**PROBLEM 1**

Shortest Path in Cagayan

Write a program to find the shortest routing and distance between two municipalities in Cagayan using the following distance table.

You are not allowed to use any other manually computed distances in your program.

Tuguegarao City - Iguig 24 miles

Tuguegarao City - Amulung 71 miles

Iguig - Aparri 103 miles

Iguig - Penablanca 59 miles

Enrile - Piat 141 miles

Enrile - Penablance 65 miles

Enrile - Amulung 101 miles

Piat - Amulung 169 miles

Amulung - Penablanca 134 miles

|  |  |
| --- | --- |
| Sample input 1: | Sample Output 1: |
| Enter City #1 : Enrile  Enter City #2 : Iguig | Shortest routing and distance :  Enrile - Penablance - Iguig 124 miles |

|  |  |
| --- | --- |
| Sample input 2: | Sample Output 2: |
| Enter City #1 : Piat  Enter City #2 : Aparri | Shortest routing and distance :  Piat - Amulung – Tuguegarao - Iguig - Aparri 367 miles |

**PROBLEM 2**

Merging at the race

The Indianapolis 500 race is just a few weeks away. They need your help in creating a program that will provide the order in which the cars are to appear at the starting line (ordering is according to the number placed on the car).

There are two garages where the race cars are stored. Each Garage will form a line of 5 cars that are sorted by the number on the car. As the cars from the two garages merge to form a single start line, your program must maintain the ordering such that the smallest numbered cars are earlier in the line than those with larger numbers.

Note: Some cars may share the same number, and the numbers range from 1 to 10.

You may assume that there are always 5 cars in each garage prior to the merging and each garage is already sorted.

|  |  |
| --- | --- |
| Sample input 1: | Sample Output 1: |
| Enter the order for garage 1 : 11234  Enter the order for garage 2 : 22345 | The Final merged order is : 1122233445 |

|  |  |
| --- | --- |
| Sample input 2: | Sample Output 2: |
| Enter the order for garage 1 : 12345  Enter the order for garage 2 : 56789 | The Final merged order is : 1234556789 |

**PROBLEM 3**

TIME CONVERTER

Given a time in seconds determine its equivalent in hours, minutes and seconds. All values will be integers only.

|  |  |
| --- | --- |
| Sample input 1: | Sample Output 1: |
| Enter time in seconds : 7423 | Hours: 2  Minutes: 3  Seconds: 43 |

|  |  |
| --- | --- |
| Sample input 2: | Sample Output 2: |
| Enter time in seconds : 2630 | Hours: 0  Minutes: 43  Hours: 50 |

**PROBLEM 4**

LEAP YEAR

In the Gregorian calendar 3 criteria must be taken into account to identify leap years:

* The year is evently divisible by 4:
* If the year can be evenly divided by 100, it is NOT a leap year, unless;
* The year is also evenly divisible by 400. the it is a leap year.

Write a program to find the next 5 leap years. The user will input a leap year and the system will have to do the verification if it is a leap year then it will return the next 5 leap years.

|  |  |
| --- | --- |
| Sample input 1: | Sample Output 1: |
| Enter leap year: 1996 | 2000 2004 2008 2012 2016 |

|  |  |
| --- | --- |
| Sample input 2: | Sample Output 2: |
| Enter leap year: 2011 | 2011 is not leap year. |

**Problem 5**

PANCAKES

Enter the number of pancakes eaten for breakfast by 10 different people (Person1, Person2, ......,Person10).

Once the data has been entered the program must analyze the data and output which person ate the most pancakes for breakfast.

|  |  |
| --- | --- |
| Sample input 1: | Sample Output 1: |
| Person 1 : 4  Person 2 : 5  Person 3 : 1  Person 4 : 2  Person 5 : 7  Person 6 : 3  Person 7 : 5  Person 8 : 1  Person 9 : 3  Person 10 : 4 | Person 5 ate the most pancakes for breakfast. |

|  |  |
| --- | --- |
| Sample input 2: | Sample Output 2: |
| Person 1 : 4  Person 2 : 9  Person 3 : 1  Person 4 : 2  Person 5 : 7  Person 6 : 3  Person 7 : 5  Person 8 : 9  Person 9 : 6  Person 10 : 4 | Person 2,8 ate the most pancakes for breakfast. |