WEEK#1 - ASSIGNMENT 1

MS Connect

Vijay Sekhar G

Objectives

- Get comfortable with .NET.
- Start thinking more carefully.
- Solve problems in C# using built in libraries and advanced concepts.

Reasonable

1. Communicating with colleagues about problem problems in English (or some other spoken language).
2. Discussing the assignment material with others in order to understand it better.
3. Helping a colleagues identify a bug in his or her code, as by viewing, compiling, or running his or her code, even on your own computer.
4. Incorporating snippets of code that you find online or elsewhere into your own code, provided that those snippets are not themselves solutions to assigned problems and that you cite the snippets' origins.
5. Sending or showing code that you've written to someone, possibly a colleagues, so that he or she might help you identify and fix a bug.

Deliverables

- Document your solutions in word file.
- Write proper comments for each line in your source code.
- Document the output of your program.
- Your program should address the problem, there should NOT be any deviations in output.

SECTION – A

①

Write a program to factor this set of 10 numbers:

```
1568349935119356465908794692834625420038747829567400460116971790883538085491 7
2433660664476917632490307814638672585613657858874527031531027860396126349167 7
3975579861259333036351503376851097779853481096525724985650532017750137021034 1
4595600740970155550030821307632684724439247467280375423212362873851418002579 7
5675056176538042651192726898139904120997378485591464985185187200571721664985 1
6430535609557825784794524984611307968323333228148007603857781150647873577291 7
7223274585173765708757820227614680395551723400986221779515851671926825791816 1
8039606817482324682147004188450160848820803218593802700721507537703882980985 9
9389886793895795772389466959828206666380770069972461140669448755991150537078 9
9994427728635642326608000381369596195236962602180745211262799013885988764524 9
```

Each of these:

- Is a 77-digit number less than 2^256.
- Is a semiprime (e.g., the product of exactly 2 primes).
- Has something in common with at least one other number in the set.

Thus, this challenge is not about general factoring of 256-bit semiprimes, but about factoring these semiprimes. It is a puzzle. There is a trick. The trick is fun.

It is possible to factor each of these numbers with surprising efficiency. Therefore, the algorithm you choose will make a much bigger difference than the hardware you use.

Rules

You may use any method of factoring. (But don't precompute the answers and just print them. You program should do actual work.)

②

A character string is said to have period k if it can be formed by concatenating one or more repetitions of another string of length k. For example, the string "abcabcabcabc" has period 3, since it is formed by 4 repetitions of the string "abc". It also has periods 6 (two repetitions of "abcabc") and 12 (one repetition of "abcabcabcabc").

Write a program to read a character string and determine its smallest period.

Input
A single character string of up to 80 non-blank characters.
Output
An integer denoting the smallest period of the input string.

Sample Input
HoHoHo
Sample Output
2

③

Nishant is bored with his mathematics homeworks. He takes a piece of chalk and starts writing a sequence of consecutive integers starting with 1 to N (1 < N < 10000). After that, he counts the number of times each digit (0 to 9) appears in the sequence. For example, with N = 13, the sequence is:

12345678910111213

In this sequence, 0 appears once, 1 appears 6 times, 2 appears 2 times, 3 appears 3 times, and each digit from 4 to 9 appears once. After playing for a while, Nishant gets bored again. He now wants to write a program to do this for him. Your task is to help him with writing this program.

Input
The input file consists of several data sets. The first line of the input file contains the number of data sets which is a positive integer and is not bigger than 20. The following lines describe the data sets. For each test case, there is one single line containing the number N.

Output

For each test case, write sequentially in one line the number of digit 0, 1,...9 separated by a space.

Sample Input

```
2
3
13
```

Sample Output

```
0 1 1 1 0 0 0 0 0 0
1 6 2 2 1 1 1 1 1 1
```

④

The Fibonacci word sequence of bit strings is defined as:

$$F(n) = \begin{cases} 0 & \text{if } n = 0 \\ 1 & \text{if } n = 1 \\ F(n-1) + F(n-2) & \text{if } n \geq 2 \end{cases}$$

Here + denotes concatenation of strings. The first few elements are:

n	$F(n)$
0	0
1	1
2	10
3	101
4	10110
5	10110101
6	1011010110110
7	101101011011010110101
8	10110101101101011010110110101101
9	1011010110110101101011011010110110101101011011010110101

Given a bit pattern p and a number n, how often does p occur in F(n)?

Input

The first line of each test case contains the integer n (0 ≤ n ≤ 100). The second line contains the bit pattern p. The pattern p is nonempty and has a length of at most 100 000 characters.

Output

For each test case, display its case number followed by the number of occurrences of the bit pattern p in F(n). Occurrences may overlap. The number of occurrences will be less than 2^63.

