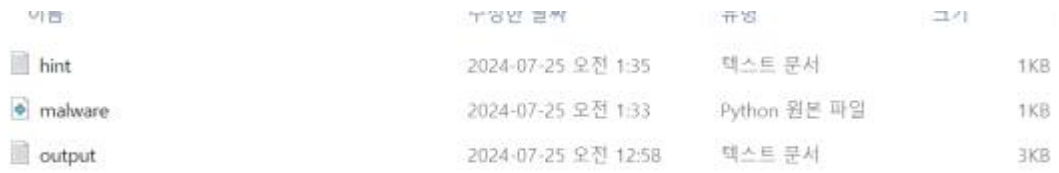
song 문제풀이



이름	수정한 날짜	유형	크기
hint	2024-07-25 오전 1:35	텍스트 문서	1KB
malware	2024-07-25 오전 1:33	Python 원본 파일	1KB
output	2024-07-25 오전 12:58	텍스트 문서	3KB

받은 압축파일을 열어보니 텍스트 문서 두 개와 파이썬 코드가 나왔다.

malware.py

C: > Users > ykhee > Desktop > song > malware.py > ...

```python
import os
import random

def get_seed(size):
    return int(os.urandom(size).hex(), 16)

input = None
output = ""

salt = get_seed(16)
random.seed(salt)

crypto = "fedcba9876543210"
malware = list(crypto)
random.shuffle(malware)
malmware = ''.join(malware)

with open("input.txt", "r") as lyrics_file:
    input = lyrics_file.read()

for char in input:
    encoded_char = (bytes(char.encode()).hex())
    output += malware[crypto.index(encoded_char[0])]
    output += malware[crypto.index(encoded_char[1])]


with open("output.txt", "w") as result_file:
    result_file.write(str(output))

```

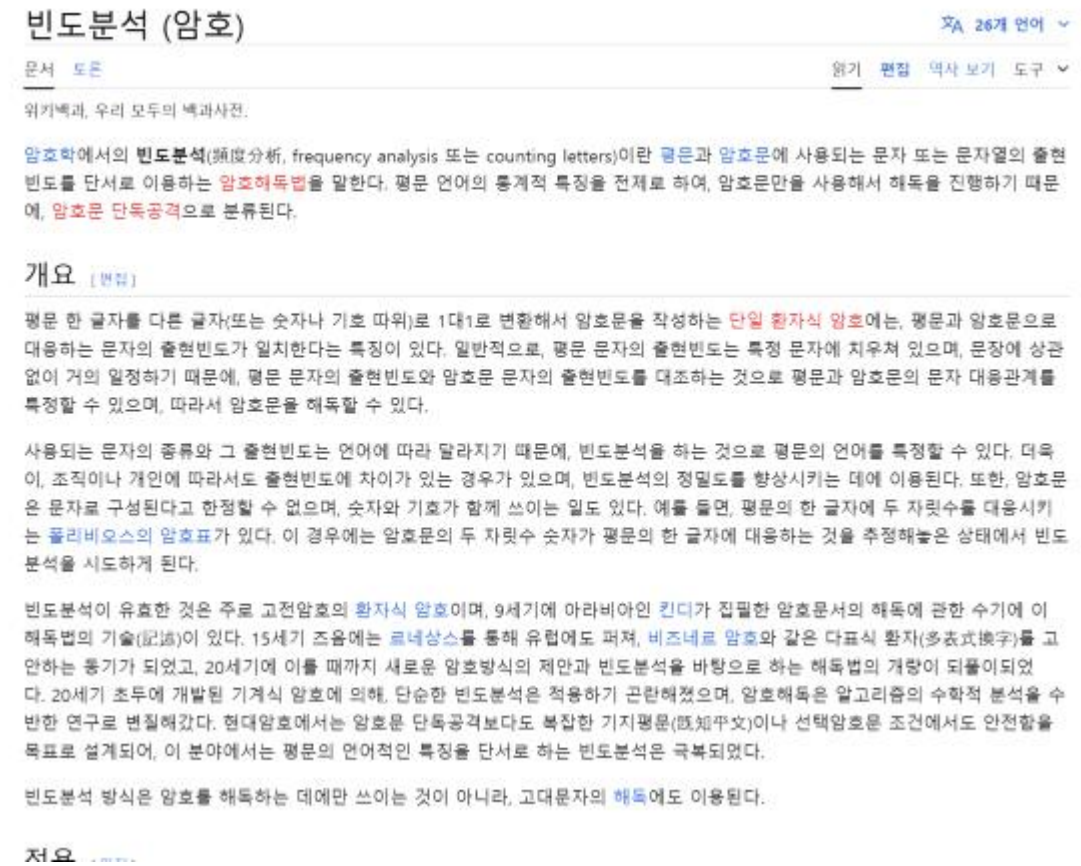
열어보니 이렇게 나왔는데 우선 코드 풀이를 해봤다.

