



Write_Up

[K.knock _ 8]
정덕호

{ NetWork _ Dialogue }

Challenge 6 Solves X

Dialogue

475

2017학년도 대학수학능력시험 문제지 1

제 3 교시 영어 영역 흘수형

1. 대화를 듣고, 여자의 마지막 말에 대한 남자의 응답으로 가장 적절한 것을 고르시오.

① Yes. That would be great.
② Sure. We had a great time.
③ Right. I already got a job.
④ Never. They haven't seen it.
⑤ No. There's no writing class.

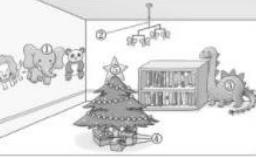
2. 대화를 듣고, 남자의 마지막 말에 대한 여자의 응답으로 가장 적절한 것을 고르시오.

① I agree. But I don't have the time for it.
② You're right. Then I'll never tell anyone.
③ Trust me. You'll realize you did the right thing.
④ I understand. But let us know if it happens again.
⑤ That's great. We've been practicing for a long time.

3. 다음을 듣고, 여자가 하는 말의 목적으로 가장 적절한 것을 고르시오.

① 놀酱 체험 프로그램을 홍보하려고
② 우산소 춘동의 장점을 소개하려고
③ 가족 비행 글이들을 소개하려고
④ 유제품 보관 방법을 설명하려고
⑤ 저지방 식단의 중요성을 강조하려고

6. 대화를 듣고, 그림에서 대화의 내용과 일치하지 않는 것을 고르시오.



7. 대화를 듣고, 여자가 할 일로 가장 적절한 것을 고르시오.

① 공연 연습 도와주기
② 캠핑 캐리어 만들기
③ 커버라 가방 구매하기
④ 여분의 배트리 카드 찾기
⑤ 아이의 무대의상 가져오기

8. 대화를 듣고, 여자가 하이킹을 할 수 있는 이유를 고르시오.

① 후대진화를 수리해야 해서
② 할머니를 찾아뵈어야 해서
③ 의사 회의에 참석해야 해서
④ 신상을 광고를 준비해야 해서
⑤ 친구들과 영화를 보러 가야 해서

9. 대화를 듣고, 남자가 지불할 금액을 고르시오.

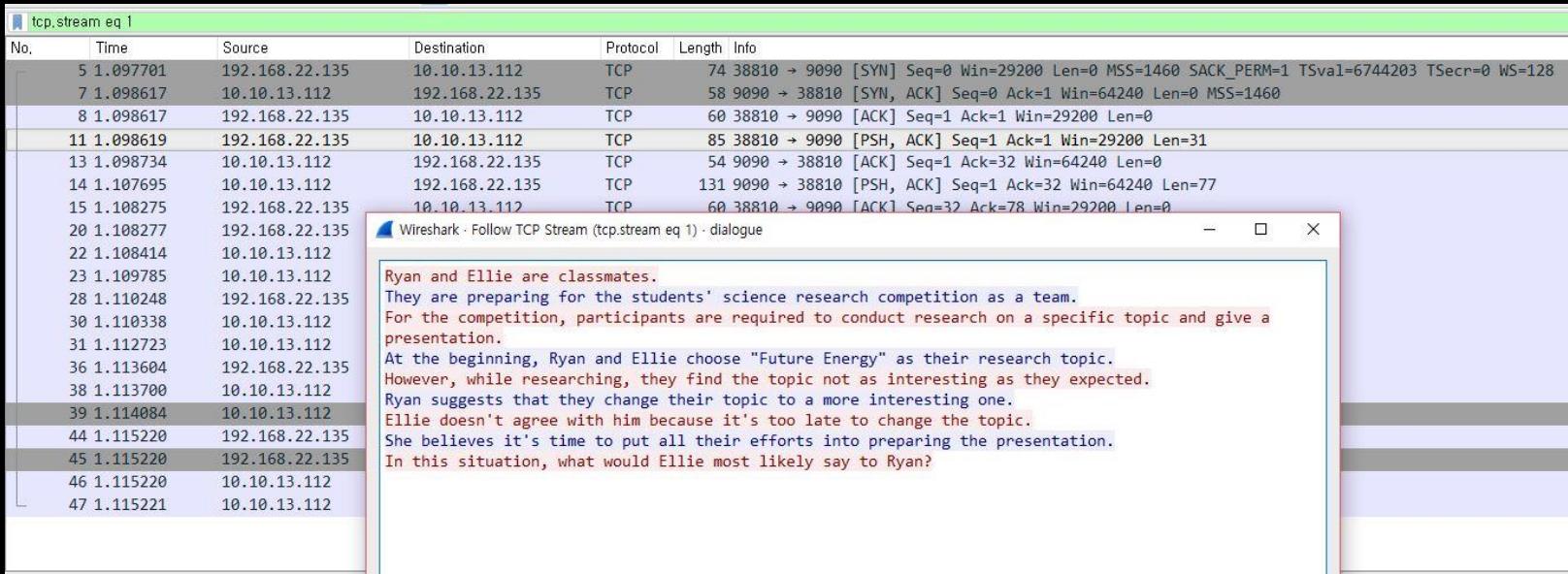
① \$36 ② \$40 ③ \$45 ④ \$47 ⑤ \$50

Ellie는 Ryan에게 뭐라고 말했을까?

dialogue.pcap...

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.22.1	192.168.22.255	UDP	305	54915 → 54915 Len=263
2	1.001641	192.168.22.1	192.168.22.255	UDP	305	54915 → 54915 Len=263
3	1.095992	192.168.22.1	192.168.22.135	SSH	90	Client: Encrypted packet (len=36)
4	1.096861	192.168.22.135	192.168.22.1	TCP	60	22 → 53732 [ACK] Seq=1 Ack=37 Win=251 Len=0
5	1.097701	192.168.22.135	10.10.13.112	TCP	74	38810 → 9090 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=6744203 TSecr=0 WS=128
6	1.097702	192.168.22.135	192.168.22.1	SSH	90	Server: Encrypted packet (len=36)
7	1.098617	10.10.13.112	192.168.22.135	TCP	58	9090 → 38810 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
8	1.098617	192.168.22.135	10.10.13.112	TCP	60	38810 → 9090 [ACK] Seq=1 Ack=1 Win=29200 Len=0
9	1.098618	192.168.22.135	192.168.22.1	SSH	154	Server: Encrypted packet (len=100)
10	1.098618	192.168.22.135	192.168.22.1	SSH	90	Server: Encrypted packet (len=36)
11	1.098619	192.168.22.135	10.10.13.112	TCP	85	38810 → 9090 [PSH, ACK] Seq=1 Ack=1 Win=29200 Len=31
12	1.098711	192.168.22.1	192.168.22.135	TCP	54	53732 → 22 [ACK] Seq=37 Ack=173 Win=2050 Len=0
13	1.098734	10.10.13.112	192.168.22.135	TCP	54	9090 → 38810 [ACK] Seq=1 Ack=32 Win=64240 Len=0
14	1.107695	10.10.13.112	192.168.22.135	TCP	131	9090 → 38810 [PSH, ACK] Seq=1 Ack=32 Win=64240 Len=77
15	1.108275	192.168.22.135	10.10.13.112	TCP	60	38810 → 9090 [ACK] Seq=32 Ack=78 Win=29200 Len=0
16	1.108276	192.168.22.135	192.168.22.1	SSH	186	Server: Encrypted packet (len=132)
17	1.108276	192.168.22.135	192.168.22.1	SSH	90	Server: Encrypted packet (len=36)
18	1.108277	192.168.22.135	192.168.22.1	SSH	234	Server: Encrypted packet (len=180)
19	1.108277	192.168.22.135	192.168.22.1	SSH	90	Server: Encrypted packet (len=36)
20	1.108277	192.168.22.135	10.10.13.112	TCP	166	38810 → 9090 [PSH, ACK] Seq=32 Ack=78 Win=29200 Len=112
21	1.108388	192.168.22.1	192.168.22.135	TCP	54	53732 → 22 [ACK] Seq=37 Ack=557 Win=2049 Len=0
22	1.108414	10.10.13.112	192.168.22.135	TCP	54	9090 → 38810 [ACK] Seq=78 Ack=144 Win=64240 Len=0
23	1.108705	10.10.13.112	192.168.22.135	TCP	425	38810 → 9090 [PSH, ACK] Seq=78 Ack=144 Win=64240 Len=0

