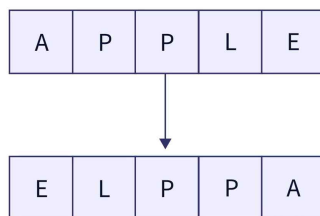


Reverse a string in C



플립(Flip)¹⁾

[문제] 1차원 binary pixel 배열 $P[N]$ 이 있다. 이 배열의 특정 구간 $[i:j]$ 을 **뒤집는** 동작(operation)인 **flip(i, j)**이 함수로 제공된다. 이 동작은 배열의 특정 구간 $P[i:j]$ 의 모든 값에 대하여 0은 1로, 1은 0으로 바꾼다. 일종의 toggle 작업이다. 만일 **flip(r, r)**이면 하나의 값 $P[r]$ 을 바꾸는 것과 같다. 아래에 제시된 배열 $P[30]$ 의 예를 들어 설명해보자.

우리는 처음 상태에서 아래와 같이 5번의 **flip(i, j)** 작업을 하였다. 그 진행 결과는 아래 표에 순차적으로 표시되어 있다. 여러분은 입력 파일에 제시된 모든 flip 작업을 마친 후, $P[N]$ 에서 존재하는 1's run의 개수 R 과 '1'의 개수 V 를 정수로 출력해야 한다. 1's run이란 숫자 1로만 구성된 최대 길이의 substring이다.

flip(i,j)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
초기	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
[9, 23]	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
[6, 15]	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
[13, 20]	0	0	0	0	0	0	1	1	1	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0
[2, 6]	0	0	1	1	1	1	0	1	1	0	0	0	0	1	1	1	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0
[20, 28]	0	0	1	1	1	1	0	1	1	0	0	0	0	1	1	1	0	0	0	0	1	0	0	0	1	1	1	1	1	1	0

위의 표에서 최종 $P[N]$ 에서 모든 1's run은 파란색으로 표시된 부분이며 그 개수는 $R=5$ 이다. 그리고 전체 $P[N]$ 에 존재하는 '1'의 개수는 15개이므로 여러분은 정답으로 '5 15'를 stdout에 출력해야 한다.

단 이번 문제에서 N 은 매우 큰 수가 될 수 있으므로 고정적인 C-style 배열을 사용하면 memory fault가 날 수 있다. 또 그 flip() 작업의 느린 수행 시간으로 인하여 time-out이 될 수 있으므로, 여러분은 특별한 자료구조를 고안(devise)해서 풀어야 한다. 가장 추천하는 자료구조는 link가 포함된 list< mynode > FLIP ; 이다.

1) 이번 문제는 flip (easy version)이다. 이 문제에서 $P[N]$ 을 $P[M][N]$ 으로 2차원 확장하면 상당히 까다로운 flip(hard version) 문제가 된다. 이 hard version을 감안하여 확장 가능하도록 flexible하게 작성하기를 권한다.

[입출력] 표준 입출력을 사용한다. 첫 줄에 배열의 크기 N , 그리고 flip 동작의 개수 K 가 주어진다. 이어지는 K 개의 줄에는 **flip**(**i**, **j**)의 입력인 두 정수 '**i j**'가 주어진다. 단 $10 \leq N \leq 2^{63}$ 이며 $5 \leq K \leq 1,000$ 이다. 또한 $0 \leq i \leq j \leq N$ 이다. 출력은 최종적으로 $P[N]$ 에 있는 1'run의 갯수와 그 모든 run의 길이의 합, 즉 1의 갯수를 첫 줄에 순서대로 출력해야 한다.

[예제]

stdin	stdout
30 5 // N=30, K=5	5 15 // Run, 1갯수
9 23 // flip(9,23)	
6 15	
13 20	
2 6	
20 28	

[제한조건] 프로그램 이름은 **flip.{c,cpp,py}**이다. 허용되는 token의 수는 최대 600이며 수행 시간은 0.5 초이므로 생각없이 자료구조를 구성하면 time-out될 수 있음을 유의해야 한다.