

Batch System Simulation

여러분은 time machine을 타고 과거 시절로 여행을 떠난다. 1960년대 중반에 도착한 여러분에게 주어진 컴퓨터는 배치 처리 방식으로 처리한다. 이제 여러분은 강의 시간에 들었던 배치 처리 방식을 따라 컴퓨터 사용자들이 제출한 프로그램을 처리하는 배치 OS(batch Operating System)를 흉내 내려고 한다.

즉, N 명의 사용자가 각각 자기의 프로그램을 시스템에 제출하면 OS가 이들을 제출한 순서대로 실행한다. 각 프로그램은 CPU 작업과 I/O 작업을 번갈아 가면서 반복한다. N 명의 사용자 프로그램 실행 패턴에 대한 정보가 주어질 때, 배치 운영체제가 이들을 차례대로 처리한 후, 모든 작업이 종료된 시점과 CPU 유휴시간(idle time)을 계산하려고 한다.

입력 :

입력 파일의 이름은 batch.inp이다. 첫째 줄에는 시스템이 처리해야 하는 프로그램의 수를 나타내는 정수 $N(3 \leq N \leq 1000)$ 이 주어진다. 이 N 개의 프로그램은 모두 시각 0에 제출되었다고 가정한다. 시스템이 처리하는 프로그램의 순서는 입력에서 주어지는 순서를 따른다.

각 프로그램은 아래와 같이 반복되는 CPU 작업과 I/O 작업에 소요되는 시간을 순서대로 나열한다. t_1 은 CPU 사용 시간, t_2 는 I/O 사용 시간, t_3 는 CPU 사용시간, ... 을 의미한다. 즉, $t_k(3 \leq k \leq 100)$ 에서 k 가 홀수이면 CPU 사용 시간을, 짝수이면 I/O 사용 시간을 의미한다. '-1'은 마지막 입력을 의미하며 이는 처리하지 않는다.

$$t_1, t_2, t_3, \dots, t_k, -1$$

출력 :

출력파일의 이름은 batch.out이다. N 개의 프로그램을 배치 OS가 모두 처리한 후, CPU 유휴시간과 처리가 종료된 시각을 한 줄에 출력하되, 두 값은 공백으로 구분한다.

예제 :

예 1	예 1에 대한 출력
3 2 2 3 3 3 -1 1 3 1 3 2 -1 2 3 2 -1	14 30
예 2	예 2에 대한 출력
5 2 10 3 12 5 12 -1 4 13 5 10 8 11 9 -1 6 11 8 12 5 11 9 12 -1 8 22 3 12 4 6 7 7 8 12 2 -1 3 3 8 12 3 22 12 23 12 23 32 -1	256 422

제한조건: 프로그램의 이름은 batch.{c,cpp,java}로 한다.

참고로, 아래 보인 그림은 <예 1>에 대응하는 처리 흐름도이다. 강의 시간에 언급했던 예이다.

