Question 1. (35%)

Please use the following structure to implement a singly linked list.

```
typedef struct listNode *listPointer;
typedef struct listNode {
        int data;
        listPointer link;
}
```

The singly linked list must have the following functions：

1. **add i :** add the new node whose data field is i to the end of the list. (5%)
2. **del i :** delete the *i*-th node. (5%)
3. **ins i j :** insert the new node whose data field is j after the *i*-th node. (5%)
4. **mul i j** : the data field of the *i*-th to last node multiplies by j. (5%)
5. **rev k :** treat K data as a group and reverse them. (15%)
6. **show :** print out the data in the singly linked list.

Example

- **Initial list :** empty
- add 5
- add 7
- add 9
- show
- 5 7 9

Example

- **Initial list :** 5 7 9 7 9
- del 2
- show
- 5 9 7 9
- **Initial list :** 5 7 9 7 9
- del 0 / del 6 …
- show
- 5 7 9 7 9
- (No change!)

Example

- **Initial list:** 1 2 3 4 5 7 8 9

  (index: 1 2 3 4 5 6 7 8)
- ins 5 6
- show
- 1 2 3 4 5 6 7 8 9

Example

- **Initial list:** 1 2 3 4 5 6 7 8 9
- mul 3 2
- show
- 1 2 3 4 5 6 14 8 9

Example

- **Initial list:** 1 2 3 4 5 6 7 8
- rev 3
- show
- 3 2 1 6 5 4 8 7
- (最後一組數量雖不足3，但視為一組反轉)

Input & output

- Read the file (input_1.txt)
- The **first line** presents the **initial list**.
- The **second line** contains an integer **n** indicates how many commands there are.
- The next **n** rows are the **commands**.
- Implement the functions in next slice, and write the result of “show” to the file (output_1.txt).

Example

(Input)

- 1 2 3 4 5 6
- 5
- add 9
- add 8
- rev 3
- del 1
- show

(Output)

- 2 1 6 5 4 8 9