Sample Input

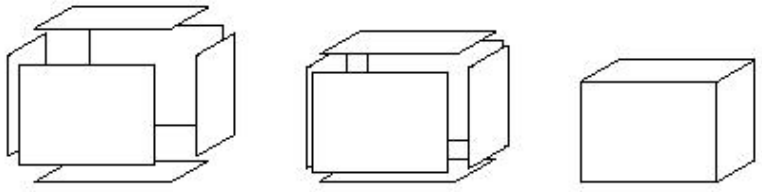
```
6
10
7
10
6
01
6
101
96
10110101101101
```

Sample Output

```
Case 1: 5
Case 2: 8
Case 3: 4
Case 4: 4
Case 5: 7540113804746346428
```

⑤

Soumitra works at a factory that produces heavy machinery. He has a simple job - he knocks up wooden boxes of different sizes to pack machinery for delivery to the customers.
Each box is a rectangular parallelepiped. Soumitra uses six rectangular wooden pallets to make a box. Each pallet is used for one side of the box.



Amar delivers pallets for Soumitra. Amar is not very smart and often makes mistakes - he brings Soumitra pallets that do not fit together to make a box. But Joe does not trust Ivan. It always takes a lot of time to explain Amar that he has made a mistake.

Fortunately, Amar adores everything related to computers and sincerely believes that computers never make mistakes. Soumitra has decided to use this for his own advantage. Soumitra asks you to write a program that given sizes of six rectangular pallets tells whether it is possible to make a box out of them.

Input

Input contains several test cases. Each of them consists of six lines. Each line describes one pallet and contains two integer numbers *w* and *h* ($1 < w; h < 10000$) -- width and height of the pallet in millimeters respectively.

Output

For each test case, print one output line. Write a single word `POSSIBLE' to the output console if it is possible to make a box using six given pallets for its sides. Write a single word `IMPOSSIBLE' if it is not possible to do so.

Sample Input

```
1345 2584
2584 683
2584 1345
683 1345
683 1345
2584 683
1234 4567
1234 4567
4567 4321
```

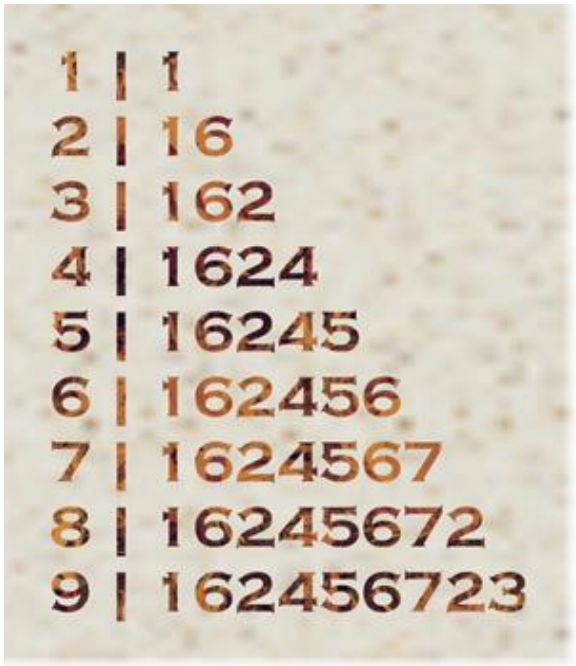
4322 4567
4321 1234
4321 1234

Sample Output

POSSIBLE
IMPOSSIBLE

⑥

Don't you think **162456723** very special? Look at the picture below if you are unable to find its specialty. (*a*/*b* means '*b* is divisible by *a*')



Given *n*, *m* (0 < *n* < *m* < 30), you are to find a *m*-digit positive integer *X* such that for every *i* (*n* < *i* < *m*), the first *i* digits of *X* is a multiple of *i*. If more than one such *X* exists, you should output the lexicographically smallest one. Note that the first digit of *X* should **not** be 0.

Input

The first line of the input contains a single integer *t* (1 < *t* < 15), the number of test cases followed. For each case, two integers *n* and *m* are separated by a single space.

Output

For each test case, print the case number and *X*. If no such number, print '-1'.

Sample Input

```
21
10
3 29
```

Sample Output

```
Case 1: 1020005640
Case 2: -1
```

⑦

A prime number is a number that has only two divisors: itself and the number one. Examples of prime numbers are: 1, 2, 3, 5, 17, 101 and 10007.

In this problem you should read a set of words, each word is composed only by letters in the range a-z and A-Z. Each letter has a specific value, the letter *a* is worth 1, letter b is worth 2 and so on until letter z that is worth 26. In the same way, letter A is worth 27, letter B is worth 28 and letter Z is worth 52.

You should write a program to determine if a word is a prime word or not. A word is a prime word if the sum of its letters is a prime number.

Input

The input consists of a set of words. Each word is in a line by itself and has L letters, where $1 < L < 20$.

Output

For each word you should print: 'It is a prime word.', if the sum of the letters of the word is a prime number, otherwise you should print: 'It is not a prime word'.

Sample Input

```
UFRN
contest
AcM
```

Sample Output

```
It is a prime word.
It is not a prime word.
It is not a prime word.
```

⑧

A Subsequence is a sequence obtained by deleting zero or more characters in a string. A Palindrome is a string which when read from left to right, reads same as when read from right to left. Given a string, find the longest palindromic subsequence. If there are many answers to it, print the one that comes lexicographically earliest.

