



## Statistics

### Problem Statement

#### Problem Statement for Cafeteria

##### Problem Statement

During the week, the cafeteria of my university is open until 2:30 PM. Normally, I will be at the university before this time anyway. Since I am a lazy person, I don't want to cook for myself and prefer eating in the cafeteria. But sometimes I don't have any classes in the morning, so I have to make sure to get to the university before the cafeteria closes.

There are several bus stops within walking distance of my home and each one has a bus connection to the university. For each bus stop, there will be a bus driving to the university every ten minutes, so I just need to remember the number of minutes after a full hour when the first bus departs from each stop, how long it takes to get to the bus stop and how long the bus needs to drive to the university.

Given a `int[] offset` (the number of minutes after a full hour when the first bus departs), `int[] walkingTime` (the number of minutes I need to walk to a bus stop) and a `int[] drivingTime` (the number of minutes the bus needs to drive to the university) you should write a method that returns the latest time when I have to leave my house in order to be at the university at 2:30 PM or before. The `i`th element of `offset`, `walkingTime` and `drivingTime` refers to the `i`th bus stop.

The format of the returned time should be HH:MM (HH the hour, MM the minute). The 12 hour time format is used (see examples).

##### Definition

Class: Cafeteria  
Method: latestTime  
Parameters: `int[], int[], int[]`  
Returns: `String`  
Method signature: `String latestTime(int[] offset, int[] walkingTime, int[] drivingTime)`  
(be sure your method is public)

##### Notes

- Times between noon and 1 PM should be written as 12:MM (where MM is the number of minutes after noon).

##### Constraints

- `offset`, `walkingTime` and `drivingTime` contain the same number of elements.
- `offset`, `walkingTime` and `drivingTime` contain between 1 and 50 elements, inclusive.
- Each element of `offset` is between 0 and 9, inclusive.
- Each element of `walkingTime` is between 1 and 30, inclusive.
- Each element of `drivingTime` is between 1 and 300, inclusive.

##### Examples

```
0)
{9}
```

```
{1}
```

```
{1}
```

Returns: "02:28"

There is only one bus stop. One bus departs at 2:29 and arrives at 2:30. Since the time to walk to the bus stop is 1 minute, the return value is 02:28.

1)

```
{6}
```

```
{9}
```

```
{120}
```

Returns: "12:17"

Leaving the house at 12:17 lets me catch the bus at 12:26 which arrives at 2:26 at the university. Note that the bus departing at 12:36 would arrive too late.

2)

```
{6,9}
```

```
{9,10}
```

```
{120,121}
```

Returns: "12:19"

Although the bus needs one minute more from the 2nd bus stop, and the walking time is also one minute longer, it is still better to leave the house at 12:19 and catch the bus at the 2nd bus stop at 12:29.

3)

```
{0,1,2,3,4,5,6,7,8,9}
```

```
{11,11,11,11,11,11,11,11,11}
```

```
{190,190,190,190,190,190,190,190,190,190,190}
```

Returns: "11:09"

This simulates a bus departing every minute.

4)

```
{7,4,0,0,2,1,6,7,7,0,8,6,0,5,0,6,7,9,0,2,4,8,4,7,
```

```
9,2,4,4,3,1,4,5,8,8,2,5,7,8,7,5,6,8,8,0,1,3,5,0,8}
```

```
{26,14,1,4,16,28,16,6,4,5,21,18,5,2,21,21,28,22,5,22,26,16,14,
```

```
19,19,19,4,12,24,4,30,16,28,20,25,2,30,18,4,6,9,22,8,3,7,29,8,30,6}
```

```
{151,264,280,89,63,57,15,120,28,296,76,269,90,106,31,222,
```

```
291,52,102,73,140,248,44,187,76,49,296,106,54,119,54,283,263,
```

```
285,275,127,108,82,84,241,169,203,244,256,109,288,9,262,103}
```

Returns: "02:07"

---

This problem statement is the exclusive and proprietary property of TopCoder, Inc. Any unauthorized use or reproduction of this information without the prior written consent of TopCoder, Inc. is strictly prohibited. (c)2010, TopCoder, Inc. All rights reserved.

---

This problem was used for:

Single Round Match 229 Round 1 - Division I, Level One

Single Round Match 229 Round 1 - Division II, Level Two

topcoder is also on



© 2015 topcoder. All Rights Reserved