

# UCF Local Programming Contest — August 30, 2014

## Fujiyama Thursday

*filename:* sushi

*(Difficulty Level:* Medium*)*

During the past year, the UCF programming team members started a weekly dinner at Fujiyama Sushi. Usually, the programming team members wait for everyone to arrive to start eating, but as a wise programming team coach from the north of campus always says: “regionals is coming.” Heeding this ominous warning, the programming team members need to spend more time to prepare. To spend more time practicing, the programming team members will start eating as they arrive. However, they will still wait until everyone is done eating to leave Fujiyama.

Each car the team members are taking holds exactly four people. As each car may take a different amount of time to arrive at Fujiyama and each team member may have different eating speeds, it is important to assign team members to cars in a careful manner. Any team member can be assigned to any car as all team members can drive any of the cars.

### The Problem:

Given the time it takes for cars to arrive and eating speeds of the team members, determine the minimum amount of time needed for all the team members to finish eating if the team members are assigned to cars optimally.

### The Input:

The first input line contains a positive integer,  $n$ , indicating the number of trips to Fujiyama Sushi. Each trip is represented by three lines. The first line for each trip contains a single integer,  $c$  ( $1 \leq c \leq 50$ ), representing the number of cars going to Fujiyama. The second line contains  $c$  integers,  $d_i$  ( $1 \leq d_i \leq 45$ ), representing the time (in minutes) it takes for the  $i^{\text{th}}$  car to arrive to Fujiyama. The third line contains  $4*c$  integers,  $t_j$  ( $1 \leq t_j \leq 75$ ), representing the time (in minutes) it takes for the  $j^{\text{th}}$  team member to finish eating.

### The Output:

For each trip, first output the heading “Trip # $d$ : ”, where  $d$  is the trip number, starting with 1. Then, output a single integer representing the minimum amount of time (in minutes) for all the team members to finish eating. Follow the format illustrated in Sample Output.

**Sample Input:**

```
3
1
40
1 2 3 4
2
10 20
5 6 3 4 8 9 1 2
3
15 20 20
10 10 10 10 20 20 20 20 30 30 30 30
```

**Sample Output:**

```
Trip #1: 44
Trip #2: 24
Trip #3: 45
```