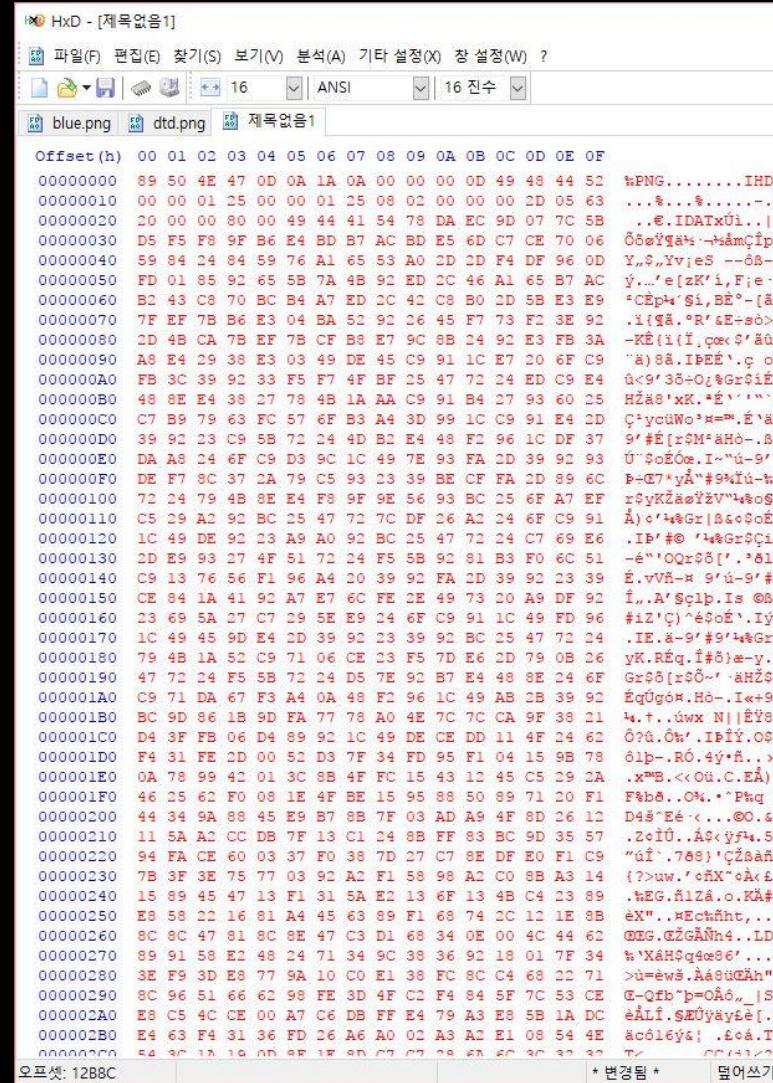
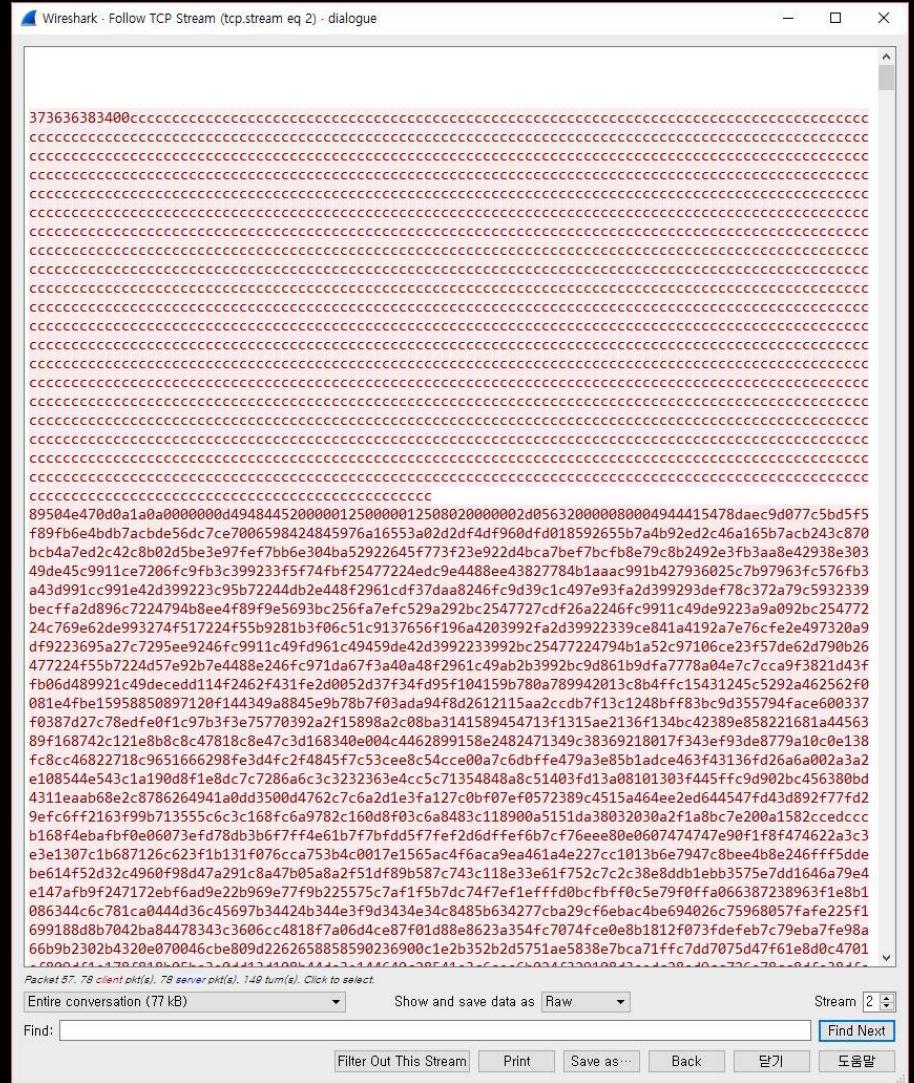
파일을 와이어 샤크로 열어보니
 일단 TCP 3-way Handshake를 통해 연결이 이루어 지는 과정이 보인다.
 그후 보이는 PSH Flag를 보니 데이터가 전달되고 있는 걸 알 수 있다.



Psh 플래그 비트가 보이는 위치에서
와이어샤크의 기능중 하나인 Follow TCP Stream 을 이용해 확인해보니
영어 문장을 주고 받고 있는 것이 보인다.

