**Easy:**

**Problem 1: Write a java program that will ask user to input a string (containing exactly one word). Then your job is to sort alphabetically all the letters in it. For simplicity, you can consider all letters will be either capital or small.**

**Example:** Suppose, user inputs “BANGLADESH”. Then you will sort all the letters in it alphabetically.

So output will be “AABDEGHLNS”.

**Input:**

BANGLA

**Output:**

AABGLN

**Input:**

BOOK

**Output:**

BKOO

**Problem 2: Given a line as keyboard input in small letters, print the next alphabet in sequence for each alphabet found in the input.**

**Input:** abcd **Output:** bcde

**Input:** the cow **Output:** uif dpx

**Problem 3: Given a line as keyboard input in small letters, do the opposite of Problem 2 Input:** bcde **Output:** abcd

**Input:** uif dpx **Output:** the cow

**Medium:**

**Problem 1: Write a java program that will ask user to input a string (containing exactly one word). Then your job is to print subsequent substring of given string.**

**Input:**

BANGLA

**Output:**

B

BA

BAN

BANG

BANGL

BANGLA

**Input:**

DREAM

**Output:**

D

DR

DRE

DREA

DREAM

**Problem 2: Write a program that will ask the user to input a word where each of it’s alphabets is unique and has not been entered before by the user. If the user does input a word which consists of duplicate alphabets, the program should reject the user’s input and ask for another word.**

**Input:**

Radeon

**Output:**

You entered Radeon.

**Input:** Hello

**Output:**

‘l’ has been counted 2 times in the word “hello”..

Please enter another word.

**Problem 3:** Write a program which takes **TWO string inputs** **(containing exactly one word in each string)** from the user. **Concatenate those two strings** with a single space in between them. **Generate a number** which is the **sum of all the letters in that concatenated string** (you have to avoid the value of that space), Where A = 65, Z = 90, a = 97 and z = 122. **Your task is to print that concatenated string and the number generated from that string.** (You are not allowed to use “+” operator to concatenate.)

**Sample Input:**

Hello

World

**Sample Output:**

Hello World 1020

**Sample Input:**

Java

CHOWDHURY

**Sample Output:**

Java CHOWDHURY

1087

**Problem 4 (Remove duplicates)**

Given a string, create a new string with all the consecutive duplicates removed. For example,

ABBCCCCCBBAB becomes ABCBAB.

**Sample Input:**

AAABBBBCDDBBECE

**Sample Output:**

ABCDBECE

**Hard:**

**Problem 1: 3­Divisibility**

Write a program that prints whether a given number is divisible by 3. The number can be huge (may contain up­to 1000 digits).

(**Hint**: A number is divisible by 3 if the **sum of its digits** is divisible by 3.)

**Input:**

141414141414141414

**Output:**

141414141414141414 is divisible by 3.

**Input:**

2368049403457746389253849640734644954763 **Output:**

2368049403457746389253849640734644954763 is divisible by 3.

**Input:**

557629788989463427894562342368049403457746389253849640734644954763 **Output:**

557629788989463427894562342368049403457746389253849640734644954763 is divisible by 3.

**Input:**

45342958456466468984465455844645876499644694466647899846655487965894564627894562342368049

40345774638 **Output:**

45342958456466468984465455844645876499644694466647899846655487965894564627894562342368049 40345774638 is not divisible by 3.

**Problem 2:** Write a program which takes **TWO string inputs (containing exactly one word in each string)** from the user. First input will be the **name of a programming team** and Second input will be the **name of a Coach** of that team. Both the name of the team and the name of the coach are **converted into a number** in the following manner: the final number is just the product of all the letters in the name, where **“A” is 1** and **“Z” is 26**. For instance: the team name “EAGLE” would be 5\*1\*7\*12\*5 = 2100 and the coach name “JAMES” would be 10\*1\*13\*5\*19 = 12350.

If **the team’s number mod 14 is less than the coach’s number mod 14**, then your program should print “**I Am Happy With My Coach**”. Otherwise, your program should print “**I Am Sad With My Coach**”. (Remember that "a mod b" is the remainder left over after dividing a by b; 34 mod 10 is 4.)

The name of the team and the coach will be a **string of capital letters** with **no spaces or punctuation, 1 to 6 characters long**.

**Sample Input:**

EAGLE JAMES

**Sample Output:**

I Am Happy With My Coach

**Sample Input:**

PRIME JOHN

**Sample Output:**

I Am Sad With My Coach

**Problem 3: (Word Reverse)**

Suppose you have a String and a CAPITAL letter in that indicates ending of a word. For example, if you have **wElovEbangladesH** where E, E and H indicates end of the words wE, lovE and bangladesH respectively. You need to reverse each word (as you know where it ends). Don’t reverse the String as a whole. To illustrate, if we give **wElovEbangladesH** as input output should be **EwEvolHsedalgnab.**  See wE became Ew, lovE became Evol and so on. (Input will contain only alphabetic characters)

**Sample Input:** merrYeatSpieS

**Sample Output:** YrremStaeSeip

**Sample Input:** programminGiSfuN

**Sample Output:**

GnimmargorpSiNuf

**Problem 4: (Mystery words)**

Write a program that takes a number and a String and then each letter in the String is replaced by a letter number of positions down the alphabet. For example, with number=3, A would be replaced by D, B would become E, and so on. (finally Z becomes C). Input will contain upper­case letters only.

**Sample Input:** 1

HELLOWORLD

**Sample Output:** IFMMPXPSME

**Sample Input:**

3

HELLOWORLD

**Sample Output:**

KHOORZRUOG

**Sample Input:**

4

HAPPYPEOPLE

**Sample Output:**

LETTCTISTPI