**20년-1차(9기) 상시채용**

**O 코딩테스트**

- 시험시간 120분

**1. 놀이공원 가격 구하기**

놀이공원의 가격 Price, 이용 횟수 Count, 내가 가진 돈 Money가 주어진다. 놀이공원 가격의 총합은 이용 횟수와 가격을 곱한 값의 총합이다. 가격이 3000원이고 3번 이용하면 3000 \* 1 + 3000 \* 2 + 3000 \* 3 이다. 내가 가진 돈보다 가격이 높을 경우 차이값을 반환하고 그렇지 않을 경우 0을 반환

1<= price <= 2,500

1<= money <= 2,500

1<= count <= 1,000,000,000

|  |
| --- |
| Test case  Input / price = 1000, count = 3, money = 5000  Answer 1000  Input / price = 2000, count = 2, money = 8000  Answer 0 |

**2. 가장 많이 매칭되는 키오스크 찾기**

(1,2,3…의 번호를 가진) N개의 키오스크가 있다. N개의 키오스크 중 가장 많이 사용한 키오스크의 횟수를 구해보라.

(키오스크가 하나 이상 비어있을 경우)

1. 키오스크가 비어있으면 가장 오랫동안 사용을 안한 키오스크로 고객을 보낸다.

2. 그런 키오스크가 여러 개라면, 순번이 낮은 키오스크로 고객을 보낸다.

(키오스크가 전부 꽉 차 있을 경우)

1. 가장 빨리 끝날 것 같은 키오스크를 찾는다.

2. 그런 키오스크가 여러 개라면, 순번이 낮은 키오스크로 고객을 보낸다.

(test case)

[현재 시간(xx월xx일xx시간xx분xx초)] [들어온 시간, 걸린시간(1~60분)]

1. [’12-01-23-31-00’] [’12-01-23-32-00’, ‘30’]

2. [’12-01-23-35-00’] [’12-01-23-37-00’, ‘20’]

3. [’12-01-23-40-00’] [’12-01-23-43-00’, ‘20’]

(키오스크가 2개일 경우)

1. 키오스크 1번 사용 중 : 종료 시간 -> 12월 02일 00시 02분 00초

2. 키오스크 1,2번 사용 중 : 종료 시간 -> 12월 02일 00시 02분 00초, 12월 01일 23시 57분 00초

3. 키오스크가 둘다 사용 중이며 2번 키오스크가 가장 빨리 끝나므로 3번째 고객은 2번 키오스크를 사용한다. 즉 종료 시간 -> 12월 02일 00시 02분 00초, 12월 02일 00시 03분 00초로 업데이트 될 것이다.

그렇다면 최종적으로 1번 키오스크 1명 이용, 2번 키오스크를 2명이 이용했으므로 가장 많이 사용한 키오스크는 2번이며 횟수는 2번이다.

**3. 손님이 기다린 총 시간 찾기**

은행 창구의 개수, 손님이 도착한 시각과 은행창구를 이용할 시간이 주어진다. 그때 손님이 기다린 총 시간을 구해라

|  |
| --- |
| Test case  Input / n = 2, [[1,7],[2,8],[5,5]]  Answer 3  세번 째 손님이 5초에 도착했을 때 비어있는 은행 창구가 없다 따라서 가장 빨리 끝나는 8초일 때까지 3초가 걸린다. |

**4. SQL 조건문**

차 정보에 관한 Table이 주어진다. 이름, 아이디, 속도, 충전 속도가 주어진다. 속도가 150이상이고 충전 속도가 50이하인 차들의 정보를 검색하는 쿼리문을 작성하시오.

**1. 놀이공원 이용하기 위한 부족한 금액 구하기**

어떤 놀이 공원의 요금은 본인이 탑승한 기구에 비례하여 받는다. 예를 들어, 기본료가 1000원일 때, 3번 기구를 탄다고 가정하면 (1000 \* 1) + (1000 \* 2) + (1000 \* 3) = 6000원을 내야한다.

이 때, 기본료 Price, 이용 횟수 Count, 가지고 있는 돈 Money가 주어질 때 부족한 금액을 구하여라. (단, 돈을 충분히 가지고 있을 땐 0을 출력)

**2. 가장 많이 매칭되는 키오스크 찾기**

(1,2,3…의 번호를 가진) N개의 키오스크가 있다. 가장 많이 사용한 키오스크의 횟수를 구하라.

(키오스크가 하나 이상 비어있을 경우)

1) 키오스크가 비어있으면 가장 오랫동안 사용을 안한 키오스크로 고객을 보낸다.

2) 그런 키오스크가 여러 개라면, 순번이 낮은 키오스크로 고객을 보낸다.

(키오스크가 전부 꽉 차 있을 경우)

1) 가장 빨리 끝날 것 같은 키오스크를 찾는다.

2) 그런 키오스크가 여러 개라면, 순번이 낮은 키오스크로 고객을 보낸다.

(test case)

[현재 시간(xx월xx일xx시간xx분xx초)] [들어온 시간, 걸린시간(1~60분)]

1) [’12-01-23-31-00’] [’12-01-23-32-00’, ‘30’]

2) [’12-01-23-35-00’] [’12-01-23-37-00’, ‘20’]

3) [’12-01-23-40-00’] [’12-01-23-43-00’, ‘20’]

(키오스크가 2개일 경우)

1) 키오스크 1번 사용 중 : 종료 시간 -> 12월 02일 00시 02분 00초

2) 키오스크 1,2번 사용 중 : 종료 시간 -> 12월 02일 00시 02분 00초, 12월 01일 23시 57분 00초

3) 키오스크가 둘다 사용 중이며 2번 키오스크가 가장 빨리 끝나므로 3번째 고객은 2번 키오스크를 사용한다. 즉 종료 시간 -> 12월 02일 00시 02분 00초, 12월 02일 00시 03분 00초로 업데이트 될 것이다.

그렇다면 최종적으로 1번 키오스크 1명 이용, 2번 키오스크를 2명이 이용했으므로 가장 많이 사용한 키오스크는 2번이며 횟수는 2번이다.

**3. 대기 시간의 총 합 구하기**

창구의 개수(n), [고객이 도착한 시각(초), 소요 시간] 의 꼴로 데이터가 주어질 때, 고객이 기다린 시간의 총 합을 구하여라.

ex) n = 1, [[1,4],[5,3],[6,4]] 일 때, 6초에 도착한 손님은 8초까지 기다려야 하므로 총 기다린 시간은 2초

**4. (SQL) WHERE 절에 조건 두 개 사용하기(AND)**

SELECT \*

FROM CARS\_INFO

WHERE SPEED >= 150 AND CHARGE <= 50