No.	Time	Source	Destination	Protocol	Length	Info
43	1.115219	192.168.22.135	192.168.22.1	SSH	90	Server: Encrypted packet (len=36)
44	1.115220	192.168.22.135	10.10.13.112	TCP	115	38810 → 9090 [PSH, ACK] Seq=301 Ack=311 Win=29200 Len=61
45	1.115220	192.168.22.135	10.10.13.112	TCP	60	38810 → 9090 [FIN, ACK] Seq=362 Ack=311 Win=29200 Len=0
46	1.115220	10.10.13.112	192.168.22.135	TCP	54	9090 → 38810 [ACK] Seq=311 Ack=362 Win=64240 Len=0
47	1.115221	10.10.13.112	192.168.22.135	TCP	54	9090 → 38810 [ACK] Seq=311 Ack=363 Win=64239 Len=0
48	1.115221	192.168.22.135	192.168.22.1	SSH	154	Server: Encrypted packet (len=100)
49	1.115311	192.168.22.1	192.168.22.135	TCP	54	53732 → 22 [ACK] Seq=37 Ack=1649 Win=2051 Len=0
50	1.997484	192.168.22.1	192.168.22.255	UDP	305	54915 → 54915 Len=263
51	2.992343	fe80::f086:d8d0:a5e.. ff02::1:3	LLMNR	86	Standard query 0x695b A isatap	
52	2.992503	192.168.22.1	224.0.0.252	LLMNR	66	Standard query 0x695b A isatap
53	3.001763	192.168.22.1	192.168.22.255	UDP	305	54915 → 54915 Len=263
54	3.018433	Vmware_c0:00:08	Broadcast	ARP	42	Who has 192.168.22.2? Tell 192.168.22.1
55	3.099448	192.168.22.1	192.168.22.255	UDP	305	54915 → 54915 Len=263
56	4.081263	Vmware_c0:00:08	Broadcast	ARP	42	Who has 192.168.22.2? Tell 192.168.22.1
57	4.683786	192.168.22.1	192.168.22.135	TCP	66	55508 → 9000 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
58	4.683973	192.168.22.135	192.168.22.1	TCP	66	9000 → 55508 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
59	4.684046	192.168.22.1	192.168.22.135	TCP	54	55508 → 9000 [ACK] Seq=1 Ack=1 Win=525568 Len=0
60	4.684604	192.168.22.1	192.168.22.135	TCP	1078	55508 → 9000 [PSH, ACK] Seq=1 Ack=1 Win=525568 Len=1024
61	4.684780	192.168.22.1	192.168.22.135	TCP	1514	55508 → 9000 [PSH, ACK] Seq=1025 Ack=1 Win=525568 Len=1460
62	4.684789	192.168.22.1	192.168.22.135	TCP	642	55508 → 9000 [PSH, ACK] Seq=2485 Ack=1 Win=525568 Len=588
63	4.684845	192.168.22.135	192.168.22.1	TCP	60	9000 → 55508 [ACK] Seq=1 Ack=1025 Win=31360 Len=0
64	4.685083	192.168.22.135	192.168.22.1	TCP	60	9000 → 55508 [ACK] Seq=1 Ack=2485 Win=34176 Len=0
65	4.685083	192.168.22.135	192.168.22.1	TCP	60	9000 → 55508 [ACK] Seq=1 Ack=2485 Win=34176 Len=0
▶ Ethernet II, Src: Vmware_c0:00:08 (00:50:56:c0:00:08), Dst: Vmware_6b:da:e5 (00:0c:29:6b:da:e5)						
▶ Internet Protocol Version 4, Src: 192.168.22.1, Dst: 192.168.22.135						
▶ Transmission Control Protocol, Src Port: 55508, Dst Port: 9000, Seq: 1025, Ack: 1, Len: 1460						
▼ Data (1460 bytes)						
Data: 89504e470d0a1a0a000000d494844520000012500000125...						
0030	08 05 99 ad 00 00 89 50	4e 47 0d 0a 1a 0a 00 00PNG.....			
0040	00 0d 49 48 52 00 00	01 25 00 00 01 25 08 02	..IHDR.. %.%			
0050	00 00 00 2d 05 63 20 00	00 80 00 49 44 41 54 78	...c ..IDATx			
0060	da ec 9d 07 7c 5b d5 f5	f8 9f b6 e4 bd b7 ac bd			
0070	e5 6d c7 ce 70 06 59 84	24 84 59 76 a1 65 53 a0	.m..p.Y \$.Yv.e\$.			
0080	2d 2d f4 fd 96 0d fd 01	85 92 65 5b 7a 4b 92 ede[zK..			
0090	2c 46 a1 65 b7 ac b2 43	c8 70 bc b4 a7 ed 2c 42	,F.e...C.p....,B			
00a0	c8 b6 2d 5b e3 7f fe	7b b6 e3 04 b4 52 92 26	[....{....R.&			
00b0	45 f7 73 f2 3e 92 2d 4b	ca 7b fe bf cb e7 9c	E.s,>,-K .{....			
00c0	8b 24 92 e3 fb 3a a8 e4	29 38 e3 03 49 de 45 c9	.\$....).8..I.E.			
00d0	91 1c e7 20 6f c9 fb 3c	39 92 33 f5 f7 4f bf 25	... o..< 9.3..0.%			
00e0	47 72 24 ed c9 e4 48 8e	e4 38 27 78 4b 1a aa c9	Gr\$...H.. .8'xk...			
00f0	91 b4 27 93 60 25 c7 b9	79 63 fc 57 6f b3 a4 3d	...'.%.. yc.Wo.=			
0100	99 1c c9 91 e4 2d 39 92	23 c9 5b 72 24 4d b2 e49. #.[#\$M..			
0110	48 f2 96 1c df 37 da a8	24 6f c9 d3 9c 1c 49 7e	H....7.. \$o....I~			
0120	62 6 8 30 20 20 1 57	6 37 8 70 5 82 92 30			

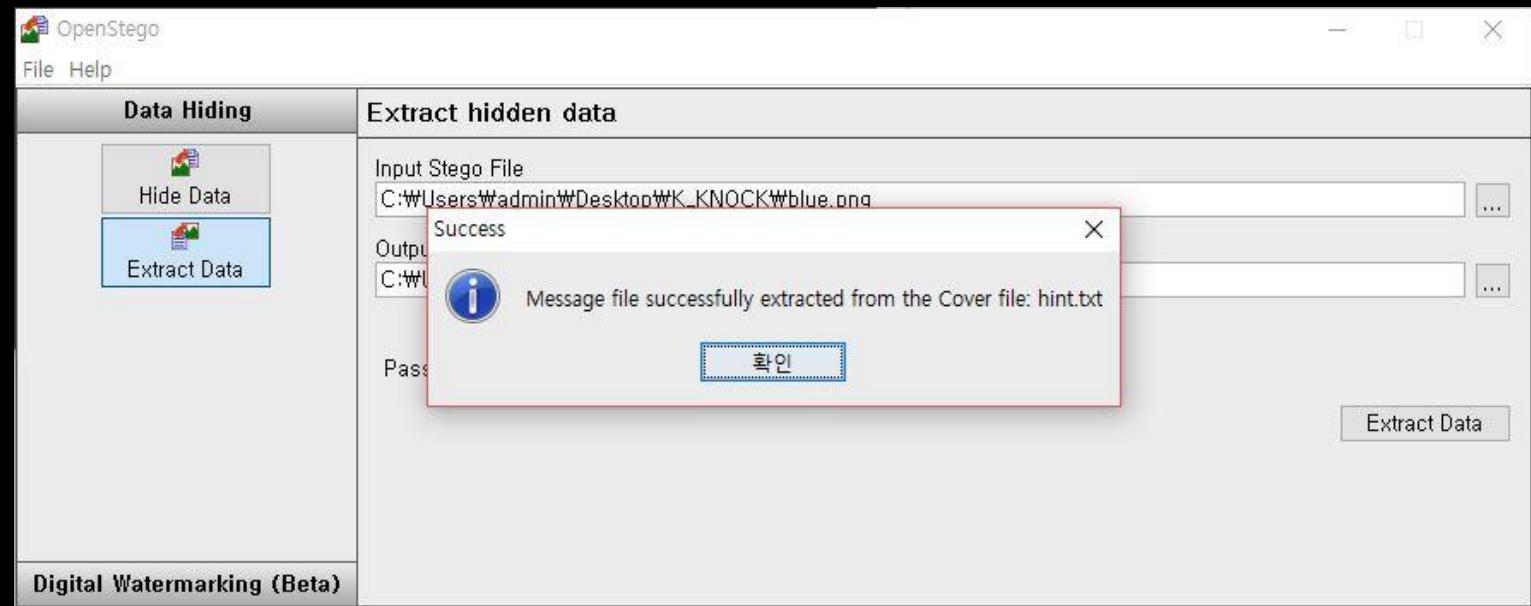
영어 문장 주고 받기가 끝난후
 FIN이후에 다시 SYN 플래그가 보인다.
 아까 (10.10.13.112) 와는 다른 (192.168.22.1) 연결이다.
 그래서 주고 받는 Data를 확인해보니
 총 바이트 크기로 예상되는 76684가 보이고
 그 다음 PNG라는 시그니처가 확인된다.



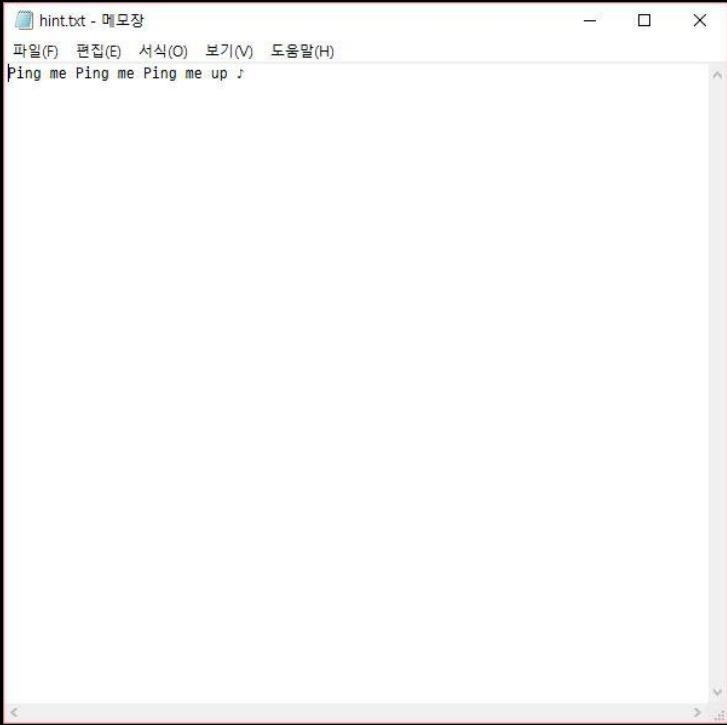
Follow TCP Stream 을 이용한후, 출력 형식을 Raw 형식으로 바꿔서
Hex 에디터에 불인후 PNG 시그니처 앞부분을 제거 하여 사진 파일을 얻었다.



