

Week 5

- 주의 사항: 부정행위 금지, STL 사용 금지(string 사용가능), 인터넷 금지, 이중 연결 리스트(Doubly Linked list)를 이용하여 구현할 것.
- 표준 입출력 사용을 권장 (C는 scanf / printf, C++은 cin / cout)

문제 1

자연수를 저장하는 노드로 이루어진 이중 연결 리스트와 노드의 주소를 가리키는 포인터 변수 p 를 생성하는 프로그램을 구현한다. 이중 연결 리스트는 data를 저장하지 않는 노드인 header와 trailer를 갖고 있는 형태로 구현한다. 입력으로 주어질 명령어는 7개이며(예제 참고), 이에 따라 구현 해야 하는 함수는 다음과 같다. 메인 함수의 시작 전, p 는 리스트의 **begin()**을 사용해 초기화 해야 한다.

- **begin()** : p 가 리스트의 header의 다음 노드를 가리키게 한다.
- **end()** : p 가 리스트의 trailer를 가리키게 한다.
- **insert(e)** : p 가 가리키는 노드의 앞 부분에 $e(1 \leq e \leq 10,000)$ 를 저장하는 노드를 삽입한다.
- **erase()** : p 가 가리키는 노드를 삭제하고, p 가 가리키는 노드를 **begin()**을 이용하여 변경한다. 단, p 가 trailer를 가리키는 경우, 연산을 수행하지 않고, 리스트가 비어 있는 경우, "empty"를 출력한다.
- **nextP()**: p 를 p 가 가리키는 노드의 다음 노드로 이동시킨다. 단, p 가 trailer를 가리키는 경우, 이동시키지 않는다.
- **prevP()**: p 를 p 가 가리키는 노드의 이전 노드로 이동시킨다. 단, p 가 header의 다음 노드를 가리키는 경우, 이동시키지 않는다.
- **finddivisor(e)** : 리스트 내에서 $e(1 \leq e \leq 10,000)$ 의 약수가 저장된 노드의 index를 오름차순으로 모두 출력한다.(노드의 index는 0부터 시작한다.) 리스트 내에 e 의 약수가 저장된 노드가 없으면 -1을 출력한다.

입력

첫 번째 줄에 명령어의 수 $N(1 \leq N \leq 1,000)$ 이 주어진다. 두 번째 줄부터 N 개의 줄에는 명령어가 하나씩 주어진다.

출력

출력해야 하는 명령어가 주어질 때마다 그 결과를 한 줄 씩 출력한다.

예제 입출력

예제 입력	예제 출력
20	-1
finddivisor 2	empty
erase	0
insert 1	0 2
prevP	-1
erase	0
begin	
insert 2	
finddivisor 2	
erase	
insert 3	
insert 4	
insert 5	
finddivisor 4	
begin	
erase	
begin	
nextP	
end	
finddivisor 1	
finddivisor 6	