Constraints

• Maximum length of string is 1000.
• Each string has characters ‘a’ to ‘z’ only.

Input

Input consists of several strings, each in a separate line.

Output

For each line in the input, print the output in a single line.

Sample Input

aabbaabb
computer
abzla
samhita

Sample Output

aabbaa
c
aba
aha

SECTION – B

①

Without changing the actual function, call it in such a way that it prints "True"

```
void Puzzle(out int x, out int y)
{
  x = 0;
  y = 1;
  Console.WriteLine(x == y);
}
```

②

In this programming puzzle, you are to produce Exceptions that can be caught but are thrown again at the end of the catch block.

```
try
{
    while(true)
        try
        {
            // you are only allowed to modify code between this try { } brackets
        }
        catch(Exception ex2) {  }
}
catch(Exception ex1)
{
    // your goal is to reach this catch block by modifying the code ...
    // in the inner try block above
    // You win if you reach this part and execute on of the following code lines
    Console.WriteLine("You won!"); // for C#
    // Or
    System.out.println("You won!"); // for Java
}
```

You can freely put code before and after this snippet.
The shortest code to reach the outer catch block wins.

③

Given the following C# program outputting False, inject a 'malicious' line of code such that the program outputs True.

```
class Program
{
   static void Main()
   {
      System.Console.Write("False");
      ;
   }
}
```

Your answer should consist of a string that replaces the second semicolon, and causes the program to output True, the whole True and nothing but True (not even a newline). The shortest answer wins.

④

Give an example of a situation where the C# method below returns false:

```
public class Giraffe : Animal
{
  public bool Test()
  {
    return this is Giraffe;
  }
}
```

Rules: The code lines above must not be changed in any way, but you would put the code into an application so that the "project" compiles and the method is called. When run, the method must return false.

⑤

For example, storing "www.google.com" as a string in .NET will result in the object being encoded as UTF-16, and it will consume twice as much memory as plain ASCII... but for the purpose of storing DNS names, ASCII is overkill.

The only characters that are needed are:

A..Z (case insensitive)
0..9
Hyphen
underscore
period

Asterisk (not legal DNS, but used in ACLs such as *.google.com) always as a leading character, never elsewhere.

A total of 40 characters.. Which fits within single byte with plenty room to spare. What is the most compact way to store a string of characters in memory and maximizing in memory storage?

⑥

Write C# Program to Convert a 2D Array into 1D Array.

Input : Given 2-D Array(Matrix) is :

1 4 3
7 3 8

Output : Converted 1-D Array is :

1
4
3
7
3
8

⑦

Write C# Program Displays the Abbreviation of a Text.

Input: Wipro Technologies Ltd

Output: W.T.L

⑧

Write C# Program Reads a String and find the Sum of all Digits in the String.

Input:

Enter the Length of the sentence :
6
Enter the string1 containing both digits and alphabet :
SAN193

Output

NO. of Digits in the string1 = 3
Sum of all digits = 13

⑨

Write a C# Program to convert structure to binary array and back to structure.

⑩

1. Write C# Program Implements Use of Indexers. Indexers allow instances of a class or struct to be indexed just like arrays.

2. Write C# Program to Combine Two Delegates. Here when the multicast delegate is called, it invokes the delegates in the list, in order.

3. Write C# Program to demonstrate Generic Delegate.

4. Write C# Program Converts Feet to Inches using Delegates.

5. Write C# Program Finds Largest Element in a Matrix.

6. Write C# Program Implements Stack with Push and Pop operations: use interfaces, classes.

7. Write a C# Program to find the Frequency of the Word "is" in a given Sentence.

8. Write a C# Program to generate Strings randomly.

9. Write a C# Program to lists all Substrings in a given String.

10. Write a C# Program to demonstrate Properties of the Interface.

11. Write a C# Program to demonstrate IDumpable Interface.

12. Write a C# Program to demonstrate IDisposable Interface.

13. Write a C# program to create custom list implementing IList, IEnumerable,IDisposable.

14. Write a C# Program to illustrate handling an event declared in an interface

15. Write a C# Program Creates Anonymous Method.

16. Write a C# Program to demonstrate Method Hiding.
17. Write a C# Program Creates Obsolete Class.
18. Write C# Program finds Product of 2 Numbers using Recursion.
19. Write a C# Program to demonstrate Boxing Operations.
20. Write a C# Program to demonstrate Multilevel Inheritance.
21. Write a C# Program to demonstrate Hierarchical Inheritance.
22. Write a C# Program to demonstrate Multilevel Inheritance.
23. Write a C# Program to demonstrate Partial Class.
24. Write a C# Program to demonstrate Sealed Class.
25. Write a C# Program to demonstrate Dictionary, Hash Table Class.
26. Write a C# Program to demonstrate Generic Classes and Constraints.
27. Write a C# Program to demonstrate Structure and Enum.