이 코드가 input.txt 파일을 변환해서 output.txt 파일을 생성하는 것 같은데
받은 압축파일에는 input 파일은 없고 output 파일만 있었다.

58801d1715dc1586808b8117dc88801d1715dc50d18f83dc178310dc838a8adc158684dc118e1d8517dc808b17808584dc8f10dc86848385dc50d18fdc88801d8485dc141cdc838b85dc15801d848
5dc8e88dc158684dc118310dc45868315dc1586808b8117dc86831884dc8d84848bdc8e86dc8e8e86dc458684dc118310dc1586815dc1586808b8117dc86831884dc8d84848bdc8e86dc8e8e86c
24784878e8b85dc1586808b81dc1784878e8b85dc558e8bd115dc108e14dc15848a8adc8f84dc41868315dc108e14dc1586808b89dc15868315dc50dc878e148a85dc8d84dc50d18fdc158684dc8
e8b84dc8315dc158684dc1783808adc50d18fdc158684dc8f8317158841ddc8e88dc8f10dc178483dc8e86dc8e8e86dc458684dc8f8317158841ddc8e88dc8f10dc178483dc8e86dc8e8e86c250dc118
317dc8d1d8e89848bdc881d8e8fdc83dc108e148b81dc838184dc4583898088b81dc8f10dc17148a89808b81dc158edc158684dc8f8317178417dc411d8015808b81dc8f10dc1c8e848f17dc888e1d
dc158684dc888411dc45868315dc8a8e8e89dc8315dc8f84dc158e8e89dc158edc8f84dc47868e8e89dc158edc8f84dc8884848a808b81dc8f84c247808b81808b81dc881d8e8fdc8684831d158387
8684dc881d8e8fdc158684dc1c83808bdc4583898088b81dc8f10dc8f8417178838184dc881d8e8fdc158684dc1848408b17dc471c84838980808b81dc8f10dc8a8417178e8bdc881d8e8fdc158684dc8d
1d83808bdc47848480808b81dc158684dc8d848314151 0dc15861d8e148186dc158684c24c83808bdc408e14dc8f838584dc8f84dc83dc408e14dc8f838584dc8f84dc83dc8d848a80841 8841ddc8d8
48a80841 8841ddc4c83808bdc408e14dc8d1d848389dc8f84dc858e118bdc538b85dc8d14808a85dc8f84dc141cdc8d848a80841 8841ddc8d848a80841 8841ddc4c83808bdc5e86dc8a8415dc1586
84dc8d148a8a841517dc888a10dc8e86dc8a8415dc1586848fdc1d83808bdc5f10dc8a808884dc8f10dc8a8e1884dc8f10dc851d801884dc8015dc87838f84dc881d8e8fdc4c83808bc2408e14dc8f8
38584dc8f84dc83dc408e14dc8f8385 84dc8f84dc83dc8d848a80841 8841ddc8d848a80841 8841dc2458601d85dc1586808b8117dc1586801d85dc47848b85dc83dc1c1d8310841ddc158edc1586
84dc8e8b8417dc141cdc838d8e1884dc538a8adc158684dc868315 84dc158688315dc108e14d11884dc8684831d85dc568317dc15141d8b8485dc108e141ddc171c801d8015dc158edc83dc858e18
84dc8e86dc8e8e86dc408e141ddc171c801d8015dc141cdc838d8e1884dc8e86dc8e8e86c250dc118317dc87868e89808b81dc808bdc158684dc871d8e1185dc5d14808a85808b81dc8f10dc1d83
808bdc141cdc808bdc158684dc878a8e1485dc58838a8a808b81dc8a808984dc8317868417dc158edc158684dc811d8e148b85dc568e1c808b81dc8f10dc8884848a808b8117dc1586841 0dc118e
148a85dc851d8e118bdcc25d1415dc1586841 0dc8b8418841ddc858085dc8418841ddc8a80188485dc548d8d808b81dc838b85dc888a8e11808b81dc508b86808d80158485dc8a8c0f80158485dc
45808a8adc8015dc8d1d8e8984dc141cdc838b85dc8015dc1d83808b8485dc858e118bdc5015dc1d83808b8485dc858e118bc2c2585a5351dc8017dc5f14177387dc7317dc83dc817c8e85dc8d8e
7c1715dc887c1ddc177c8a18738b81dc574558dc1c1d7c8d73848b8b17

