

MISC - ez_dialogue - WriteUp

題目附件如下：

文件名	修改時間	類型	大小
mystery_dialogue	2025/6/1 18:36	文件	56,433 KB
secret.zip	2025/6/1 21:17	ZIP 壓縮文件	10,187 KB
這也許能幫到你.txt	2025/6/1 21:18	TXT 文件	1 KB

其中文檔給了一些簡單的hint，我們可以知道flag星人喜歡十進製的7；壓縮包及內容被處理了；通訊文件數據做了簡單的運算，那麼就能推斷出三個信息，後續肯定有關於7的用處；需要處理壓縮包；我們需要對通訊文件的處理恢復一下

```
secret.dll.dll d3rpg.ini d3rpg.exe d3rpg.dll d3rpg.d3ssad 1.txt flag.txt server.py countle_puzzle.py 這也許能幫到你.txt
1 人類聯軍收集到的一些有用的信息：
2
3
4 截獲到的通訊文件數據都被做了一些簡單的運算
5 在數據庫中找到了一個加密壓縮包，這似乎是我們的臥底臨終前送回來的，為了不被敵軍識破，他特地進行了加密和某些隱寫
6 flag星人特別喜歡數字7
```

用010查看一下通訊文件的具體情況，首先根據題目，不難猜出這原本應該是一個音頻文件，那麼常見的音頻文件有比如wav, mp3，而他們的文件頭分別是RIFF和ID3。分析這個通訊文件，雖然它並不沒有明顯的文件頭，但可以看出來它的格式為ABCC式，這恰恰對應了wav的文件頭RIFF的格式，所以可以推斷出原本的文件是wav文件

而且題目中也說了簡單的運算，而運算我們常見的幾種就是與，或，異或.....

那麼要運算多少位呢，實際上這裏就用到了文檔中的提示，要跟7進行運算

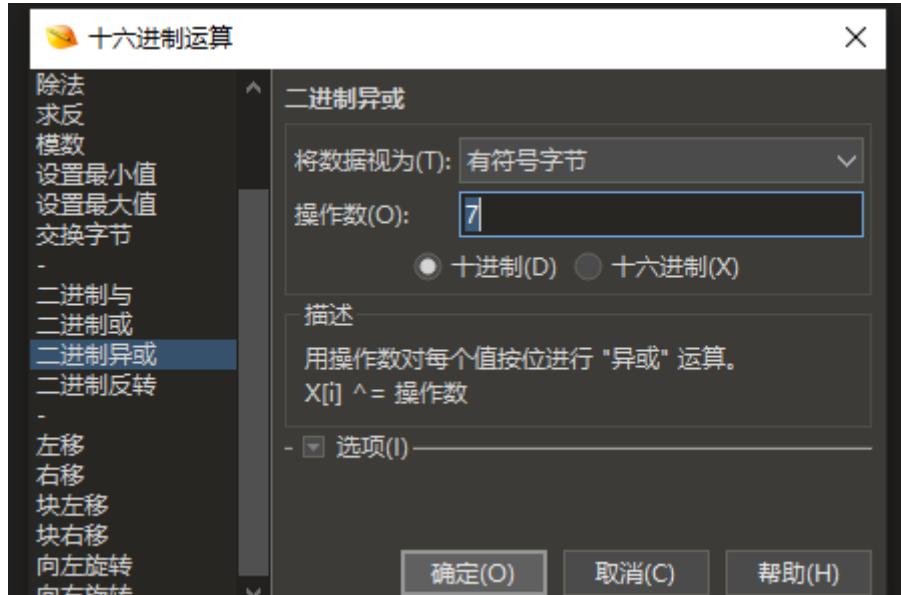
010 Editor - D:\网安实验室\出题\题目\ez_audio\附件\mystery_dialogue

文件(F) 编辑(E) 搜索(S) 规则(V) 格式(O) 脚本(I) 模板(L) 调试(D) 项目(P) 工具(T) 窗口(W) 帮助(H)

