#include <iostream>

#include <string>

using namespace std;

struct Node

{

int data;

bool mark;

Node \*next;

};

void prepend(int x , int m, Node\* &p)

{

if(m == -1 && p == NULL)

{

p = new Node;

p->data = x;

p->mark = false;

p->next = NULL;

}

else

{

Node \*New = new Node;

New->data = x;

New->mark = false;

New->next = p;

p = New;

}

}

void remove(int y, Node\* &p)

{

for(int i = 0; i < y; i++)

{

while(p->mark)

p = p->next;

p = p->next;

}

while(p->mark)

p = p->next;

p->mark = true;

}

void Delete(Node\* &head)

{

Node\* p = head;

Node\* pprev;

Node\* temp;

while(head != NULL && head->mark)

{

temp = head;

head = head->next;

delete temp;

p = head;

pprev = head;

}

while(p != NULL)

{

if(p->mark)

{

pprev->next = p->next;

temp = p;

p = p->next;

delete temp;

}

else

{

pprev = p;

p = p->next;

}

}

}

void clear (Node\* &p)

{

Node\* temp;

while(p != NULL)

{

temp = p;

p = p->next;

delete temp;

}

}

int main ()

{

string x;

int y , m = -1 , n = 0;

Node \* head;

head = NULL;

Node \* p;

while(x != "end")

{

cin >> x;

if(x != "end" && (x == "prepend" || x == "remove"))

cin >> y;

if(x == "prepend")

{

prepend(y , m , head);

p = head;

while(p != NULL)

{

cout << p->data;

if(p->mark)

cout << '\*';

if(p->next != NULL)

cout << ' ';

p = p->next;

}

cout << endl;

m++;

}

else if(x == "remove" && y <= m && m > -1)

{

m--;

p = head;

remove(y , p);

n++;

p = head;

if(n%3 != 0)

{

while(p != NULL)

{

cout << p->data;

if(p->mark)

cout << '\*';

if(p->next != NULL)

cout << ' ';

p = p->next;

}

cout << endl;

}

}

if (n%3 == 0 && n !=0 && x != "end" && x != "prepend")

{

if(m == -1)

head = NULL;

Delete(head);

p = head;

while(p != NULL)

{

cout << p->data;

if(p->next != NULL)

cout << ' ';

p = p->next;

}

if(head != NULL)

cout << endl;

if(m == -1)

head = new Node;

}

}

if(m > -1)

clear(head);

return 0;

}