

Homework week 2

Basic data structures

*Note: Two consecutive numbers in the same line must be separated by only **one** space.

1. Given a list A of n integer numbers, your task is to write a program to count the number of pairs (i,j) that $A[i]=A[j]$.

Input:

- The first line contains an integer number n
- The second line contains n integer numbers separated by spaces.

Output: Write to the screen an integer number (i.e., the number of pairs (i,j) that $A[i]=A[j]$)

Example

Keyboard	Screen
6 5 2 4 2 2 5	4

2. The linked list structure has two following operations:
 - insert (p, x) : insert an integer number x at position p , elements from p are moved backward one position.
 - delete (p) : delete element at position p , elements after p are moved forward one position.

Start from an empty list, your task is to implement a linked list, perform operations read from the keyboard.

Input:

- The first line contains an integer number n which is the number of operations.
- The next n lines contain the operation description. One operation is in one line in either format:
 - insert $p\ x$
 - delete p

where p is the position and x is the number.

Output: Write the resulting linked list to the screen in one line. Numbers are separated by spaces.

Example

Keyboard	Screen
5 insert 0 1 insert 1 3 insert 1 4 insert 2 2 delete 1	1 2 3

3. Given a list of integer numbers, your task is to read these numbers into a doubly linked list, and implement function `count_triplets()` to count all positions p such that the sum of elements at positions $p-1$, p , and $p+1$ is zero.

Input:

- The first line contains an integer number n which is the number of numbers.
- The second line contains n integer numbers to read into `count_triplets` function

Output: Write to the screen an integer number that is the result of function `count_triplets()`.

Example

Keyboard	Screen
6 5 -3 -2 2 9 -11	2

4. The Queue structure has two following operations:

- `enqueue (x)`: insert integer number x at the tail the queue.
- `dequeue ()`: remove the element at the head of the queue.

Start from an empty queue, your task is to implement a queue, perform operations from the keyboard and write the resulting queue to the screen.

Input:

- The first line contains an integer number n which is the number of operations

- The next n lines contain the operation description. One operation is in one line in either format:

- enqueue x
- dequeue

where x is the number to enqueue

Output: Write the resulting queue to the screen. Numbers are separated by spaces.

Example

Keyboard	Screen
7 enqueue 1 enqueue 2 enqueue 3 enqueue 4 dequeue dequeue enqueue 5	3 4 5

5. The Stack structure has two following operations:

- push (x): insert integer number x to the top of the stack
- pop (): remove the element at the top of the stack

Start from an empty stack, your task is to implement a stack, perform operations from the keyboard and write the resulting stack to the screen

Input:

- The first line contains an integer number n which is the number of operations
- The next n lines contain the operation description. One operation is in one line in either format:

- push x
- pop

where x is the number to push to the stack.

Output: Write the resulting stack to the screen. Numbers are separated by spaces.

Example

Keyboard	Screen
7 push 1 push 2 push 3 push 4 pop pop push 5	1 2 5