

## CPU Scheduling Algorithm – RR

CPU 스케줄링을 위한 RR 알고리즘을 프로그램으로 구현하라. 이번 과제는 실제로 프로세스가 실행되는 과정에 가깝게 흉내내는 것이다. 모든 프로세스는 CPU burst와 I/O burst를 번갈아 가면서 요청하며, 그 횟수는 프로세스마다 다르다. CPU는 시스템 내에 하나 밖에 없어 많은 프로세스가 ready queue에 기다리는 상황이 발생할 수 있다. 하지만 I/O 장치는 충분히 많기 때문에 각 프로세스가 I/O작업에 필요한 자원은 곧바로 할당받아 사용할 수 있다고 가정한다.

각 프로세스에 관해 다음과 같은 정보가 주어진다.

- ① 프로세스가 system에 도착한 시각 (즉, 사용자가 작업을 요청한 시각)
- ② 각각의 CPU burst와 I/O burst의 시간

어떤 프로세스가 ready queue에 들어가는 경우는 아래와 같이 크게 3가지 경우가 있다.

- 1. 이제 막 시작하는 프로세스(즉, ready queue에 처음으로 들어가는 프로세스 또는 사용자가 작업을 요청한 프로세스)
- 2. 입출력을 완료한 프로세스
- 3. 자기에게 할당된 time slice를 다 소진한 프로세스

방금 언급한 세 가지 종류의 프로세스가 어떤 시각에 동시에 ready queue에 도착했다면 앞에서 언급한 순서대로 우선순위를 갖는다. 즉, 이제 막 시작하는 프로세스가 ready queue에 가장 먼저 들어간다. 만약 동일한 시각에 입출력을 마친 프로세스들이 다수이면 프로세스 ID가 작은 것이 먼저 ready queue에 들어간다.

### 입력 :

입력 파일의 이름은 rr.inp이다. 첫째 줄에는 총 프로세스의 개수  $n$ 과 time slice를 나타내는  $q$ 값이 주어진다. 이어서  $n$ 개의 프로세스에 대한 정보가 차례로 주어진다.

각 프로세스에 관한 정보는 2줄로 구성된다. 첫째 줄에는 두 정수  $t$ 와  $k$ 가 주어지는데,  $t$ 는 프로세스가 system에 도착한 시간인 시각,  $k$ 는 CPU burst와 I/O burst의 총 횟수를 나타낸다. 둘째 줄에는  $k$ 개의 정수가 공백으로 구분되어 주어지는데 이는 각각 CPU burst, I/O burst, CPU burst, I/O burst, ..., CPU burst 순환에 따른 각각의 크기를 나타낸다. 처음과 마지막 값은 항상 CPU burst 값이며 따라서  $k$ 는 항상 홀수이다.

프로세스가 시스템에 도착한 시각  $t$ 는 감소하지 않는 순서로 주어진다.

$n$ 의 값은 최대 100이며, 프로세스 ID는 1부터  $n$ 까지 입력 순서대로 부여된다.  $k$ 의 값은 최대 99이다.

### 출력 :

출력파일의 이름은 rr.out이다.  $n$ 개의 프로세스가 모두 마친 후 각 프로세스가 ready queue에서 대기한 총 시간을 프로세스 번호 순서대로 출력하되, 프로세스 번호와 대기시간을 보여라.

### 예제 :

입력 예	입력 예에 대한 출력
4 5	1 22
0 7	2 27
3 20 5 25 6 15 5	3 24
0 7	4 4
7 11 4 18 10 31 3	
0 5	
10 30 8 20 4	
20 5	
5 19 10 20 3	

제한조건: 프로그램은 `rr.{c,cpp,java}`로 한다.