

	3주차 B																
제목	우선순위기반 멀티태스크 성능측정																
내용	<p>김인하는 운영체제를 개발하는 회사에서 일하고 있다. 운영체제의 성능 예측을 위해 각 태스크들의 평균 대기시간(average waiting time)과 평균 응답시간(average response time)을 계산하는 프로그램을 작성하고자 한다.</p> <ul style="list-style-type: none"><li>- 대기시간: 프로세스가 대기상태로 진입하여 실제 CPU를 할당 받아 실행을 시작하는 데까지 기다리는 시간</li><li>- 응답시간: 프로세스가 대기상태로부터 결과를 만들어서 사용자에게 알려주는 데 걸리는 시간</li></ul> <p>해당 운영체제는 태스크마다 우선순위가 정해져, 수행 중인 태스크가 끝날 경우 대기중인 태스크들 가운데 가장 우선순위가 높은 태스크를 골라서 실행한다. (이것을 우선순위 기반 스케줄링 방식이라고 한다.) 단 우선순위가 같은 태스크가 여러 개가 있을 경우 발생한 시간이 빠른 순서대로 수행한다.</p> <p>예를 들어 다음과 같은 세 개의 태스크가 있다고 가정하자. (숫자가 작을 수록 우선 순위는 높다)</p> <table><tr><th>Task id</th><th>발생시간(초)</th><th>실행시간(초)</th><th>우선순위</th></tr><tr><td>0</td><td>0</td><td>10</td><td>7</td></tr><tr><td>1</td><td>2</td><td>10</td><td>7</td></tr><tr><td>2</td><td>8</td><td>5</td><td>4</td></tr></table> <p>첫 번째 태스크인 task0은 대기시간 0초 실행종료시간은 10초이다. task0이 이 끝나면 그 때까지 발생한 태스크 중 우선 순위가 높은 2번 태스크가 먼저 실행이 된다. 이 때 task2는 대기시간 2(=10-8)이며 태스크의 종료시각은 15이므로 응답시간은 7이다. Task1은 그 이후에 실행되므로, 대기시간은 13(=15-2), 응답시간은 23(=25-2)이다. 이 때의 평균 대기시간과 평균 응답시간은 각각 5초와 13초다. (소수점 이하 무시)</p> <p>하지만 우선순위가 바뀌게 되면 평균응답시간과 평균대기시간이 바뀐다. 2초와 8초 사이에 task1의 우선순위를 3으로 변경했다고 가정하면, task0은 변화가 없겠지만, task1의 대기시간과 응답시간은 8초와 18초, task2의 경우는 12초와 17초가 된다. 이 때의 평균 대기시간과 평균응답시간은 6초와 15초다. 단 이미 완료된 task의 경우 우선순위를 변경하는 명령이 입력되어도 그 효과는 없다.</p> <p>입력으로 여러 개의 태스크들에 대하여 각각의 태스크 id(태스크 번호), 해당 태스크가 발생한 시각, 완료되는데 필요한 시간(결과를 만들어 내는데 필요한 시간), 그리고 그 태스크의 우선순위가 주어진다. 또, 실행 중 특정한 task의 경우 우선순위를 변경시키는 명령도 주어진다. 이때 평균 대기시간과 평균 응답시간을 계산하는 프로그램을 작성하시오.</p>	Task id	발생시간(초)	실행시간(초)	우선순위	0	0	10	7	1	2	10	7	2	8	5	4
	Task id	발생시간(초)	실행시간(초)	우선순위													
	0	0	10	7													
	1	2	10	7													
	2	8	5	4													

		<p>사용할 수 있는 언어는 C, C++로 제한한다. 프로그램의 실행 시간은 5 초를 초과할 수 없다. C++의 경우 main 함수 내의 시작 지점에 다음 내용을 추가함으로써 cin 입력 속도를 개선할 수 있다.</p> <pre>std::ios::sync_with_stdio(false);</pre>
입력 형식		<p>입력은 standard in으로 다음과 같이 주어진다.</p> <ol style="list-style-type: none"> <li>첫 번째 줄에는 테스트케이스의 수 <math>T</math>가 주어진다. (<math>0 &lt; T \leq 1,000</math>)</li> <li>두 번째 줄에는 첫 번째 테스트 케이스에서 주어진 태스크의 숫자 <math>N</math> (<math>2 \leq N \leq 1,000</math>) 과 이 태스크들의 우선순위를 변경하는 명령의 숫자 <math>M</math> (<math>0 \leq M \leq 100, M \leq N</math>)이 빈칸을 사이에 두고 주어진다.</li> <li>세 번째 줄부터 <math>N + M + 2</math> 까지의 <math>N + M</math> 개의 각 줄에는 태스크의 정보를 나타내는 4개의 숫자, 혹은 태스크의 우선순위를 변경하는 명령이 발생 시간 순서에 따라 주어진다. <ol style="list-style-type: none"> <li>각 태스크의 정보는 태스크의 id, 태스크가 발생하는 시각과 결과를 만들어 내는데 필요한 CPU time, 그리고 우선순위를 의미하는 4개의 자연수 <math>a, b, c, d</math>가 빈칸을 사이에 두고 주어진다. (단, 태스크의 발생시간 순서대로 입력되며 같은 시각에 여러 태스크가 발생하는 경우는 없다.) (<math>0 \leq a \leq 1,000, 0 \leq b, c \leq 2^{31} - 1, 0 \leq d \leq 7</math>)</li> <li>태스크의 우선순위를 변경하는 명령어의 경우 명령을 의미하는 알파벳 N과 빈칸이 주어진 후, 변경 명령어의 입력 시각, 태스크의 id, 그리고 변경하려는 우선순위 값을 의미하는 3개의 정수가 빈칸을 사이에 두고 입력된다.</li> </ol> </li> <li>그 이후에는 위의 2, 3단계가 <math>T - 1</math> 번 반복된다.</li> </ol>
출력 형식		<p>출력은 standard out으로 표시하며, 총 <math>T</math>줄로 이루어진다. 각 줄에는 각 테스트케이스 별로 주어진 태스크들의 평균 대기시간과 평균 응답시간을 의미하는 두 숫자를 빈칸을 사이에 두고 출력한다. (단, 소수점 이하는 무시하고 정수로 표시할 것)</p>
예	입력	<pre>2 3 0 0 0 10 7 1 2 10 7 2 8 5 4 3 1 0 0 10 7 1 2 10 7 N 6 1 3 2 8 5 4 (빈 줄)</pre>
	출력	<pre>5 13 6 15 (빈 줄)</pre>