뭔가 이렇게 암호로 변환한 파일인 것 같았다.

이렇게 변환하기 위해서 정한 규칙 코드인 것 같은데 잘 모르겠어서 힌트를 열어봤다.



빈도분석 (암호)

26개 언어

문서 토론 읽기 편집 역사 보기 도구

위키백과, 우리 모두의 백과사전.

암호학에서의 **빈도분석**(頻度分析, frequency analysis 또는 counting letters)이란 평문과 암호문에 사용되는 문자 또는 문자열의 출현 빈도를 단서로 이용하는 암호해독법을 말한다. 평문 언어의 통계적 특징을 전제로 하여, 암호문만을 사용해서 해독을 진행하기 때문에, 암호문 단독공격으로 분류된다.

개요 [편집]

평문 한 글자를 다른 글자(또는 숫자나 기호 따위)로 1대1로 변환해서 암호문을 작성하는 단일 환자식 암호에는, 평문과 암호문으로 대응하는 문자의 출현빈도가 일치한다는 특징이 있다. 일반적으로, 평문 문자의 출현빈도는 특정 문자에 치우쳐 있으며, 문장에 상관없이 거의 일정하기 때문에, 평문 문자의 출현빈도와 암호문 문자의 출현빈도를 대조하는 것으로 평문과 암호문의 문자 대응관계를 특정할 수 있으며, 따라서 암호문을 해독할 수 있다.

사용되는 문자의 종류와 그 출현빈도는 언어에 따라 달라지기 때문에, 빈도분석을 하는 것으로 평문의 언어를 특정할 수 있다. 더욱이, 조직이나 개인에 따라서도 출현빈도에 차이가 있는 경우가 있으며, 빈도분석의 정밀도를 향상시키는 데에 이용된다. 또한, 암호문은 문자로 구성된다고 한정할 수 없으며, 숫자와 기호가 함께 쓰이는 일도 있다. 예를 들면, 평문의 한 글자에 두 자릿수를 대응시키는 폴리비오스의 암호표가 있다. 이 경우에는 암호문의 두 자릿수 숫자가 평문의 한 글자에 대응하는 것을 추정해놓은 상태에서 빈도분석을 시도하게 된다.