귀..엽당
움.. 멀까 하다가
스테가노 그래피 기법이 의심되어
툴중 하나인 OpenStego를 이용해보았다.



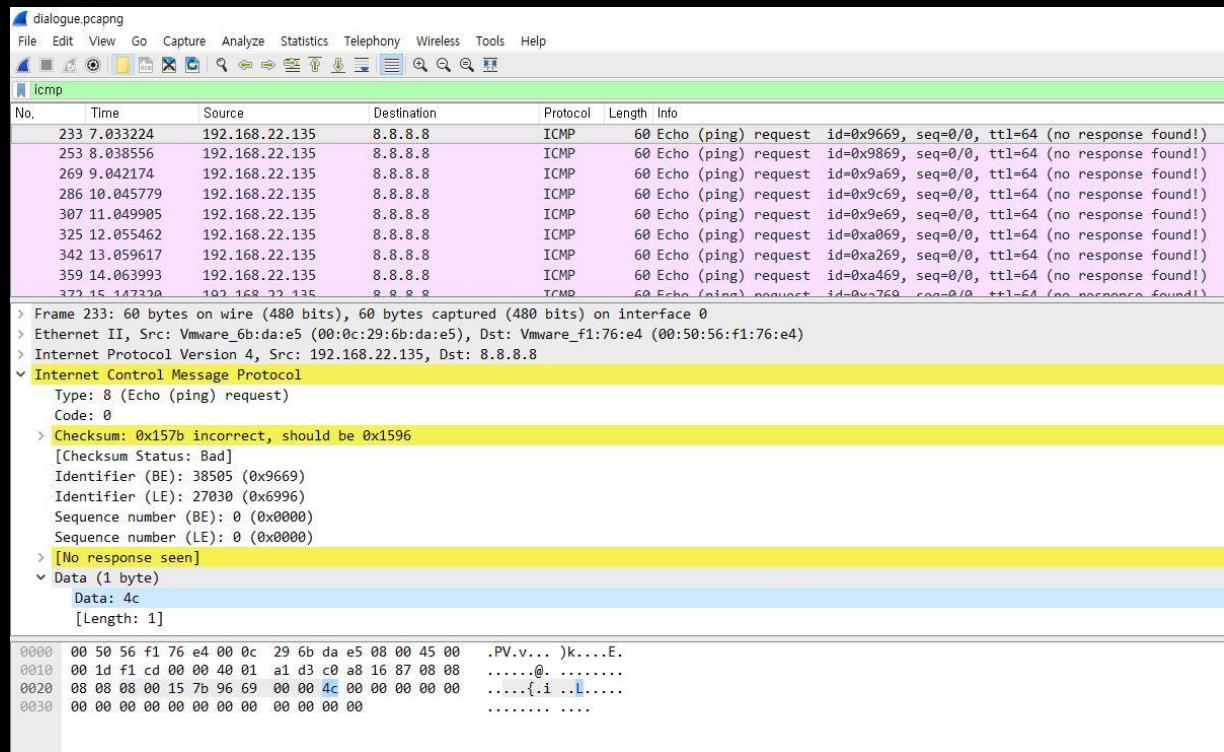
스테가노 그래피는 사진같은 곳에 데이터를 숨기는 것이므로
특수한 툴을 이용하여 데이터를 추출 시도해보았다.
결과는 성공적,
Hint.txt 라는 파일을 얻었다.



핑미 펑미 펑미 업

Ping을 열심히 강조하고 있다.

와이어샤크를 다시 켜서
Ping과 관련된 프로토콜인 ICMP를 검색해보니



1byte에 데이터씩을 전송중인 팽 패킷이 여러 개 보인다.

4C는 아스키코드에서 L에 해당하는데
팽으로 저런 데이터를 보낼 이유는 없으니

매우 의심되어 1byte씩 다 모은후 ascii로 변환해보았다.



그렇다
발표에 집중하젠다

{ Digital Forensic }

> RDP <

Challenge 4 Solves X

RDP

491

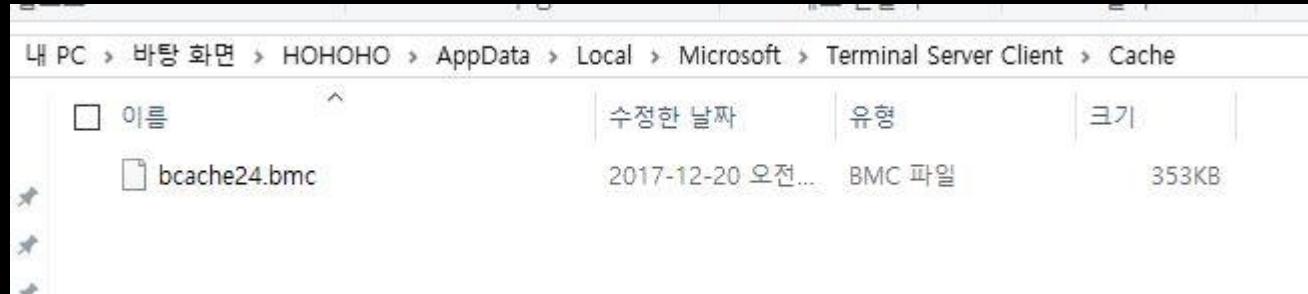
황아저씨는 은아저씨가 자신의 컴퓨터로 다른 PC에 원격을 걸어 작업하는 것을 보았다.
은아저씨는 작업이 끝나고 연결을 종료하자 음흉한 황아저씨는 원격 대상 PC로 연결했던 화면을 훔쳐보려한다.

황아저씨가 되어서 원격 대상 PC의 바탕화면을 찾아보자
Ps. 첨부파일 : 황아저씨의 윈도우 사용자(HOHOHO) 폴더
Flag는 원격 대상 PC의 바탕화면에 있음

HOHOHO.zip

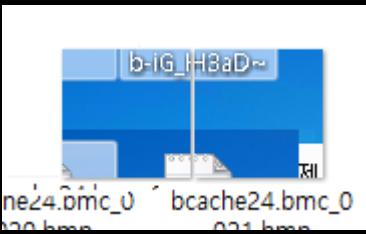
Key

SUBMIT



RDP는 해당 원격 연결을 한
컴퓨터의 정보를
bmc라는 파일에 저장해놓는다.