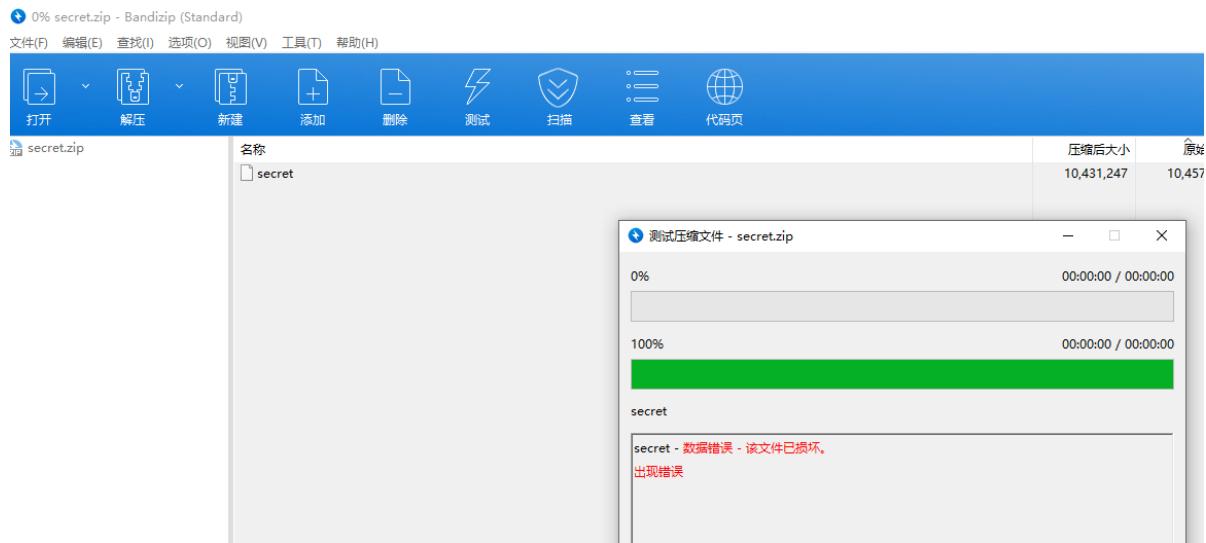
起始页 mystery_dialogue x

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
000:0000	55	4E	41	41	B3	C6	76	04	50	46	51	42	61	6A	73	27
000:0010	17	07	07	07	06	07	07	06	43	AB	07	07	8F	5F	06	07
000:0020	05	07	17	07	63	66	73	66	97	C6	76	04	07	07	AE	08
000:0030	A6	E6	3B	2C	8E	CD	74	3B	38	C7	29	38	3B	C2	C1	35
000:0040	49	D9	32	1D	E0	F3	5B	FC	18	13	A7	DA	8C	29	27	CF
000:0050	D7	3A	04	C7	41	39	3D	CD	D6	28	OB	DB	E0	12	7C	FE
000:0060	B9	F1	7E	1F	CA	DE	9E	36	05	C1	DD	39	1B	C7	0D	3A
000:0070	82	CE	9C	2B	FC	D8	7B	16	1A	F9	2A	F5	B5	1B	2A	D1
000:0080	6F	33	C3	88	38	B8	7B	3C	1E	CB	2E	2E	11	E3	c3A4	8
000:0090	FE	0B	C6	05	A9	EA	C2	27	C1	D5	E9	31	B3	C5	E8	38
000:00A0	56	C6	99	3E	F4	97	82	21	51	EF	62	0F	65	00	4F	EE
000:00B0	AC	23	9B	C8	2C	3E	8F	C6	FF	38	6C	C5	75	30	08	D5
000:00C0	A4	26	B2	EB	C2	04	FD	0C	06	E2	67	2F	B4	CB	1D	3C
000:00D0	B7	C7	A8	38	DF	C4	FB	33	6D	2D	9D	1A	28	F6	25	F8
000:00E0	86	17	D9	E7	D8	2C	17	CD	BD	3B	29	C7	0E	38	91	C2
000:00F0	39	35	FA	DF	6E	1E	BA	F2	79	FD	F5	13	E2	DB	25	28
000:0100	B2	C0	0F	39	06	C7	16	39	A4	C0	3B	28	C2	DB	12	12
000:0110	5E	FD	E5	F2	41	1E	1C	DE	21	35	A2	C2	05	38	2C	C7
000:0120	C1	3B	FC	CE	FD	2C	BA	E7	A2	17	FB	F9	53	F6	7E	1A
000:0130	81	D2	E1	33	E2	C4	AF	38	AC	C7	2E	3C	9A	CB	7A	2F
000:0140	D8	E3	18	OB	A7	04	DE	EB	B4	26	2D	D5	58	30	71	C5
000:0150	F0	38	87	C6	3B	3E	83	C8	CD	23	22	EE	8F	00	47	0F
000:0160	7E	EF	66	22	OB	C8	89	03	SE	C6	F6	38	AE	C5	05	30
000:0170	AB	D5	E2	27	8D	EA	E0	05	D3	08	3F	E3	0B	2E	2B	CB
000:0180	69	36	C7	93	38	22	C3	7A	33	16	D1	D3	1B	0F	F5	
000:0190	44	9F	51	16	1B	E7	87	2B	9E	CE	F8	3B	18	C7	E6	39
