**문제 설명**

**셔틀버스**

카카오에서는 무료 셔틀버스를 운행하기 때문에 판교역에서 편하게 사무실로 올 수 있다. 카카오의 직원은 서로를 '크루'라고 부르는데, 아침마다 많은 크루들이 이 셔틀을 이용하여 출근한다.

이 문제에서는 편의를 위해 셔틀은 다음과 같은 규칙으로 운행한다고 가정하자.

* 셔틀은 09:00부터 총 n회 t분 간격으로 역에 도착하며, 하나의 셔틀에는 최대 m명의 승객이 탈 수 있다.
* 셔틀은 도착했을 때 도착한 순간에 대기열에 선 크루까지 포함해서 대기 순서대로 태우고 바로 출발한다. 예를 들어 09:00에 도착한 셔틀은 자리가 있다면 09:00에 줄을 선 크루도 탈 수 있다.

일찍 나와서 셔틀을 기다리는 것이 귀찮았던 콘은, 일주일간의 집요한 관찰 끝에 어떤 크루가 몇 시에 셔틀 대기열에 도착하는지 알아냈다. 콘이 셔틀을 타고 사무실로 갈 수 있는 도착 시각 중 제일 늦은 시각을 구하여라.

단, 콘은 게으르기 때문에 같은 시각에 도착한 크루 중 대기열에서 제일 뒤에 선다. 또한, 모든 크루는 잠을 자야 하므로 23:59에 집에 돌아간다. 따라서 어떤 크루도 다음날 셔틀을 타는 일은 없다.

**입력 형식**

셔틀 운행 횟수 n, 셔틀 운행 간격 t, 한 셔틀에 탈 수 있는 최대 크루 수 m, 크루가 대기열에 도착하는 시각을 모은 배열 timetable이 입력으로 주어진다.

* 0 ＜ n ≦ 10
* 0 ＜ t ≦ 60
* 0 ＜ m ≦ 45
* timetable은 최소 길이 1이고 최대 길이 2000인 배열로, 하루 동안 크루가 대기열에 도착하는 시각이 HH:MM 형식으로 이루어져 있다.
* 크루의 도착 시각 HH:MM은 00:01에서 23:59 사이이다.

**출력 형식**

콘이 무사히 셔틀을 타고 사무실로 갈 수 있는 제일 늦은 도착 시각을 출력한다. 도착 시각은 HH:MM 형식이며, 00:00에서 23:59 사이의 값이 될 수 있다.

**입출력 예제**

| **n** | **t** | **m** | **timetable** | **answer** |
| --- | --- | --- | --- | --- |
| 1 | 1 | 5 | ["08:00", "08:01", "08:02", "08:03"] | "09:00" |
| 2 | 10 | 2 | ["09:10", "09:09", "08:00"] | "09:09" |
| 2 | 1 | 2 | ["09:00", "09:00", "09:00", "09:00"] | "08:59" |
| 1 | 1 | 5 | ["00:01", "00:01", "00:01", "00:01", "00:01"] | "00:00" |
| 1 | 1 | 1 | ["23:59"] | "09:00" |
| 10 | 60 | 45 | ["23:59","23:59", "23:59", "23:59", "23:59", "23:59", "23:59", "23:59", "23:59", "23:59", "23:59", "23:59", "23:59", "23:59", "23:59", "23:59"] | "18:00" |

**에디터**



1

#include <string>

2

#include <vector>

3

​

4

using namespace std;

5

​

6

string solution(int n, int t, int m, vector<string> timetable) {

7

   string answer = "";

8

   int time\_h[10];

9

   int time\_m[10];

10

   time\_h[0]=9;

11

   time\_m[0]=0;

12

   int a = 0;

13

   int cnt = 0 ;

14

   for(int i=1; i<n; i++){

15

       a = time\_m[i-1] + t;

16

       if(a >= 60){

17

           time\_h[i] = time\_h[i-1] + 1;

18

           time\_m[i] = a - 60;

19

      }else{

20

           time\_h[i] = time\_h[i-1];

21

           time\_m[i] = a;

22

          }

23

      }

24

   //타임테이블 파싱

25

   int tt\_h[2000];

26

   int tt\_m[2000];

27

   int total = timetable.size();

28

   for(int i = 0; i < total; i++) {

29

       string tmp = "";

30

       tmp += timetable[i][0];

31

       tmp += timetable[i][1];

32

       tt\_h[i] = stoi(tmp);

33

34

       tmp = "";

35

       tmp += timetable[i][3];

36

       tmp += timetable[i][4];

37

       tt\_m[i] = stoi(tmp);

38

  }

39

   // m명과 tt 처리

40

   int cnt=0;

41

   for(int i=0; i<total; i++){

42

       if(tt\_h[i] < 9){

43

           cnt+=1;

44

      }

45

       if(tt\_h[i] == 9 && tt\_m[i]==0){

46

            cnt+=1;

47

      }

48

  }

49

   int sum = total - cnt;

50

   cnt = 0;

51

   for(int i=1; i<n-1; i++){

52

       //tt\_h[i]를 뒤진다.

53

       for(int j=0; j<toatl; j++){

54

           if( time\_h[i-1] <= tt\_h[j] < time\_h[i])

55

               cnt+=1;

56

           if(tt\_h[j] == time\_h[i] && tt\_m[j]==0)

57

               cnt+=1;

58

          }

59

    }

60

     sum -= cnt;

61

   if(sum > m){

62

       z = sum-m

63

  }

64

   else{

65

       ans\_h = time\_h[n-1];

66

       ans\_m = time\_m[n-1];

67

  }

68

69

70

      }

71

72

  }

73

74

   return answer;

75

}

**실행결과**

실행 결과가 여기에 표시됩니다.

#include <string>

#include <vector>

using namespace std;

string solution(int n, int t, int m, vector<string> timetable) {

string answer = "";

int time\_h[10];

int time\_m[10];

time\_h[0]=9;

time\_m[0]=0;

int a = 0;

int cnt = 0 ;

for(int i=1; i<n; i++){

a = time\_m[i-1] + t;

if(a >= 60){

time\_h[i] = time\_h[i-1] + 1;

time\_m[i] = a - 60;

}else{

time\_h[i] = time\_h[i-1];

time\_m[i] = a;

}

}

//타임테이블 파싱

int tt\_h[2000];

int tt\_m[2000];

int total = timetable.size();

for(int i = 0; i < total; i++) {

string tmp = "";

tmp += timetable[i][0];

tmp += timetable[i][1];

tt\_h[i] = stoi(tmp);

tmp = "";

tmp += timetable[i][3];

tmp += timetable[i][4];

tt\_m[i] = stoi(tmp);

}

// m명과 tt 처리

int cnt=0;

for(int i=0; i<total; i++){

if(tt\_h[i] < 9){

cnt+=1;

}

if(tt\_h[i] == 9 && tt\_m[i]==0){

cnt+=1;

}

}

int sum = total - cnt;

cnt = 0;

for(int i=1; i<n-1; i++){

//tt\_h[i]를 뒤진다.

for(int j=0; j<toatl; j++){

if( time\_h[i-1] <= tt\_h[j] < time\_h[i])

cnt+=1;

if(tt\_h[j] == time\_h[i] && tt\_m[j]==0)

cnt+=1;

}

}

sum -= cnt;

if(sum > m){

z = sum-m

}

else{

ans\_h = time\_h[n-1];

ans\_m = time\_m[n-1];

}

}

}

return answer;

}