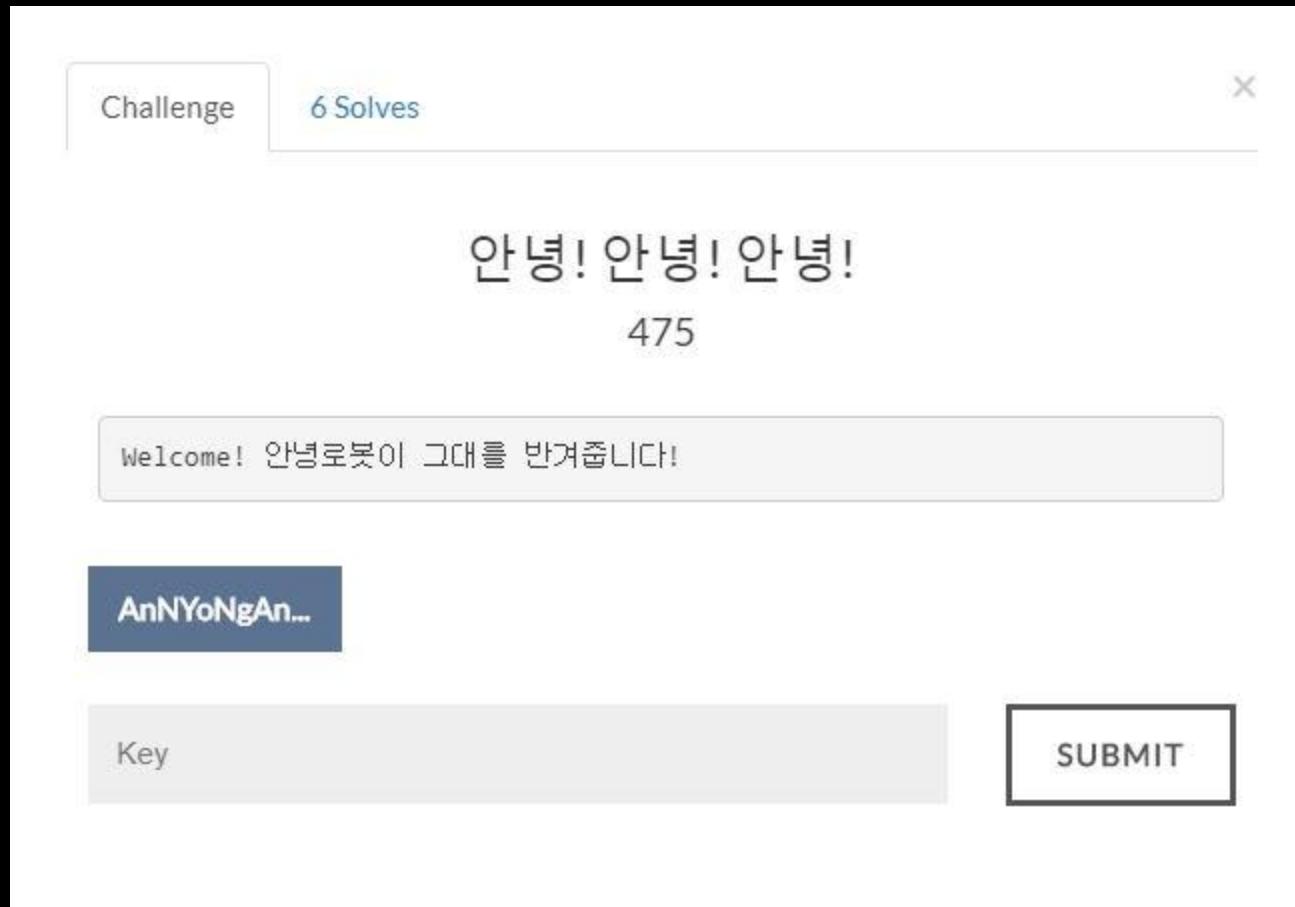
```
C:\Python27>python bmc-tools.py -s bcache24.bmc -d ./  
[+++] Processing a single file: 'bcache24.bmc'.  
[==] Successfully exported 22 files.
```

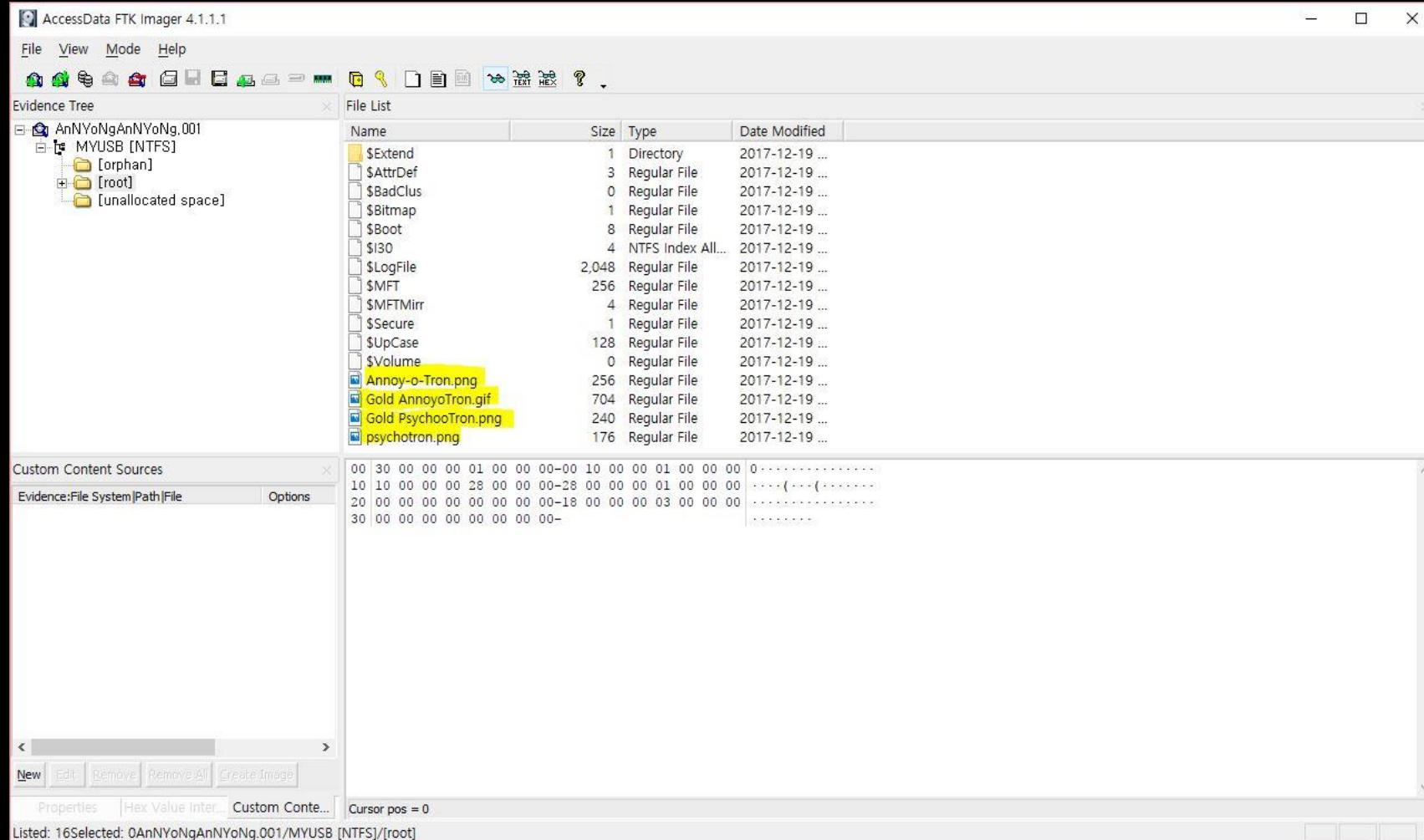


Bmc 파일을 깨어서
비정화된 것을 확인해보니
Flag가 보인다.

{ Digital Forensic }

> 안녕!안녕!안녕!





해당파일을 FTK Imager라는 툴로 열어보니
사진파일 4개가 보인다.

ナフト
吸

제대로 한다 험..

파일(F) 편집(E) 찾기(S) 보기(V) 분석(A) 기타 설정(X) 창 설정(W) ?

