

Daily Challenge

Happy Coding from necse



SkillRack

Prime Sentence

The program must accept a string **S** representing a sentence as the input. The program must find the sentence value of S based on the following conditions.

- All **26** alphabets from **a** to **z** or from **A** to **Z** have values are from **1** to **26** respectively.
- All **10** digits from **0** to **9** have values are from **0** to **9** respectively.
- All the other characters have the value **0**.
- The sentence value is obtained by summing up all the numeric values of every single character in S.

Then the program must print the output based on the following conditions.

- If the sentence value is a prime number, then the program must print "**Prime**".
- If the sentence value is a prime number after removing exactly one word in the sentence, then the program must print "**Almost Prime**".
- Otherwise, the program must print "**Composite**".

Boundary Condition(s):

1 <= Length of S <= 1000

Input Format:

The first line contains S.

Output Format:

The first line contains a string value based on the given conditions.

Example Input/Output 1:

Input:

Lift dog

Output:

Prime

Explanation:

Lift -> (12+9+6+20) -> **47**

dog -> (4+15+7) -> **26**

So the sentence value is **73** (47+26), which is a **prime number**.

Hence the output is Prime.

Example Input/Output 2:

Input:

Is 24 the answer?

Output:

Almost Prime

Explanation:

Is -> (9+19) -> **28**

24 -> (2+4) -> **6**

the -> (20+8+5) -> **33**

answer? -> (1+14+19+23+5+18+0) -> **80**

So the sentence value is **147** (28+6+33+80), which is **not a prime number**.

If the word "**answer?**" is **removed**, the sentence value becomes **67**, which is a **prime number**.

Hence the output is Almost Prime.

Example Input/Output 3:

Input:

Hello World!

Output:

Composite

Max Execution Time Limit: 50 millisecs

Ambiance

Java (12.0)



```

1 import java.util.*;
2 public class Hello {
3
4     public static void main(String[] args) {
5
6         Scanner sc = new Scanner(System.in);
7         String str[] = sc.nextLine().toLowerCase().split(" ");
8
9         List<Integer> li = new LinkedList<>();
10        int total=0;
11
12        for(String i:str){
13            int temp = getNum(i);
14            li.add(temp);
15            total+=temp;
16        }
17
18        //System.out.println(isPrime(4));
19
20        if(isPrime(total)){
21            System.out.println("Prime");
22            return;
23        }
24
25
26        for(int i:li){
27            if(isPrime(total-i)){
28                System.out.println("Almost Prime");
29                return;
30            }
31        }
32
33        System.out.println("Composite");
34
35    }
36    public static int getNum(String str){
37        int res=0;
38        for(char i:str.toCharArray()){
39            if(i>='a' && i<='z') res+=i-'a'+1;
40            if(i>='0' && i<='9') res+=i-'0';
41        }
42        return res;
43    }
44
45    public static boolean isPrime(int x){
46
47        if(x<2) return false;
48
49        boolean flag=false;
50
51        for(int i=2;i<x;i++){
52            if(x%i==0) return false;
53        }
54        return true;
55    }
56 }

```

1912067@nec

Please wait while we run the program

