

# Linked List

## Problem:

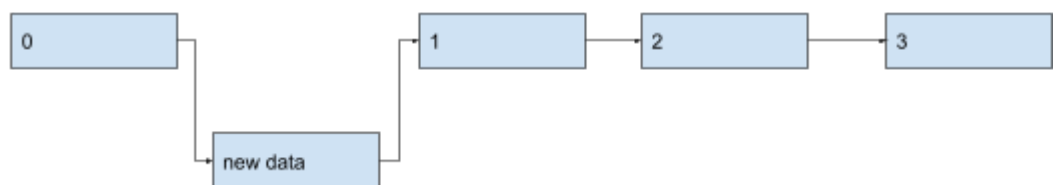
創建一個名為 `myList` 的 class, 其內部結構需為動態配置記憶體體的 Linked List 架構, 且需要以 `template` 來實作(即為此 class 可以放任意 type 的資料, 我們將只會測試 `c++` 原始 type), 並完成以下函數。

- Constructor: 創建一個空的 `myList` class 。
- Destructor: 將 `myList` 裡所有內容刪除, 釋放所有用到的記憶體。
- `push_back`:
  - a. 傳入一個變數, 其型態為此 class 宣告時的型態。
  - b. 在 class 尾端新增一個節點, 其值為傳入的數值。
- `push_front`:
  - a. 傳入一個變數, 其型態為此 class 宣告時的型態。
  - b. 在 class 最前端新增一個節點, 其值為傳入的數值。
- `insert`:
  - a. 傳入兩個變數, 第一為非負整數 `index`, 第二為所需插入的變數, 其型態為此 class 宣告時的型態。
  - b. 在 class `index` 的位置新增一個節點, 其值為傳入的數值。  
ex: 例如 `insert(1, new data)` 則如下圖所示

original



after insert(1, new data)



- c. (PS 如果 list 目前為空時, `insert(0, new data)` 為合法操作)
- `pop_back`:
  - a. 不傳入任何值, 將 list 的最後一個節點移除, 並釋放其記憶體。
- `pop_front`:
  - a. 不傳入任何值, 將 list 的第一個節點移除, 並釋放其記憶體。
- `pop_index`:
  - a. 傳入一個非負整數 `index`

- b. 將list的index (index 從0算起)位置節點移除，並釋放其記憶體。
- c. ex:例如pop\_index(2)則如下圖所示。

original



after pop\_index(2)



- print:
  - a. 不傳入任何數值，將list從頭至尾的數值輸出。

### Limit:

請勿使用任何**STL**所提供的資料結構來完成，新增與刪除需使用 **new**, **delete**, **malloc**, **free**等函數。

請使用助教所提供的**main.cpp**，並請勿修改任何文字。然而你可以修改任何**mylist.h**中的任何程式碼以達成目標。

### Hint:

- template需要在宣告處就將其實作，因此這次可以將全部實做放置於mylist.h檔案內。
- 為了達成linked list結構你可能會需要自行額外宣告class來輔助其完成。
- 請好好釋放記憶體，助教將會看原始碼來確保有適當的釋放記憶體。

## Input:

本題將維護兩個myList class分別的型態為int與double

輸入第一行為一個正整數T, 代表接下來有T行指令。

接下來T行, 每行開頭為一個正整數(1~7), 分別代表不同的7種指令。

- 指令1: push\_back
  - 會有接續兩個數, 分別為整數與倍精度浮點數, 代表myList型態為int與double所需要呼叫push\_back的參數。
- 指令2: push\_front
  - 會有接續兩個數, 分別為整數與倍精度浮點數, 代表myList型態為int與double所需要呼叫push\_front的參數。
- 指令3: insert
  - 會有接續三個數, 第一個為非負整數index, 而後兩個分別為整數與倍精度浮點數, 代表myList型態為int與double所需要呼叫push\_back的參數。
- 指令4: pop\_back
  - myList型態為int與double分別呼叫pop\_back函數。
- 指令5: pop\_front
  - myList型態為int與double分別呼叫pop\_front函數。
- 指令6: pop\_index
  - 會有接續一個非負整數index, 第一個為非負整數index代表myList型態為int與double所需要呼叫pop\_index的參數。
- 指令7: print
  - myList型態為int與double分別呼叫print函數。

## Output

當指令為7時, 將會印出list中的數值。

**Sample input:**

```
27
1 1 1.0
1 2 2.0
1 3 3.0
1 4 4.0
1 5 5.0
7
2 8 8.8
2 9 9.9
7
3 3 -2 -2.2
3 6 -4 -4.4
7
4
4
7
5
5
7
6 2
6 2
7
4
4
4
7
3 0 88 88.88
7
```

**Sample output:**

```
int_list: 1 2 3 4 5
double_list: 1 2 3 4 5
```

int\_list: 9 8 1 2 3 4 5

double\_list: 9.9 8.8 1 2 3 4 5

int\_list: 9 8 1 -2 2 3 -4 4 5

double\_list: 9.9 8.8 1 -2.2 2 3 -4.4 4 5

int\_list: 9 8 1 -2 2 3 -4

double\_list: 9.9 8.8 1 -2.2 2 3 -4.4

int\_list: 1 -2 2 3 -4

double\_list: 1 -2.2 2 3 -4.4

int\_list: 1 -2 -4

double\_list: 1 -2.2 -4.4

int\_list:

double\_list:

int\_list: 88

double\_list: 88.88