16 ANSI 16 진수

Annoy-o-Tron.png

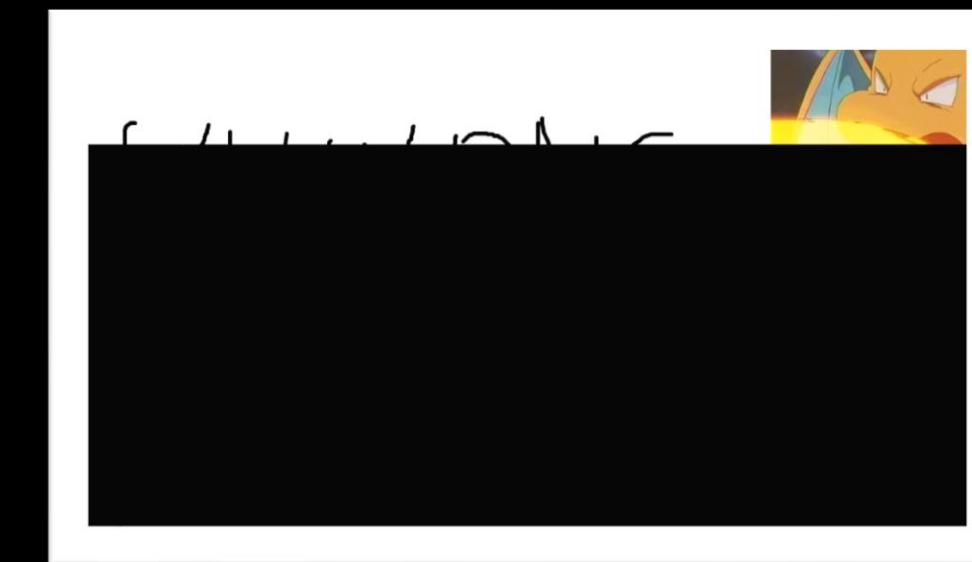
Offset(h)	00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
0002FEA0	48 49 48 49 48 49 48 49 48 49 48 49 48 49 48 49
0002FEB0	48 49 48 49 48 49 48 49 48 49 48 49 48 49 48 49
0002FEC0	48 49 48 49 48 49 48 49 48 49 48 49 48 49 48 49
0002FED0	48 49 48 49 48 49 48 49 48 49 48 49 48 49 48 49
0002FEE0	48 49 48 49 48 49 48 49 48 49 48 49 48 49 48 49
0002FEE0	48 49 48 49 48 49 48 49 48 49 48 49 48 49 48 49
0002FF00	48 49 48 49 48 49 48 49 48 49 48 49 48 49 48 49
0002FF10	48 49 48 49 48 49 48 49 48 49 48 49 48 49 48 49
0002FF20	48 49 48 49 48 49 48 49 48 49 48 49 48 49 48 49
0002FF30	48 49 48 49 48 49 48 49 48 49 48 49 48 49 48 49
0002FF40	48 49 48 49 48 49 48 49 48 49 48 49 48 49 48 49
0002FF50	48 49 48 49 48 49 48 49 48 49 48 49 48 49 48 49
0002FF60	48 49 48 49 48 49 48 49 48 49 48 49 48 49 48 49
0002FF70	48 49 48 49 48 49 48 49 48 49 48 49 48 49 48 49
0002FF80	48 49 48 49 48 49 48 49 48 49 48 49 48 49 48 49
0002FF90	48 49 48 49 48 49 48 49 48 49 48 49 48 49 48 49
0002FFA0	48 49 48 49 48 49 48 49 48 49 48 49 48 49 48 49
0002FFB0	48 49 48 49 48 49 48 49 48 49 48 49 48 49 48 49
0002FFC0	48 49 48 49 48 49 48 49 48 49 48 49 48 49 48 49
0002FFD0	48 49 48 49 48 49 48 49 48 49 48 49 48 49 48 49
0002FFE0	48 49 48 49 48 49 48 49 48 49 48 49 48 49 48 49
0002FFF0	48 49 48 49 48 49 48 49 48 49 48 49 48 49 48 49
00030000	89 50 4E 47 0D 0A 1A 0A 00 00 00 0D 49 48 44 52
00030010	00 00 04 80 00 00 02 88 08 02 00 00 00 87 49 16
00030020	5A 00 00 00 01 73 52 47 42 00 AE CE 1C E9 00 00
00030030	00 04 67 41 4D 41 00 00 B1 8F 0B FC 61 05 00 00
00030040	00 09 70 48 59 73 00 00 0E C3 00 00 0E C3 01 C7
00030050	6F A8 64 00 00 FF A5 49 44 41 54 78 5E EC FD 05
00030060	70 24 D7 82 A0 0B 7B DF BF 6F F7 ED BC 9D 9D DD
00030070	C1 CB BE 66 68 DB DD 6E 06 B1 54 CC A8 22 95 98
00030080	4B C5 2C 86 66 E6 6E 61 31 93 18 9A DB CD 28 68
00030090	31 97 B0 C1 F6 BD 73 F7 CE CE CE 7B FF F8 3F 59
000300A0	A5 96 65 75 BB AF 61 C6 3B FB C7 F9 E2 8B 13 59
000300B0	D9 59 20 47 64 84 BF 38 99 27 5F FB 0A 02 81 40
000300C0	20 10 08 04 02 81 40 20 3F 09 30 C0 20 10 08 04
000300D0	02 81 40 20 10 08 E4 27 02 06 18 04 02 81 40 20
000300E0	10 08 04 02 81 FC 44 C0 00 83 40 20 10 08 04 02
000300F0	81 40 20 90 9F 08 18 60 10 08 04 02 81 40 20 10
00030100	08 04 F2 13 01 03 0C 02 81 40 20 10 08 04 02 81
00030110	40 7E 22 60 80 41 20 10 08 04 02 81 40 20 10 C8
00030120	4F 04 0C 30 08 04 02 81 40 20 10 08 04 02 F9 89
00030130	80 01 06 81 40 20 10 08 04 02 81 40 20 3F 11 30
00030140	C0 20 10 08 04 02 81 40 20 3F 16 1A 36 36 2C 15
00030150	13 03 A4 A0 A3 29 E8 58 44 54 7C 58 2A 3A 81 8A
00030160	46 D1 30 F9 B0 2C 12 01 C9 26 01 13 C0 14 26 95

오프셋: 2FAB1 블록 2FAB1-2FAB4

길이: 4

덮어쓰기

이후에 새로운 PNG 시그니처 부분을
파로 빼보니 아래와 같다.



파이도 잘되잖아 있다
앞선 4개의 파일을
잘 조합해줫으면 할 것 같다.

파일(F) 편집(E) 찾기(S) 보기(V) 분석(A) 기타 설정(X) 창 설정(W) ?

