

在 Python 中，list 是一個基本且常用的資料結構，它的概念與 ADT list 幾乎相同，且因為可以直接取得某個特定元素的值，在大多時候甚至會被當成 python 中的陣列使用。Python list 可以允許放入各種不同資料型態的資料，無論是整數、文字，或是另一個物件，都可以被放入同一個 list 之中。在本題中，要請大家實作一個簡易版的 python list，並使用 python 的語法對 list 物件進行操作，而為了降低麻煩，我們只考慮在 list 中放入整數的情形。常見的 python list method 有以下幾個

1. `aList.append(elmnt)`: 將 `elmnt` 這個整數放入 `aList` 之中，並且放在最後面。
2. `aList.insert(pos, elmnt)`: 將 `elmnt` 這個整數放在 list 的第 `pos` 個 index 位置，如果 `pos` 大於或等於 list 的長度，請直接將 `elmnt` 放在 list 的最後。
3. `aList.remove(elmnt)`: 若 `elmnt` 有在 `aList` 之中出現，則將第一個出現的移除。
4. `aList.index(elmnt)`: 取得 `elmnt` 在 `aList` 之中第一次出現的 index。

5. `aList.count(elmnt)`: 取得 `elmnt` 在 `aList` 之中出現的次數。
6. `aList.extend(anotherList)`: 將 `anotherList` 接到 `aList` 之後。舉例來說，如果 `aList` 的內容是 `[1,2,3]`，而 `anotherList` 的內容是 `[5,6,7]`，則執行完此指令後，`aList` 的內容會變成 `[1,2,3,5,6,7]`，`anotherList` 則不受影響。
7. `aList.reverse()`: 將 `aList` 之中的元素順序全部顛倒。
8. `aList.sort()`: 依照大小順序將 `list` 之中的元素由小到大排好。

另外，python 中的 `list` 會以 `aList = list()` 或 `aList = []` 的方式宣告並初始化這個物件為空的 `list`。在本題的輸入資料中，我們一律使用前者作為初始化之指令。就如同 C++ 中的 `List* aList = new List()`，代表相同的意義。只是在 python 中回傳的不會是指標，而是一個空的物件。

輸入輸出格式

系統會提供一共 30 組測試資料，每組測試資料裝在一個檔案裡。在每個檔案中會有若干行，其中每一行包含一個 python 指令，python 的指令可能有上述的多種，此外對於以下的幾種，請根據對應的指令輸出應有的結果

1. `aList`: 請依序印出 `aList` 之中的內容，以逗號分隔其中的各個整數，並在前後印出中括號。因此若 `aList` 之中沒有內容，請直接印出 `[]`。
2. `aList.count(elmnt)`: 請計算 `aList` 之中 `elmnt` 出現的次數，並印出這個次數。
3. `aList.index(elmnt)`: 請印出 `elmnt` 在 `aList` 之中的 `index`，若此元素並沒有在 `aList` 中出現，請印出 `-1`。

對於每一個需要印出東西的指令，請於印出結果後換行。在本題中，以指令宣告的 `list` 物件不會超過 30 個，且作為物件名稱之字串不會超過 5 個字元，但指令的數量沒有上限。測試資料的內容分布如下表所示：

測試資料	append	insert	remove	index	count	extend	reverse	sort
1-10	Y	Y						
11-15	Y	Y	Y					
16-20	Y	Y	Y	Y	Y			
21-25	Y	Y	Y	Y	Y	Y		
26-30	Y	Y	Y	Y	Y	Y	Y	Y

舉例來說，若輸入為

```
a = list()
a
a.append(5)
a.append(10)
a.append(15)
a.append(20)
```

```
a.append(25)
a
a.insert(3, 17)
a.remove(20)
a.append(15)
a
b = list()
b.append(15)
b.append(25)
b.append(50)
b.remove(17)
b
b.index(17)
a.extend(b)
a
a.count(15)
a.index(15)
a.reverse()
a
a.sort()
a
```

則輸出應該為

```
[]
[5,10,15,20,25]
[5,10,15,17,25,15]
[15,25,50]
-1
[5,10,15,17,25,15,15,25,50]
3
2
[50,25,15,15,25,17,15,10,5]
[5,10,15,15,15,17,25,25,50]
```