ACSL

American Computer Science League

**All-Star #8**

**008 -2013 - 2014**

**Lorenz Cipher Machine Encode**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A | 11000 | B | 10011 | C | 01110 | D | 10010 | E | 10000 | F | 10110 |
| G | 01011 | H | 00101 | I | 01100 | J | 11010 | K | 11110 | L | 01001 |
| M | 00111 | N | 00110 | O | 00011 | P | 01101 | Q | 11101 | R | 01010 |
| S | 10100 | T | 00001 | U | 11100 | V | 01111 | W | 11001 | X | 10111 |
| Y | 10101 | Z | 10001 |  |  |  |  |  |  |  |  |
| + | 11011 | / | 00000 | 9 | 00100 | 8 | 11111 | 4 | 01000 | 3 | 00010 |

**PROBLEM**: This problem is adapted from an activity from the Center for Innovation in Mathematics Teaching. The Lorenz Cipher was an encoding/decoding device used during World War II. Just using a one-to-one conversion would not be very sophisticated, so a system of key letters and binary calculations was added to complicate the process.

Key #1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |

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

Key #2

The encoding and decoding process is the same. It begins by selecting/knowing a starting point in the Key #1 array and a starting point in the Key #2 array. The binary representations of the sequence of those letters are XORed to the binary representations of the sequence of letters to be encoded or decoded. The key positions circle around as if they were on a wheel when they reach the end of the array. The 9 in the conversion table is used as a space between words. The other characters in the bottom row are not used in encoding messages but could be used in decoding messages . Using starting position 1 for Key# 1and 4 for Key #2, the word THE is encoded as follows:

T = T xor A (Key#1 position 1) xor B (Key#2 position 4) = 00001 xor 11000 xor 10011= 01010 = R

H = H xor B (position 2) xor A (position 1) = C

E = E xor C (position 3) xor A (position 2) = N

**INPUT**: There will be 10 lines of input. Each line will contain the starting position to use in the Key #1 array, the starting position to use in the Key #2 array and the message to encode or decode. The first 5 lines will be encoding problems and the last 5 lines will be decoding problems.

**OUTPUT**: For each line of input print the encoded or decoded message.

SAMPLE INPUT SAMPLE OUTUT

1. 1, 1, SECRET 1. S+BGAV

2. 1, 3, COMPUTER 2. HOZM89OX

3. 2, 4, DENVER 3. DNIIYW

4. 5, 2, ACSL9CONTEST 4. EGISEMCXKH8R

5. 10, 4, CHERRY9CREEK 5. MOTK8YVBGAKN

6. 1, 1, S+BGAV 6. SECRET

7. 1, 3, HOZM89OX 7. COMPUTER

8. 2, 4, DNIIYW 8. DENVER

9. 5, 2, EGISEMCXKH8R 9. ACSL9CONTEST

10. 10, 4, MOTK8YVBGAKN 10. CHERRY9CREEK

ACSL

American Computer Science League

**All-Star #8**

**008 -2013 - 2014**

**Lorenz Cipher Machine\_Encode**

TEST DATA

TEST INPUT TEST OUTPUT

1. 1, 1, FIRST 1. FMXYL  
2. 2, 4, FIRST 2. FJ/X9

3. 11, 2, FIRST 3. EFKRT

4. 13, 3, FIRST9PLACE 4. 3WR8UHHM/A9

5. 7, 1, AIRPORT 5. GZY9H+Y

6. 6, 2, /+8X4Y4U 6. COLORADO

7. 10, 3, Z8KBT4T 7. ADVISOR

8. 12, 2, U98F+B/ 8. PERFECT

9. 9, 4, DDII9ET9L9GV4 9. PERFECT9SCORE

10. 2, 2, 3KGA4LDJC/8VS 10. LORENZ9CIPHER