000:01A0	F5	C2	B5	36	A8	DE	9B	1F	9F	F1	A6	FE	C4	12	2C	DB
000:01B0	BF	28	4B	CO	3A	39	05	C5	DD	3A	0A	CF	A3	29	87	DA
000:01C0	45	13	30	FC	00	F2	2C	15	1D	6C	DF	A8	35	4C	C2	3F
000:01D0	3B	C7	78	3B	73	5C	50	2D	87	E6	C9	08	DC	F8	7C	F7
000:01E0	39	19	E7	D3	64	32	9D	C4	BA	38	CB	C7	D4	3D	24	CA
000:01F0	D7	20	AE	E2	43	DC	79	03	02	EB	47	25	88	D6	C9	30
000:0200	3D	C5	FA	3B	8G	C6	DD	3F	11	D7	14	23	F1	EE	AD	01
000:0210	1B	0E	AD	E0	12	21	7A	C9	E9	3E	2A	C6	1E	38	ED	C5
000:0220	88	31	D4	22	27	67	E9	0F	05	A9	0A	77	E4	B1	2E	"1MO"
000:0230	A9	CC	C7	3C	72	C7	7E	38	71	C3	F4	34	BD	D1	0B	1B
000:0240	E5	F5	63	FA	29	15	5B	D8	18	2A	22	CE	46	3A	14	C7
000:0250	B1	39	56	C1	22	36	65	DD	CA	10	71	FO	F4	FF	93	11
000:0260	73	D3	4C	37	E1	71	39	00	C7	9A	3C	CF	02	29	sUL7Aw9.	C7:I.)
000:0270	3A	DC	68	14	12	FB	2C	F3	DA	10	BA	0D	31	34	F3	C3
000:0280	4D	3E	56	C7	32	3B	EB	CD	B5	2D	42	E5	F2	09	BE	07
000:0290	55	C7	D4	77	38	F6	C7	7E	3D	6E	002U2Ax80C=	=	=	
000:02A0	AC	CA	26	20	72	E1	6F	0D	5B	02	35	EC	FB	25	F0	D7
000:02B0	3D	3F	06	C5	F8	38	E2	C6	68	3F	AC	D7	5D	24	CO	ED
000:02C0	CA	02	FF	0E	DE	E1	C0	21	F6	CA	4C	3D	03	C6	DF	38
000:02D0	59	2B	28	CC	09	3B	5B	C7	5C	3B	CE	C3	7D	34	62	DD
000:02E0	Y-1.	1C	91	37	10	DC	F9	11	54	FF	E1	F0
000:02F0	44	1C	BB	F4	81	FB	04	14	9A	D9	BB	2A	B4	CF	87	3A
000:0300	0D	C7	8F	39	B4	C1	92	37	10	DC	F9	11	54	FF	E1	F0
000:0310	63	10	B9	DD	3B	37	84	C1	98	39	09	C7	66	3A	EC	CF
000:0320	68	2A	FC	99	15	F4	FB	56	F4	A1	1C	16	D0	BC	34	h^U.
000:0330	A7	C3	6D	38	6F	C7	EF	3C	6A	CC	0B	2D	OC	E4	1B	09
000:0340	90	06	CA	E9	C4	18	9C	D4	54	31	OB	C4	D8	38	1F	C6

