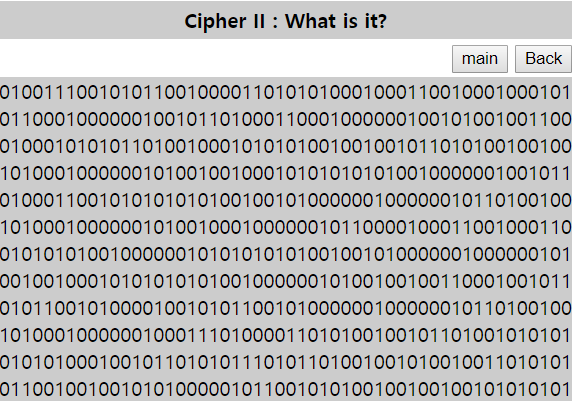
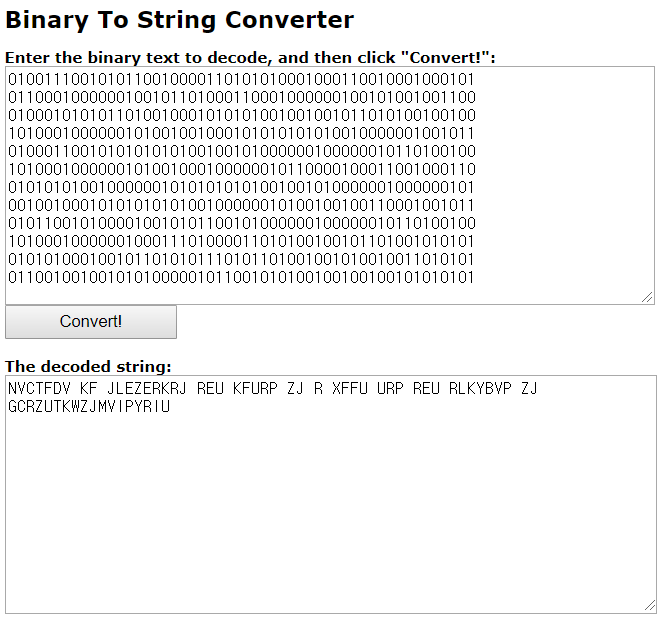
< 19번 워게임 풀이문서 >

김구민

다음과 같이 0과 1로 이루어진 문장들이 빼곡히 나열되어 있다.

이진수를 문자로 변환시켜야 할 것 같은 느낌이 강하게 들었다.

<http://string-functions.com/binary-string.aspx> 사이트를 참고하여 이진수를 문자로 변환시켰다.



NVCTFDV KF JLEZERKRJ REU KFURP ZJ R XFFU URP REU RLKYBVP ZJ GCRZUTKWZJMVIPYRIU

띄어쓰기가 있는 것으로 보아 인증키가 아닌 것 같고 여기서 또 복호화를 시켜야 할 것 같다.

그러나 암호에 대해서 아는 게 많이 없어서 아무리 봐도 무슨 암호 알고리즘을 써서 풀어야 할지 감이 잡히지 않아서 구글링해보니 문자열 사이의 공백과 전혀 알 수 없는 문자가 아닌 알파벳으로 이루어져 있는 점으로 미뤄보아 치환암호일 확률이 높다는 글을 읽고 알게 되었다.