16 ANSI 16 진수

Annoy-o-Tron.png Gold PsychooTron.png

Offset(h)	00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F	
0003B320	A4 23 96 BE BE 9E ED 78 70 D0 99 D8 EF DA C9 D9	#-ñäixpØiÚÉÙ
0003B330	36 33 23 12 EB 20 A1 F7 13 FA 41 C8 30 D6 00 A1	63#.ë ï.úAÉOÖ.i
0003B340	07 47 B8 7A 3F 57 E7 2E D6 DB B7 72 4C 7A 2D EE	.G,z?Wç.ÖÜ·rLz-i
0003B350	EA 17 E3 86 E5 A4 7E 39 6D 8C EC 36 44 76 AF 45	è.ä+å~9mEi6Dv-E
0003B360	52 A6 48 BA 2E 92 AA 5D DA 85 E8 D2 CE C5 B5 E1	R;H°.'~]Ú...éòíÅuá
0003B370	2D ED 7C 5C 13 DE 86 F4 D2 84 B7 34 B3 D0 5D 9B	-i \.\.PtöÖ,,·4^D] >
0003B380	8C 90 61 C1 OD F5 F4 A6 7A 6A 53 3D 1E A9 B8 13	€.aÁ.öö;zjS=.@..
0003B390	51 05 96 C9 A1 05 D2 3F 4F 7A E7 94 EE 90 F2 16	Q.-É;.Ö?Ozç"i.ö.
0003B3A0	9E 80 21 08 82 20 08 F2 FF C0 B7 0F 30 04 41 10	ž€!., .öýÀ·.0.A.
0003B3B0	04 41 10 04 41 10 E4 BF 02 03 0C 41 10 04 41 10	.A..A.ä...A..A.
0003B3C0	04 41 10 E4 82 C0 00 43 10 04 41 10 04 41 10 04	.A.ä,À.C..A..A..
0003B3D0	B9 20 30 C0 10 04 41 10 04 41 10 04 41 2E 08 0C	' 0À..A..A..A...
0003B3E0	30 04 41 10 04 41 10 04 41 90 0B 02 03 0C 41 10	0.A..A..A.....A.
0003B3F0	04 41 10 04 41 10 E4 82 C0 00 43 10 04 41 10 04	.A..A.ä,À.C..A..
0003B400	41 10 04 B9 10 BE F9 E6 6F 0B 40 69 96 54 13 38	A..'.ñæo.®i-T.8
0003B410	AF 00 00 00 00 49 45 4E 44 AE 42 60 82 49 48 49IEEND@B`,IHI
0003B420	48 49 48 49 48 49 48 49 48 49 48 49 48 49 48 49	HIHIHIHIHIHIHIHI
0003B430	48 49 48 49 48 49 48 49 48 49 48 49 48 49 48 49	HIHIHIHIHIHIHIHI
0003B440	48 49 48 49 48 49 48 49 48 49 48 49 48 49 48 49	HIHIHIHIHIHIHIHI
0003B450	48 49 48 49 48 49 48 49 48 49 48 49 48 49 48 49	HIHIHIHIHIHIHIHI
0003B460	48 49 48 49 48 49 48 49 48 49 48 49 48 49 48 49	HIHIHIHIHIHIHIHI
0003B470	48 49 48 49 48 49 48 49 48 49 48 49 48 49 48 49	HIHIHIHIHIHIHIHI
0003B480	48 49 48 49 48 49 48 49 48 49 48 49 48 49 48 49	HIHIHIHIHIHIHIHI
0003B490	48 49 48 49 48 49 48 49 48 49 48 49 48 49 48 49	HIHIHIHIHIHIHIHI
0003B4A0	48 49 48 49 48 49 48 49 48 49 48 49 48 49 48 49	HIHIHIHIHIHIHIHI
0003B4B0	48 49 48 49 48 49 48 49 48 49 48 49 48 49 48 49	HIHIHIHIHIHIHIHI
0003B4C0	48 49 48 49 48 49 48 49 48 49 48 49 48 49 48 49	HIHIHIHIHIHIHIHI
0003B4D0	48 49 48 49 48 49 48 49 48 49 48 49 48 49 48 49	HIHIHIHIHIHIHIHI
0003B4E0	48 49 48 49 48 49 48 49 48 49 48 49 48 49 48 49	HIHIHIHIHIHIHIHI
0003B4F0	48 49 48 49 48 49 48 49 48 49 48 49 48 49 48 49	HIHIHIHIHIHIHIHI
0003B500	48 49 48 49 48 49 48 49 48 49 48 49 48 49 48 49	HIHIHIHIHIHIHIHI
0003B510	48 49 48 49 48 49 48 49 48 49 48 49 48 49 48 49	HIHIHIHIHIHIHIHI
0003B520	48 49 48 49 48 49 48 49 48 49 48 49 48 49 48 49	HIHIHIHIHIHIHIHI
0003B530	48 49 48 49 48 49 48 49 48 49 48 49 48 49 48 49	HIHIHIHIHIHIHIHI
0003B540	48 49 48 49 48 49 48 49 48 49 48 49 48 49 48 49	HIHIHIHIHIHIHIHI
0003B550	48 49 48 49 48 49 48 49 48 49 48 49 48 49 48 49	HIHIHIHIHIHIHIHI
0003B560	48 49 48 49 48 49 48 49 48 49 48 49 48 49 48 49	HIHIHIHIHIHIHIHI
0003B570	48 49 48 49 48 49 48 49 48 49 48 49 48 49 48 49	HIHIHIHIHIHIHIHI
0003B580	48 49 48 49 48 49 48 49 48 49 48 49 48 49 48 49	HIHIHIHIHIHIHIHI
0003B590	48 49 48 49 48 49 48 49 48 49 48 49 48 49 48 49	HIHIHIHIHIHIHIHI
0003B5A0	48 49 48 49 48 49 48 49 48 49 48 49 48 49 48 49	HIHIHIHIHIHIHIHI
0003B5B0	48 49 48 49 48 49 48 49 48 49 48 49 48 49 48 49	HIHIHIHIHIHIHIHI
0003B5C0	48 49 48 49 48 49 48 49 48 49 48 49 48 49 48 49	HIHIHIHIHIHIHIHI
0003B5D0	48 49 48 49 48 49 48 49 48 49 48 49 48 49 48 49	HIHIHIHIHIHIHIHI
0003B5E0	49 40 49 49 49 49 49 49 49 49 49 49 49 49 49 49	HTHTHTHTHTHTHTHT

오프셋: 2BE74

블록 2BE74-2BE77

길이: 4

덮어쓰기

Png의 Footer 시그니처가
이 파일에서 두번 나온것을 보면
이 파일속 파일에
나오는 내용이 맨 뒤의 내용 같다.

파일(F) 편집(E) 찾기(S) 보기(V) 분석(A) 기타 설정(X) 창 설정(W) ?

16 ANSI 16 진수

Annoy-o-Tron.png Gold PsychoTron.png psychotron.png

Offset(h)	00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00017E20	48 49 48 49 48 49 48 49 48 49 48 49 48 49 48 49
00017E30	HIHIHIHIHIHIHIHI
00017E40	48 49 48 49 48 49 48 49 48 49 48 49 48 49 48 49
00017E50	HIHIHIHIHIHIHIHI
00017E60	48 49 48 49 48 49 48 49 48 49 48 49 48 49 48 49
00017E70	HIHIHIHIHIHIHIHI
00017E80	48 49 48 49 48 49 48 49 48 49 48 49 48 49 48 49
00017E90	HIHIHIHIHIHIHIHI 5SèR..ýóIDATEíþ.
00017EA0	54 76 46 DF 25 E9 47 54 F7 30 55 F5 20 6B 7A 5B
00017EB0	TvF0%éGT=OUö kz[
00017EC0	EE 76 04 37 DF 05 D7 3A 83 CB DD 41 59 4F 90 DF
00017ED0	iv.78.*:fÉÝAYO.8
00017EE0	95 8C ED A0 DF FF 56 ED CA FF 57 FB CD FF 56 7F
00017EF0	•El ByViÉyWúiyV.
00017F00	F6 CA 83 25 FF BA 35 F7 1F 57 02 FF 7E DE EF 6F
00017F10	öÈf%y°5÷.W.y~Biø
00017F20	65 9E FF B8 E0 FB EE 79 9F EE E7 BC BB 41 CE 7A
00017F30	ežy,áúiyYíçl»Aíz
00017F40	75 2E F3 EC D4 4C C0 3A 22 01 C3 0E D6 B5 84 DB
00017F50	u.óíÖLÀ:".Ä.Öµ,,Ù
00017F60	AD 84 DB C3 0A 5A 43 EC 14 AF CB 09 7E 97 52 EF
00017F70	.,ÙÄ.ZCi.~E.~—Ri
00017F80	DE 67 BC 7B 9C F4 44 05 39 8A 3C BB 14 7A BC 9B
00017F90	Pg4{œöD.9š<».z4>
00017FA0	27 7C 27 53 F0 76 8E 7F C7 3B 2B 27 5B C2 67 81
00017FB0	'!S8vŽ.Ç;+'[Ag.
00017FC0	84 85 A6 98 E9 DA A8 A9 AA 68 5F E8 60 74 4D 0E
00017FD0	...!~éÚ@®h è`tM.
00017FE0	3B 01 13 47 78 C9 28 01 93 C7 CE C1 9F 87 27 A6
00017FF0	;..GxE(.“çÍÁY#”!
00018000	CD 5F 16 DB 05 FF FA B7 06 B7 32 69 A9 3C 03 OC
00018010	í_.Ù.yú..·2i@<..
00018020	FC 33 F9 FC 44 02 3F 86 35 B8 B5 6D 06 5E 8D C1
00018030	ü3üüD.?+5,µm.^ Á
00018040	6B 32 FC 4C A8 BC 80 D1 53 BF 37 76 83 4B A1 08
00018050	k2üL“éÑS;7vfK;
00018060	E1 03 2F 36 76 05 21 60 F0 81 D6 14 3B 13 68 B5
00018070	á./6v.!`8.Ö.;.hu
00018080	4B BA C1 34 FF D9 83 26 83 8F B5 B5 34 77 30 F8
00018090	KºÁ4yÜf&f.mu4w0ø
000180A0	D4 F0 B1 E7 13 FC 30 D6 E0 D6 96 D2 B3 69 55 CC
000180B0	Öö±ç.ü0öAö-Ö³iUI
000180C0	Vp).fOÖ.Uú..?5..
000180D0	56 DE 29 0A 83 4F D4 1A DC FA 02 03 3F 35 03 07
000180E0	.Ä.B(°±5. “<x.S>
000180F0	0E C4 0F DF 28 B0 B1 35 7F A0 98 3C D7 90 24 9B
00018100	‘.,3<48.i\$R=.öe)
00018110	B4 B8 03 33 8B 34 38 02 ED 24 52 3D 1E F5 65 29
00018120	F.å(“ÈG.Ù.R>†”.
00018130	46 02 E5 28 B4 85 C8 47 03 D9 18 52 3E 86 94 8D
00018140	#mc.lêH!%£ %Ákzb
00018150	23 A4 63 09 31 EA 48 21 25 A3 A0 89 C1 6B 7A 42
00018160	36 92 62 34 35 FB 6B 02 50 4C 01 0A A7 46 02 E6
00018170	6’b45ük.PL..SF.æ
00018180	09 4C 0B 80 F1 13 B4 04 B3 7E 15 A9 A5 4A 20 6A
00018190	L.€ñ.‘.^~.Ø¥J j
000181A0	0E A0 89 5E 56 01 23 B5 FB 2C BA 7D 16 ED 01 B3
000181B0	. %^V.#mu,º}.i.^
000181C0	06 D2 64 F1 E5 26 D8 79 97 0D E8 4B A8 DB 8A FE
000181D0	.Ödfñå&Øy-.èK”Ùsp
000181E0	72 1F A9 DC 6D 96 ED 30 4A B6 3D 56 C0 90 83 6D
000181F0	r.ØUm-iOJ¶=VÀ.fm
00018200	B7 CE 04 A3 04 8C FE 5E 78 4A 34 F8 AE F0 1D 02
00018210	·í.‡.Gp^xJ4øø§..
00018220	D3 3E 60 DC 05 8C 5B 81 7E 23 D0 AE 21 D5 3F 10
00018230	Ó>`Ù.G[.~#Ðø!Ó?
00018240	AA AF 80 6A 19 50 BD 0F D4 73 80 CA 1F A8 BC B1
00018250	“-€j.P¶.Ôs€È.”‡
00018260	83 A1 4E 3F 37 A0 E6 52 25 07 9B A1 46 03 0E ED
00018270	f;N?7 æR%.>F..í
00018280	80 02 66 54 F0 4C 4A 21 3D 0D 0C 75 6A E9 BD 80
00018290	€.FT8LJ!=..ujé¤€
000182A0	D1 0B 98 BC 81 5E 68 85 8F 30 70 F1 0E 36 2E 11
000182B0	N.“‡.^h....Opñ.6..
000182C0	DE 41 2D 10 2E 50 BB 23 D0 DA 6B 6E 40 E1 0A E4
000182D0	PA-..P»#ÐÚkn@á.ä
000182E0	2E A4 CC 99 94 4F 06 F2 F1 F8 7D 54 8C A0 18 06
000182F0	.ñí”“O.øñø}TE ..
00018300	94 43 80 F2 3D AA B6 0A BD DC 76 4F 8A EE 18 65
00018310	”C€ð=“¶.¤ÜvoŠi.e
00018320	6F 70 F9 05 C4 DD CG DA 2F 44 C5 3B 06 7B F9 G1
00018330	~ à Kññññ nññ..!~

