Problem 8	**Wagon Train**	5 Points

Program Name: wagon.cpp **Input File: wagon.dat**

During the 1800s, there were countless wagon trains that traveled across the United States. The horses required to pull the wagons could travel 18-22 miles per day depending on the terrain. This is why, along east-west highways, so many towns are spaced 20 miles apart. These towns are a result of people settling where the wagons camped.

As a member of the Ranching Heritage Appreciation Society (RHAS), you and a group of 100 other people love to take spring-break wagon train adventures along America's Highways. The challenge is that you do not always travel from east to west. Thus, the towns are sometimes improperly spaced for wagon travel.

A travel strategy is a rule of the form, "Camp overnight at the first town you arrive at after travelling a minimum of *M* miles." You are to write a program that will determine what, if any, travel strategy will allow your wagon train to complete the journey within the five days of spring break. The journey is limited to five days because the weekends before and after the journey are required for assembly and disassembly of the wagon. A journey can be completed with a travel strategy if it follows the rules of travel:

1. The wagon train begins at the start point.
2. By the end of the fifth day, the wagon train must have reached the end point.
3. The wagon train may reach the end point at any time during any travel day.
4. The wagon train cannot travel more than 22 miles in any single day.
5. When the wagon train arrives in a town, it makes a decision based on the travel strategy. If the wagon train has traveled at least *M* miles, it will camp for the night. If the wagon train has traveled less than *M* miles, it will continue to the next town. Note that if *M* is improperly chosen, this may result in the wagon train travelling more than 22 miles.
6. Towns are considered to be a single point.
7. Distances between towns are rounded to the nearest mile.

Consider the example journey and sample strategies in Table 4, given the above rules.

Town	Journey Segment Distance	Sample Strategy with *M*=8	Sample Strategy with *M*=14	Sample Strategy with *M*=18
Pratt (start)	N/A	Start	Start	Start
Cullison	10	Day 1 = 10 (camp)	Day 1 = 10 (<14)	Day 1 = 10 (<18)
Wellsford	8	Day 2 = 8 (camp)	Day 1 = 18 (camp)	Day 1 = 18 (camp)
Haviland	4	Day 3 = 4 (<8)	Day 2 = 4 (<14)	Day 2 = 4 (<18)
Brenham	6	Day 3 = 10 (camp)	Day 2 = 10 (<14)	Day 2 = 10 (<18)
Greensburg	5	Day 4 = 5 (<8)	Day 2 = 15 (camp)	Day 2 = 15 (<18)
Mullinville	8	Day 4 = 13 (camp)	Day 3 = 8 (<14)	Day 2 = 23 (error)
Bucklin	10	Day 5 = 10 (camp)	Day 3 = 18 (camp)	N/A
Ford	10	Journey unfinished	Day 4 = 10 (<14)	N/A
Willroads Gardens	12	Journey unfinished	Day 4 = 22 (camp)	N/A
Dodge City (end)	5	Journey unfinished	Day 5 = 5 (finish)	N/A

Table 4 : Sample Journey from Pratt, Kansas to Dodge City, Kansas

In Table 4, you can see that the strategy (*M*=8) does not work because the wagon does not reach the end point by the end of the fifth day. The strategy (*M*=18) does not work because the wagon train will travel more than 22 miles on the second day and result in overstressed horses. The strategy (*M*=14) is successful because it allows the journey to be completed by the end of the fifth day without traveling more than 22 miles on any given day.

Your job is to write a program that, given the list of journey segment distances, will determine if a strategy ($1 \le M \le 22$) exists. Note that the town names are unimportant and that no journey segment distance will be greater than 22 miles.

CONTINUED NEXT PAGE

PROBLEM 8 CONTINUED

Input

Input to your program is a sequence of journey definitions. A journey definition uses exactly one line of input and consists only of integers. The first integer N ($1 \le N \le 25$) represents the number of journey segments. The next N integers are the journey segment distances. Your program should read to the end of the file.

Output

For each journey in the input file, your program should indicate if the journey is OK. If there exists at least one travel strategy that will allow the journey to be completed while following the rules of travel, your program should print "`Journey is OK`" on a line by itself to the screen. If there are no strategies that will allow the journey to be completed, your program should print "`Journey is not OK`" on a line by itself to the screen.

Example: Input File

```
10 10 8 4 6 5 8 10 10 12 5
5 20 20 20 20 20
5 22 22 22 22 22
10 3 7 5 3 12 12 12 19 12 5
```

Output to screen

```
Journey is OK
Journey is OK
Journey is OK
Journey is not OK
```