結合上面的信息，簡單的嘗試後就能得到是要對文件進行異或操作，而操作數則是7



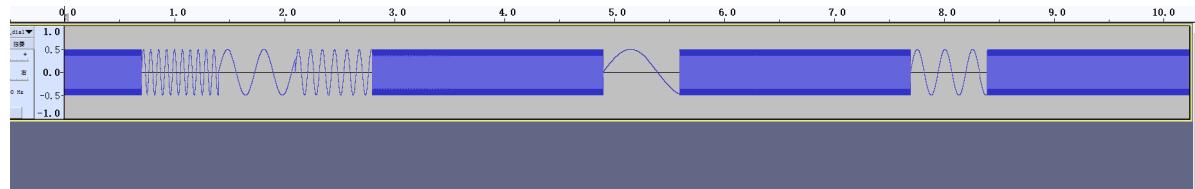
而打開壓縮包，嘗試解壓會發現數據錯誤，實際上這是將加密壓縮包偽造成了無加密的樣子，所以這裏需要用010修改一下他的加密位（分別修改frflags和deflags）



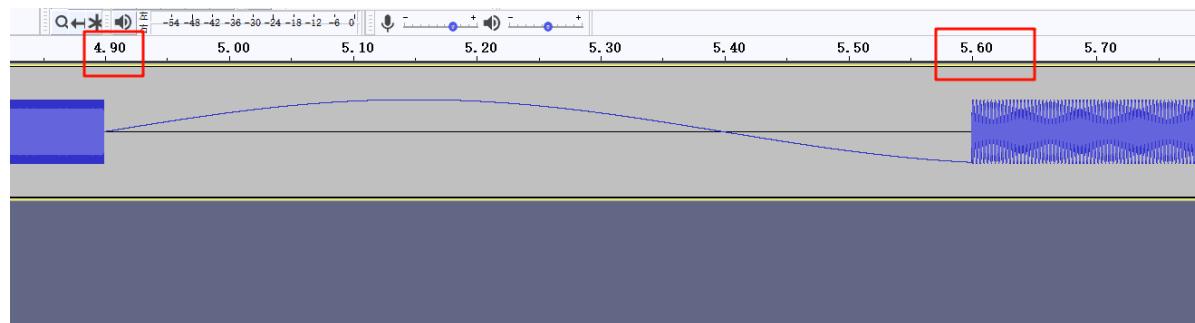
修改後需要密碼，直接爆破即可，密碼為1264

解壓後得到一個文件，用010打開簡單分析可以發現是一個倒置的pdf文件，用工具或腳本簡單處理一下即可

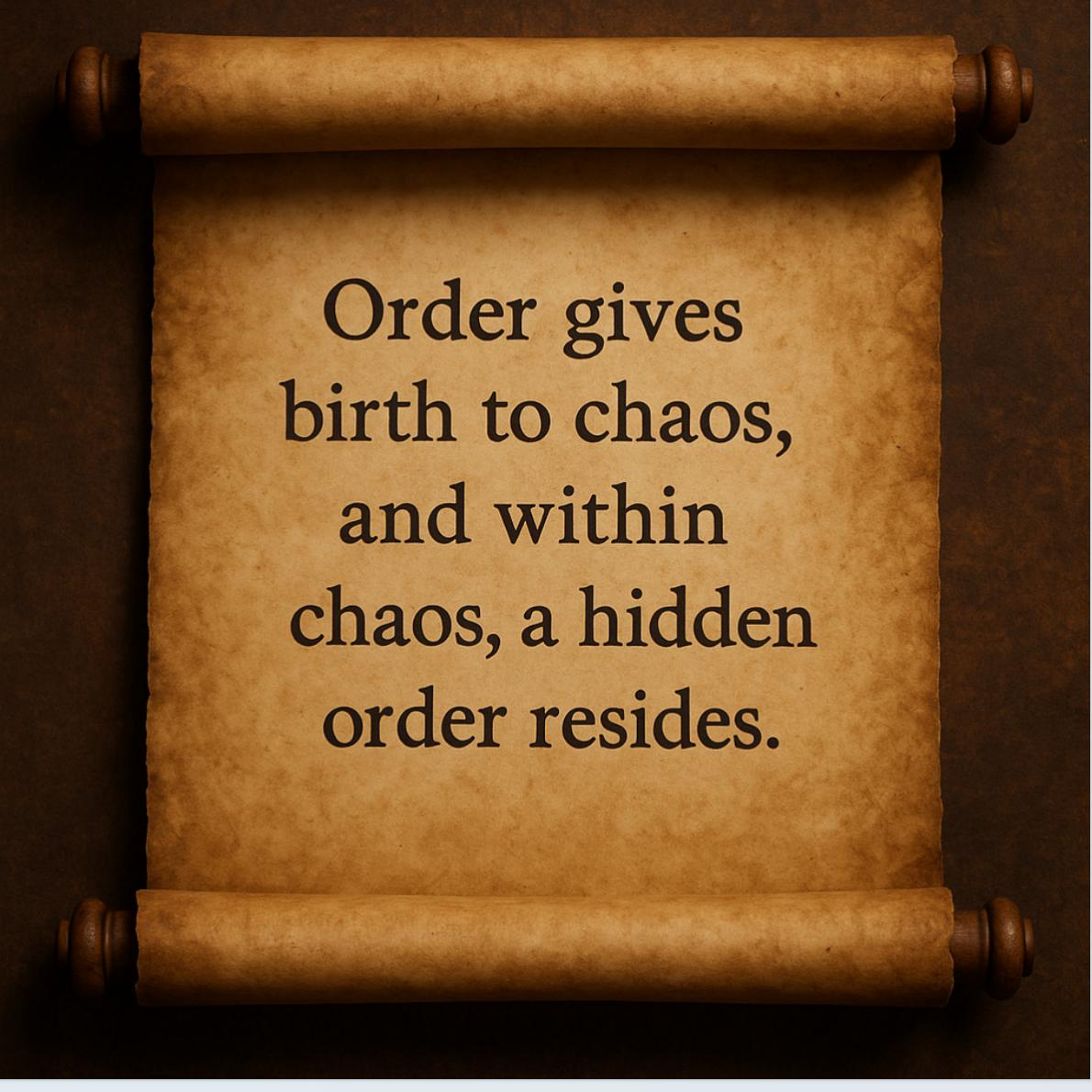
這邊先對音頻進行分析，用audacity打開，稍微放大可以看到裏面有各種各樣奇怪的波形



