
arr[2]=33
print("after assigning at index 2-->",arr)  #here we can see
that there is no change in other elements position. the old
value at that position is updated. that's it

# now let me INSERT a value into this array
arr.insert(2,333)
print("after INSERTING at Index 2-->",arr) #here we can see that
the new value is inserted at index 2, and the old value at index
2 and other values that are next to that index are moved one
place right.

# similarly let me INSERT a value at first index
arr.insert(0,11)
print("inserting the new value at the first index-->",arr) #here
evry value that is right to first index are moved right. so this
O(n) time complexity

#let me INSERT at last position
print("current length of array-->",len(arr))
arr.insert(7,777)
print("inserting the new value at the last index-->",arr) #here
no values are moved because we are inserting the new value at
the last index so no more values are in right side of that last
index. so time complexity is O(1)
```

OUTPUT:

initial array--> array('i', [1, 2, 3, 4, 5])
after assigning at index 2--> array('i', [1, 2, 33, 4, 5])

after INSERTING at Index 2--> array('i', [1, 2, 333, 33, 4, 5])
inserting the new value at the first index--> array('i', [11, 1, 2, 333, 33, 4, 5])
current length of array--> 7
inserting the new value at the last index--> array('i', [11, 1, 2, 333, 33, 4, 5, 777])