# Test3_zad1

Реализирайте едносвързан списък, който има следните операции:

1. add(X,pos) - добавя числото X на позиция pos в писъка, като ако няма възможност да се добави на такава позиция(отрицателен индекс или прекаленно голям индекс) добавя елемента в края на списъка и извежда след това съобщение на стандартният изход add_last.
2. remove(pos) - премахва елемента, който е на позиция pos, ако няма възможност да се премахне на такава позиция(отрицателнен индекс или прекаленно голям индекс) не се премахва елемент, а на стандартният изход се извежда текста remove_failed
3. print() - изкарва на стандартният изход числата от списъка, като след всяко число се принтира символа #. При празен списък не се изкарва нищо на стандартният изход.
4. reverse() - обръща списъка на обратно, т.е. последният елемент става вече първи,предпоследният втори и т.н.
5. is_palindrom() - проверява дали списъка е палиндром(поредица,която се чете еднакво отпред назад и отзад напред), като ако е палиндром принтира на стандартният изход true, а ако не е false
6. count(X) - преброява, колко пъти се среща числото X в списъка и извежда резултата на стандартният изход
7. remove_all(X)-премахва всички срещания на числото X в списъка
8. group(startPos,endPos)-сумира елементите между подадените две позиции(включително позициите) и ги замества в масива със сумата им. (Пример: при масив 1,2,3,4,5,6,7,8,9 и group(2,4) получаваме списък 1,2,12,6,7,8,9) Ако позициите не са валидни операцията не се изпълнява, а на стандартният изход се принтира fail_grouping

**Input Format**

всеки тест започва с число N показващо броя на тестващите редици с операции. След това ще се подадат K на брой операции, като първо се подава числото K и след това всяка една операция. Всяка операция е на нов ред, като първо е името на операцията, а след това ако операцията има параметри те се подават с разделител интервал

**Constraints**

ще бъдат подадени максимум 1 милион операции.

**Output Format**

Изхода е спрямо указанията по-горе, като изхода от всяка тестова редица от операции се извежда на нов ред. При изкарването на резултатите не принтирайте никакви символи(интервали,табулации или нещо друго), които на са в указанието.

```
пример:
Вход:
2
3
add 1 0
add 2 1
print
5
add 10 0
```

```
add 20 0
add 30 2
remove 1
print

Изход:
1#2#
20#30#
```

Тестовете покриват всяка една функция и може да си тествате функционалността по време на писане като ги стартирате.

### Sample Input 0

```
4
4
add 1 0
add 2 1
add 3 2
print
4
add 1 0
add 2 0
add 3 0
print
4
add 1 0
add 2 1
add 3 1
print
4
add 1 2
add 2 2
add 3 2
print
```

### Sample Output 0

```
1#2#3#
3#2#1#
1#3#2#
add_lastadd_last1#2#3#
```

### Sample Input 1

```
6
5
add 1 0
add 2 1
add 3 2
remove 0
print
5
add 1 0
add 2 0
add 3 0
remove 1
print
5
add 1 0
add 2 1
add 3 1
```

```
remove 2
print
5
add 1 2
add 2 2
add 3 2
remove 3
print
8
add 1 2
add 2 2
remove 1
add 3 2
remove 1
add 5 2
remove 0
print
9
add 1 2
add 2 2
remove 1
remove 0
add 3 2
remove 1
add 5 2
remove 0
print
```

### Sample Output 1

```
2#3#
3#1#
1#3#
add_lastadd_lastremove_failed1#2#3#
add_lastadd_lastadd_lastadd_last5#
add_lastadd_lastadd_lastremove_failedadd_last5#
```

### Sample Input 2

```
4
5
add 1 0
add 2 1
add 3 2
reverse
print
5
add 1 0
add 2 0
add 3 0
reverse
print
7
add 1 0
add 2 1
add 3 1
reverse
print
reverse
print
6
add 1 2
add 2 2
add 3 2
reverse
```

```
reverse
print
```

### Sample Output 2

```
3#2#1#
1#2#3#
2#3#1#1#3#2#
add_lastadd_last1#2#3#
```

### Sample Input 3

```
4
4
add 1 0
add 2 0
add 3 0
is_palindrom
4
add 1 0
add 1 0
add 1 0
is_palindrom
4
add 1 0
add 2 0
add 1 0
is_palindrom
6
add 1 0
add 2 0
add 3 0
add 2 0
add 1 0
is_palindrom
```

### Sample Output 3

```
false
true
true
true
```