빈도분석이 유효한 것은 주로 고전암호의 환자식 암호이며, 9세기에 아라비아인 킨디가 집필한 암호문서의 해독에 관한 수기에 이 해독법의 기술(記述)이 있다. 15세기 즈음에는 르네상스를 통해 유럽에도 퍼져, 비즈네르 암호와 같은 다표식 환자(多表式換字)를 고안하는 동기가 되었고, 20세기에 이를 때까지 새로운 암호방식의 제안과 빈도분석을 바탕으로 하는 해독법의 개량이 되풀이되었다. 20세기 초두에 개발된 기계식 암호에 의해, 단순한 빈도분석은 적용하기 곤란해졌으며, 암호해독은 알고리즘의 수학적 분석을 수반한 연구로 변질해갔다. 현대암호에서는 암호문 단독공격보다도 복잡한 기지평문(旣知平文)이나 선택암호문 조건에서도 안전함을 목표로 설계되어, 이 분야에서는 평문의 언어적인 특징을 단서로 하는 빈도분석은 극복되었다.

빈도분석 방식은 암호를 해독하는 데에만 쓰이는 것이 아니라, 고대문자의 해독에도 이용된다.

정요 [편집]

이렇게 빈도분석 암호와

ASCII

80개 언어

문서 토론 읽기 편집 역사 보기 도구

위키백과, 우리 모두의 백과사전.

미국정보교환표준부호(영어: American Standard Code for Information Interchange), 또는 줄여서 ASCII(/ˈæski/, **아스키**)는 영문 알파벳을 사용하는 대표적인 문자 인코딩이다. 아스키는 컴퓨터와 통신 장비를 비롯한 문자를 사용하는 많은 장치에서 사용되며, 대부분의 문자 인코딩이 아스키에 기초를 두고 있다.

아스키는 7비트 인코딩으로, 33개의 출력 불가능한 제어 문자들과 공백을 비롯한 95개의 출력 가능한 문자들로 총128개로 이루어진다. 제어 문자들은 역사적인 이유로 남아 있으며 대부분은 더 이상 사용되지 않는다. 출력 가능한 문자들은 52개의 영문 알파벳 대소문자와, 10개의 숫자, 32개의 특수 문자, 그리고 하나의 공백 문자로 이루어진다.

아스키가 널리 사용되면서 다양한 아스키 기반의 확장 인코딩들이 등장했으며, 이들을 묶어서 아스키라고 부르기도 한다. 대표적으로 7비트 인코딩을 유지한 ISO/IEC 646과, 원래 아스키 코드 앞에 비트 0을 넣어 8비트 인코딩을 만든 IBM 코드 페이지와 ISO 8859가 있다. 이 인코딩들은 언어군에 따라 같은 숫자에 서로 다른 문자가 배당된 경우가 많다.

ASCII

```
 !"#$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_
`abcdefghijklmnopqrstuvwxyz{|}~
```

ASCII (1967년 및 그 이후)

MIME / IANA	us-ascii
다른 이름	ISO-IR-006[1]
언어	영어
분류	ISO 646 시리즈
확장 버전	• 유니코드 • ISO/IEC 8859 (시리즈) • KOI-8 • OEM (시리즈) • Windows-125x (시리즈) • 기타
이전 인코딩	ITA 2, FIELDATA
다음 인코딩	ISO 8859, 유니코드
그 밖의 관련 인코딩	PETSCII

역사 [편집]

아스키 코드는 지금의 미국 국가표준 협회(ANSI)의 전신인 미국 표준 협회(ASA)가 주도한 X3 위원회가 개발했다. 그 아래의 X3.2 소위원회는 1960년 10월 6일 아스키 표준화 작업을 시작하여, 1963년 표준화 초판을 발간했고,[2][3] 1967년 개정했으며,[4][5] 가장 최근의 업데이트는 1986년에 있었다.[6]

제어 문자표 [편집]

아스키 코드에 대한 위키문서였다. 빈도분석 암호에 대해 알아야 풀 수 있는 문제인가?

코드 내용에서 중간에 16진수로 변환하는 내용이 있는 건 알았다. 그럼 아스키 코드를 이용해서 암호화시킨 건가?

인코딩과 디코딩 변환 사이트를 이용해보기로 했다.

뭔가 이것저것 여러번 시도해보긴 했는데 딱히 더 나오는 건 없었다.