對其中的波形放大後，可以發現同一種波形的持續時間是0.7s（與提示第一條相對應），除此之外似乎並沒有明顯的規律

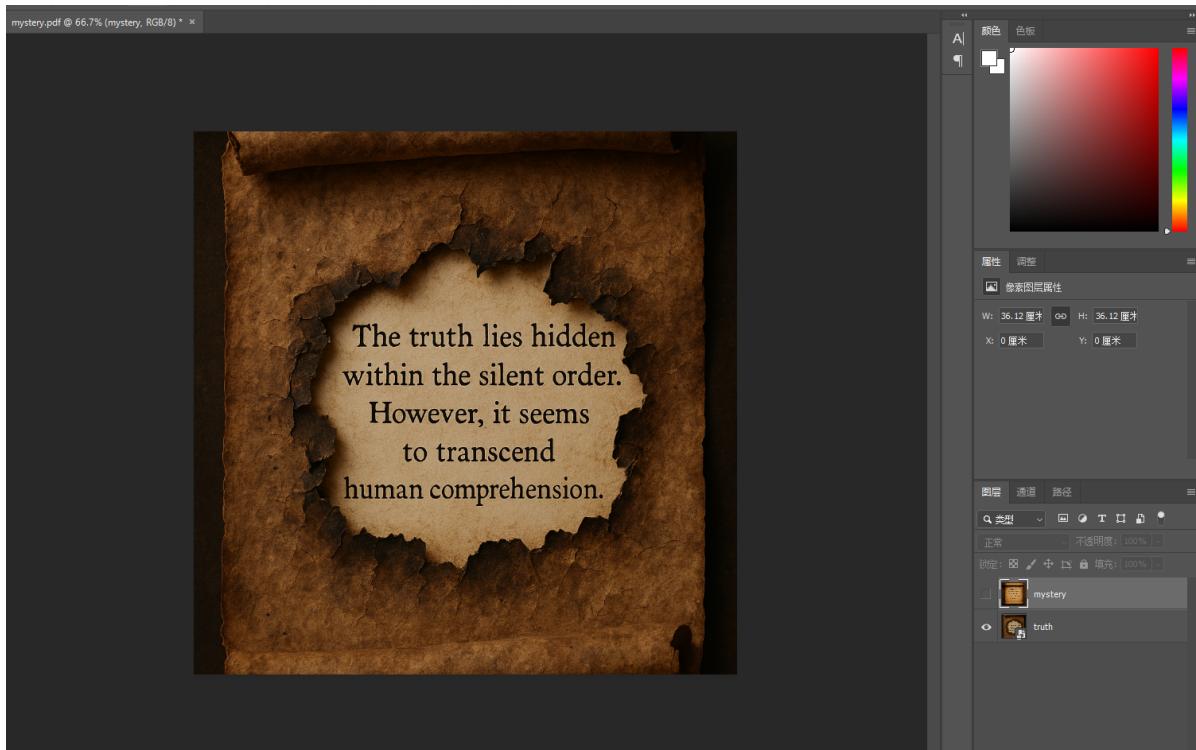


再來看看提出來的pdf，裏面是一張圖片，圖片中給了提示"Order gives birth to chaos, and within chaos, a hidden order resides"，這似乎告訴我們在混亂的波形中藏了穩定的信息，但可能還是無從下手（當然了如果您僅靠這層提示就解出來了，是很厲害）



Order gives
birth to chaos,
and within
chaos, a hidden
order resides.

事實上這裏是pdf多圖層隱寫，將pdf丟進ps可以發現下面還藏了一個圖層，這個圖層告訴我們"The truth lies hidden within the silent order. However, it seems to transcend human comprehension"，這裏有兩個關鍵提示，一個是真理潛藏在無聲的秩序中，也就是需要留意沒有聲音的部分，而第二個提示則是暗示我們需要關注人類無法理解的部分。這裏其實是考察了一個熱知識：人耳能夠聽到的頻率在20hz~20000hz，在這個範圍外的聲音，人耳是無法聽到的，這也就形成了即便有波形但是我們聽起來卻是靜音的情況，對應了人類無法理解



至此得到的信息其實已經比較充足了，顯然我們需要去分析不在20hz~20000hz這個範圍內的波形，問題在於怎麼分析呢？

這裏可以用腳本將音頻按照每0.7s進行切割，然後檢測每一段波形的頻率，將所有不在可聽範圍的波形頻率按順序輸出，去分析他的規律。其實不難發現會出現兩種類型的頻率，一種是低於20hz的頻率，一種則是高於20000hz的頻率，而兩種頻率，正對應著01，也就是我們最熟悉的二進製

這裏給出解析的腳本

```
import wave
import numpy as np
from scipy.fft import fft, fftfreq

filename = 'mystery_sound.wav'
sum = ''
with wave.open(filename, 'rb') as wf:
    sample_rate = wf.getframerate()
    num_frames = wf.getnframes()
    audio_data = np.frombuffer(wf.readframes(num_frames), dtype=np.int16)

duration_per_segment = 0.7 # 每段0.7秒
samples_per_segment = int(sample_rate * duration_per_segment)

# 分段分析
num_segments = len(audio_data) // samples_per_segment

for i in range(num_segments):
    segment = audio_data[i * samples_per_segment : (i + 1) * samples_per_segment]

    fft_result = np.abs(fft(segment))
    freqs = fftfreq(len(segment), d=1/sample_rate)

    positive_freqs = freqs[:len(freqs)//2]
    positive_magnitudes = fft_result[:len(fft_result)//2]
```

```
peak_freq = positive_freqs[np.argmax(positive_magnitudes)]  
  
if peak_freq < 20:  
    sum = sum+'0'  
    print(f'第{i + 1}段：主频率 = {peak_freq:.2f} Hz → bit = 0')  
elif peak_freq > 20000:  
    sum = sum+'1'  
    print(f'第{i + 1}段：主频率 = {peak_freq:.2f} Hz → bit = 1')  
else:  
    print(f'第{i + 1}段：主频率 = {peak_freq:.2f} Hz → bit = ?')  
  
print(sum)  
print(len(sum))
```

成功將音頻轉為01字符串後，用廚子轉換成ascii值即可，不過不要忘了flag星人喜歡7哦（七位長度的二進製字符串）

然後就可以解出flag星人的對話，從中獲得神之flag