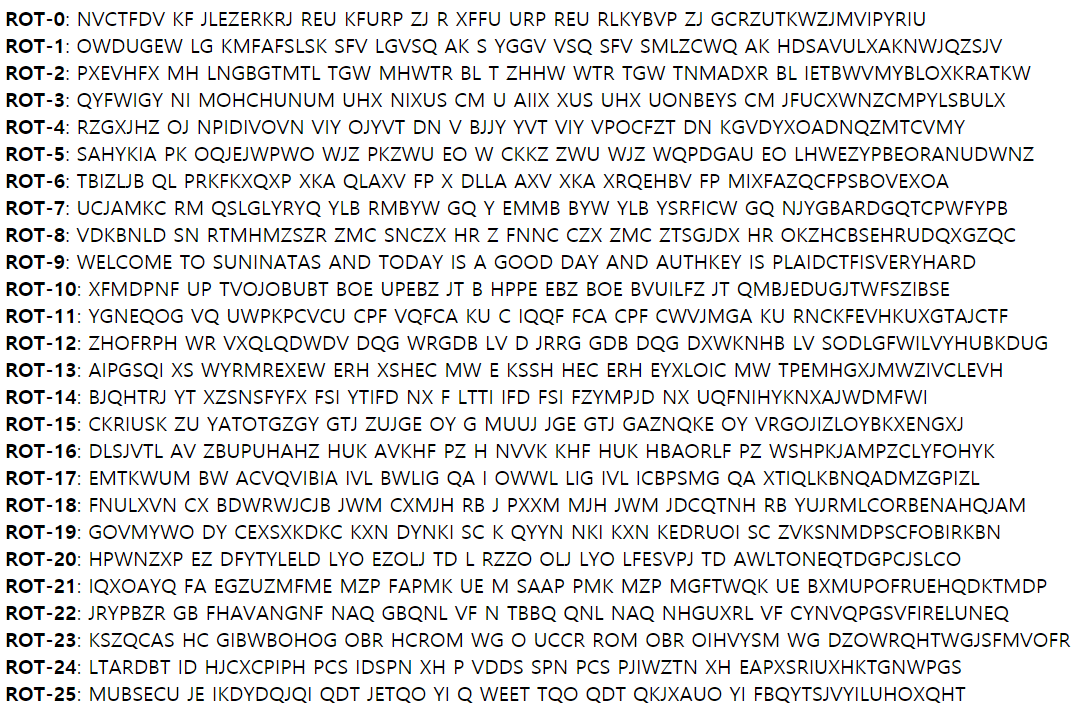
치환 암호는 일정한 법칙에 따라 평문의 문자 단위를 다른 문자 단위로 치환하는 암호화 방식

문자 단위는 한 문자 또는 비트가 될 수 있다.

대표적인 치환 암호의 종류로 카이사르 암호(시저)를 들 수 있다.

평문을 일정 거리만큼 떨어진 다른 알파벳으로 평행 이동해서 교환하는 방식

구글링하니 시저암호 복호화 사이트가 많았는데 그 중 하나를 참고해서 다음의 결과를 얻을 수 있었다.



WELCOME TO SUNINATAS AND TODAY IS A GOOD DAY AND AUTHKEY IS PLAIDCTFISVERYHARD

<Incoding>

package suninatas\_19\_Incode;

import java.util.\*;

public class Main {

public static void main(String[] args) {

Scanner s = new Scanner(System.in);

ArrayList list = new ArrayList();

String ch = "";

int num = 0;

System.out.print("암호화시킬 문자를 입력하시오 : ");

ch = s.nextLine();

System.out.print("암호키를 설정하시오 : ");

num = s.nextInt();

System.out.println();

for (int j = 1; j <= num; j++) {

for (int i = 0; i < ch.length(); i++) {

if (!Character.isDigit(ch.charAt(i))) {

if(ch.charAt(i) != ' ') {

char ch\_ = (char)(((int)ch.charAt(i)) + j);

list.add(ch\_);

}else {

list.add(' ');

}

} else

break;

}

for(int i = 0; i < ch.length(); i++) {

System.out.print(list.get(i));

}

System.out.println();

list.removeAll(list);

}

}

}

<Decoding>

package suninatas\_19\_Decode;

import java.util.ArrayList;

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner s = new Scanner(System.in);

ArrayList list = new ArrayList();

String ch = "";

int num = 0;

System.out.print("복호화시킬 문자를 입력하시오 : ");

ch = s.nextLine();

System.out.print("복호키를 설정하시오 : ");

num = s.nextInt();

System.out.println();

for (int j = 1; j <= num; j++) {

for (int i = 0; i < ch.length(); i++) {

if (!Character.isDigit(ch.charAt(i))) {

if(ch.charAt(i) != ' ') {

char ch\_ = (char)(((int)ch.charAt(i)) - j);

list.add(ch\_);

}else {

list.add(' ');

}

} else

break;

}

for(int i = 0; i < ch.length(); i++) {

System.out.print(list.get(i));

}

System.out.println();

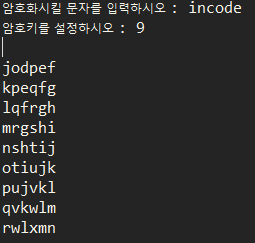
list.removeAll(list);

}

}

}

Incoding 실행 결과



================================================================

Decoding 실행 결과

