ACSL

American Computer Science League

**All-Star #9**

**008 -2013 - 2014**

**Lorenz Cipher Machine Decode**

**PROBLEM**: This problem is a continuation of Program #8 and based on an activity from the Center for Innovation in Mathematics Teaching. This time you will be given an encoded message and a procedure to determine the most likely Key #1 starting position to use to decipher the message. HISTORICAL NOTE: 99 was commonly used to separate words. That lead to the breaking of the code once it was discovered that the 99's produced a high number of slashes (/) in the encoded message.

1. Given an encoded message string Z of length n, XOR the binary values of its adjacent characters (1st and 2nd characters, 2nd and 3rd characters, 3rd and 4th characters...) using the chart from Program #8 and then convert back to alpha numeric characters to form a new string ∆Z of length n-1.

2. For each starting position in the Key #1 array (1 - 14) form a sequence of characters K of length n. Find ∆K for each in the manner given above.

3. XOR each of the 14 ∆K's with ∆Z to form 14 new character strings ∆Z XOR ∆K1-14.

4. Determine by starting position number which of the 14 resulting character strings produces the most slashes (/).

5. Using that number as the starting position in the KEY #1 array form a character string K of length n and XOR it with Z. Then XOR that result with the character string of length n formed by a given starting position of the Key #2 array. The converted alpha numeric string is the decoded message.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| A | B | C | D | E | F | G | H | I | J | K | L | M | N |

Key #1

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| A | A | B | B |

Key #2

EXAMPLE:

1. Given Z = UDZDMR+J then ∆Z = (U XOR D = 11100 XOR 10010 = 01110 = C) COOYPZT.

2. For Key#1 starting position 1, K1 = ABCDEFGH and ∆K1 = GQU3NQC. When the starting position is 2, then K2= BCDEFGHI and ∆K2 = QU3NQCL.

3. ∆Z XOR ∆K1  = COOYPZT XOR GQU3NQC = HK8XGIV. There are no slashes.

∆Z XOR ∆K2 = B8TBE84

∆Z XOR ∆K3 = DTH4OAX  
 ∆Z XOR ∆K4 = IHK+9MH

∆Z XOR ∆K5 = 4KPU+YF

∆Z XOR ∆K6 = BPROLNV

∆Z XOR ∆K7 = /RYZJ8/ .....etc.

4. When the Key #1 starting position = 7 there are 2 slashes in ∆Z XOR ∆K7. That is the greatest number of slashes produced.

|  |
| --- |
|  |

5. UDZDMR+J (Z)

XOR GHIJKLMN (Key#1 with starting position 7)

XOR BBAABBAA (Key#2 with given starting position 3)

--------------------------

= 99HERE99

**INPUT**: There will be 5 lines of input. Each line will have one encoded message string (Z) and a starting position to use with the Key #2 array.

**OUTPUT**: For each input line print the value of the Key #1 starting position that produces the character string with the most slashes from the results of the 14 ∆Z XOR ∆K operations and then print the decoded message, the result of the operations listed in step #5 above. We guarantee that the most slashes count will be unique.

SAMPLE INPUT SAMPLE OUTUT

1 XWZOFTRZ9, 1 1. 7

2. VD4/9OECN3, 1 2. 99CREEK99

3. +PATTZ9BXGE+UD, 3 3. 2

4. LKVAHI+VBVXWKIRHEZ, 3 4. 99LORENZ99

5. CIX4BA94TYZVMXHM, 1 5. 9

6. 99KEY99ARRAY99

7. 11

8. 99ENCIPHER99THIS99

9. 4

10. 99DECIPHER99IT99

ACSL

American Computer Science League

**All-Star #9**

**008 -2013 - 2014**

**Lorenz Cipher Machine\_Decode**

TEST DATA

TEST INPUT TEST OUTPUT

1. MTTPWNF+JF9V, 3 1. 5

2. 9VUZUVUD, 1 2. 99DENVERCO99  
3. CI34L/ZNT+Z, 1 3. 1

4. D+MFE9J9ILCI/4X9PL, 3 4. 99TEST99

5. K+8TYDC3/8L+PHX9Z9, 4 5. 4

6. 99MESSAGE99

7. 8

8. 99HERE99IS99HERE99

9. 12

10. 99TRY99THIS99ONE99