1. Write a Java program to print the word after removing the vowels, and arranging it in alphabetical order.

   Input format:
   Input consists of a single word(all in lowercase).

   Output format:
   The word after removing the vowels and arranging it in alphabetical order.

   Sample input:
   education

   Sample output:
   cdnt

2. Write a program to find the number of number pairs from a given set of numbers whose sum is equal to one of the elements in the set of numbers.

   Input format:
   Input consists of a set of numbers separated by space.

   Output format:
   A single number showing the total number pairs ({a,b} and {b,a} will be treated as the same) whose sum is equal to one of the elements in the set of numbers.

   Sample input:
   1 5 7 9 4

   Sample output:
   2

3. Write a program to create a class **IntegerArray** containing the following field: ***data***(an ***array*** of 5 ***integers***). The class should have the following member functions:
   inputData( ) - to input 5 integers (no duplicate numbers are to be entered).
   bubSort( ) - to sort the array in ascending order using the bubble sort technique and to display the sorted list.

   Input format:
   A single line containing 5 integers separated by space.

   Output format:
   - If all the numbers in the input are unique;

Then display the array in sorted order;
Otherwise display “INVALID INPUT”

Sample input:
7 4 9 2 0

Sample output:
0 2 4 7 9

4. A class **TellphoneCall** calculates the monthly bill of a customer. The members of the class are given below:
    - ***phNo***: 10-digit number (String)
    - ***custName***: First name of the customer(single word)
    - ***callSlots***: integer, Number of call slots used by the customer(number in the range (0 - 1000))
    - ***totalBill***: integer, Amount as per number of call slots, rates and rental charges

    Member functions:
    readCustomer(): Read the customer details to the respective fields.
    computeBill(): Calculate and display the bill for the customer.

No: of Slots	***Rate***
Upto 100	***Rs 500/-***
101 - 200	***Rs 8/- per slot***
201 - 300	***Rs 10/- per slot***
300 and above	***Rs 15/- per slot***

Rental charges are Rs 300 for all customers.

Input format:
Input consists of multiple lines.
First line should be the name of the person (first name with a single word).
Second line should be the phone number (unique 6-digit integer).
Third line should be number of call slots

Output format:
A single line containing the total bill amount.

Sample input:
John
9995123456
235

Sample output:

2650

5. The pension rules of a certain country states that a person receives Rs 500/- a week if he is over 65 years of age and extra Rs 100/- if he is over 70 years. Define a class **Person** with the following fields: ***name(string)*** and ***age(integer)***. Define a member function ***calculatePension( )*** for the class, which calculates the pension amount for the **Person** object. Write a program to read the details of a person and print the weekly pension amount he should receive. Use a constructor to read the values into ***name*** and ***age*** fields of each **Person** object. If a person is below the pensionable age, his pension amount will be 0.

Input format:

Input consists of multiple lines.
First line should be the name of the person (name can have spaces in between to separate parts of name).
Second line should be the age of the person.

Output format:

A single line containing the pension amount.

Sample input:

Joseph M
85

Sample output:

600

6. A point on a two dimensional plane can be represented by a pair of integers as (abscissa, ordinate); an x-coordinate and a y-coordinate. Write a program in Java to use a class **Point** to model a point. Given two points (x1,y1) and (x2,y2), the program should be able to calculate the midpoint between the two points using the formula: ((x1+x2)/2, (y1+y2)/2). Class **Point** should have the following fields: ***abscissa(integer)*** and ***ordinate(integer)***.
Your program should contain the following function:
midpoint(point1, point2) - Calculate the coordinates of the midpoint of the two **Point** objects passed to it and assign the result to a new **Point** object.

Input format:

Input consists of multiple lines.
First line contains the coordinates (2 integers) of the first point separated by comma.
Second line contains the coordinates (2 integers) of the second point separated by comma.

Output format:

A single line containing the coordinates (round to 2 decimal places) of the midpoint separated by comma. If the input is not following the mentioned format, display “INVALID”

Sample input:
4,5
6,7

Sample output:
5.00,6.00

7. An angle may be measured in degrees and minutes (e.g. ∠A = 80 degrees and 35 minutes). Write a program to find the sum of two angles. Define a class **Angle** to model an angle. It should have the following fields: ***degrees(integer)*** and ***minutes(integer).*** There should be a member function, ***angleSum(angle1,angle2)*** to calculate the sum of the two **Angle** objects passed to it and assign the result to a new **Angle** object.

Input format:
Input consists of multiple lines.
First line contains the values for ***degrees*** and ***minutes*** of the first angle separated by space.
Second line contains the values for ***degrees*** and ***minutes*** of the second angle separated by space.

Output format:
A single line containing the sum of angles (degrees and minutes separated by space).

Sample input:
70 35
50 40

Sample output:
121 15

8. Each student learns 5 subjects (English, Hindi, Maths, Science, Social Studies). Define a class **Student.** Each object of the class **Student** should contain the following fields: ***studentName(string)***, ***rollNo(integer), englishScore(integer), hindiScore(integer), mathsScore(integer), scienceScore(integer), ssScore(integer)***. Assume that the ***rollNo*** of each student is unique and in the range [1001 to 1100], maximum score for each subject is 100, and that the ***studentName*** has only one part. The three major operations to be performed are:
    - ***addStudent() -*** Read the student details to the respective fields.
    - ***subjectTopper() -*** Read the subject mentioned and display the ***studentName*** and ***rollNo*** of the student/students with maximum marks in the subject. Subject can be given through subject code [101: English, 102: Hindi, 103: Maths, 104: Science, 105: Social Studies]
    - ***classTopper() -*** Display the ***studentName*** and ***rollNo*** of the student/students with the maximum total marks.

Input/Output format:
Input consists of multiple lines. Each line may contain a character from {a,s,c,t}

Character a: to add student details. The next line contains values for ***studentName***, ***rollNo, englishScore, hindiScore, mathsScore, scienceScore, ssScore*** separated by spaces.

Character s: to display the ***studentName*** and ***rollNo*** of the student/students with maximum marks in the subject mentioned in the next line. The next line contains the subject code specifying the subject. Output must contain ***studentName*** and ***rollNo*** separated by space (one student per line if there are more than one student in the ascending order of their ***rollNo)***.

Character c: to display the ***studentName*** and ***rollNo*** of the student/students with the maximum total marks. Output must contain ***studentName*** and ***rollNo*** separated by space (one student per line if there are more than one student in the ascending order of their ***rollNo)***.

Character t : to terminate the program.

Sample input:

a
Hima 1005 85 95 100 71 90
a
Aman 1010 99 100 100 95 98
a
Bhama 1001 79 75 85 67 77
a
Kavya 1025 55 89 75 92 99
s
103
c
s
105
t

Sample output:

Hima 1005
Aman 1010
Aman 1010
Kavya 1025