문제1

중간에 부분합, max, min 등등에서 갱신이 필요하면 세그먼트 트리가 유리

해당 문제는 덧셈트리로 구현

어레이크기 : 100만

1. 덧셈트리 만드는 함수: ex) K값 받기, A[i]=(A[2i]+A[2i+1])로 역순으로 반복문 채우기

2. 갱신함수: logN-> A[k], A[[k/2]]... 갱신

3. 덧셈연산함수: long long sum=0; 반복문을 통해 s,e의 짝수 홀수 여부에 따라 더하는 노드를 선택, 더 이상 노드를 선택할 수 없을 때까지 반복, 끝나면 출력

.

논리:

1. N이상의 가장 작은 2^k 찾기: k 반복문으로 찾기, 2^k가 2N보다 같거나 커지면 종료

2. 트리 채우기: \* N!=2^k이면 가까운 k 찾아서 A[2^(k-1)]부터 채운다

3. 1번쿼리에서 i,v 받았을 때 i=N+i, A[i]=v; /A[[i/2]]=A[[i/2]\*2]+A[[i/2]\*2+1];, i--; -> j=k, j>00/

4. 2번쿼리: s-> 2^(k-1)+s-1, e-> 2^(k-1)+e-1

5. 수열: 1-N, 트리내에서의 인덱스: 2^(k-1)~2^(k-1)+N, i-> 2^(k-1)+i-1

Structure:

변수선언: N, k

어레이 선언: long long A[백만]

덧셈트리 함수(N, 수열)

갱신함수(i, v)

구간합 함수(s,e)

main함수

변수선언: long long q, i(1-N), v, s, e

N입력

k 찾기

수열 입력받기

덧셈트리 함수 실행

쿼리 행 입력 반복

q값에 따라 실행

문제2: 기호를 트리로 구현하여 풀이  
문제3-6: 트리를 사용하여 푸는것으로 추정됨