오프셋: 0

덮어쓰기

이 파일의 H1 이후 내용이
중간 내용으로 추정된다.

이 세개의 부분을 합쳐보니
풀캐그가 보인다
아니 gif 부분에서
아래부분이 있는 것 같다.

{ 4NNONG_

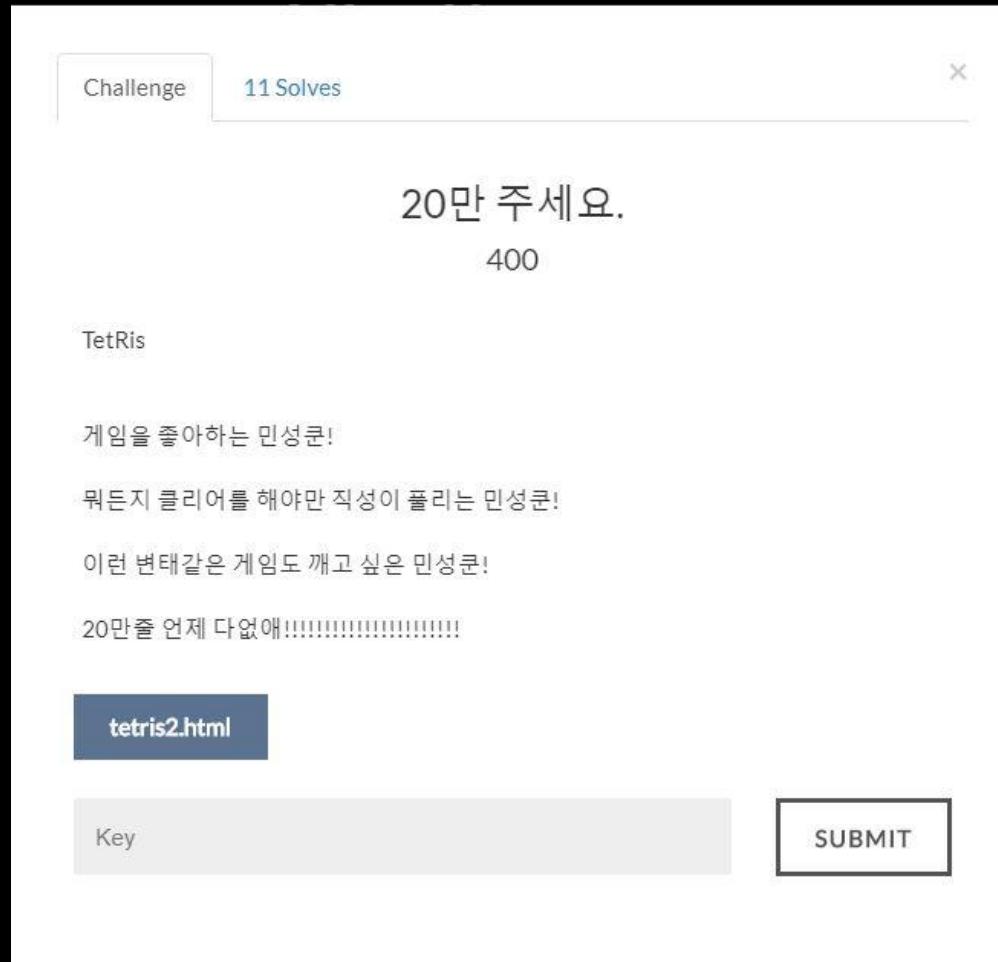


M4NGN4NONG }



{ Web }

> 20만 주세요 <



TE2RIS

GAME MODE 200000 LINE

GHOST MODE OFF

BLOCK PREVIEW 1

HOLDING 3

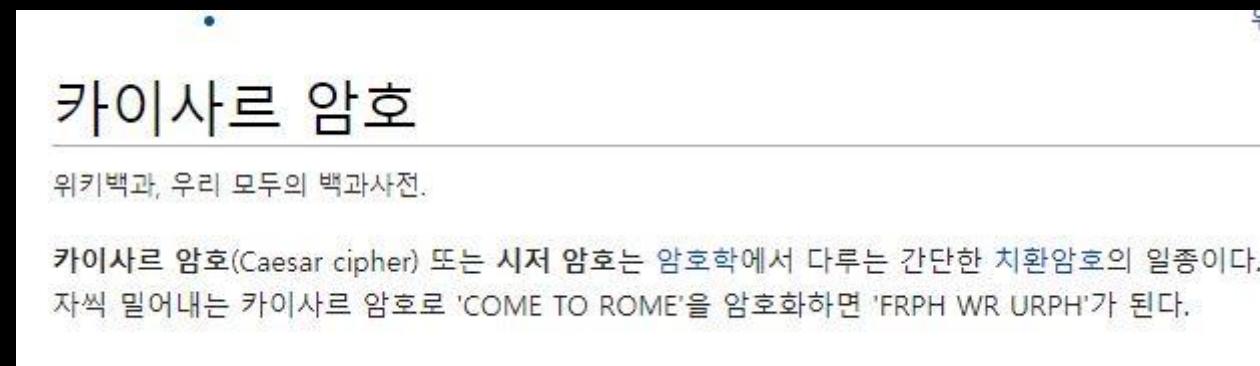
START GAME

Control
UP : ROT
SPACE : DROP
SHIFT : HOLD
G : GHOST MODE ON, OFF

```
var mapInt=[], blockInt=[], nextblo
4 var nextBlock=[];
5 var footer = "10_XQDJQY";
6
7 function putBlock(x,y){
```

테트리스화면이 보인다.
이단 소스코드부터 확인해보는데
뭐걸 떠하니 풀려고가 보인다.

보통 Flag is 로 시작하는데
앞의 문자가 이상하다.



카이사르 암호의 형식이다.

Vbqw yi {00k_qhu_I0_XQDJQY}
Flag is {Y0u_are_S0_HANTAI}

ABCDEFGHIJKLMNOPQRSTUVWXYZ
QRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ

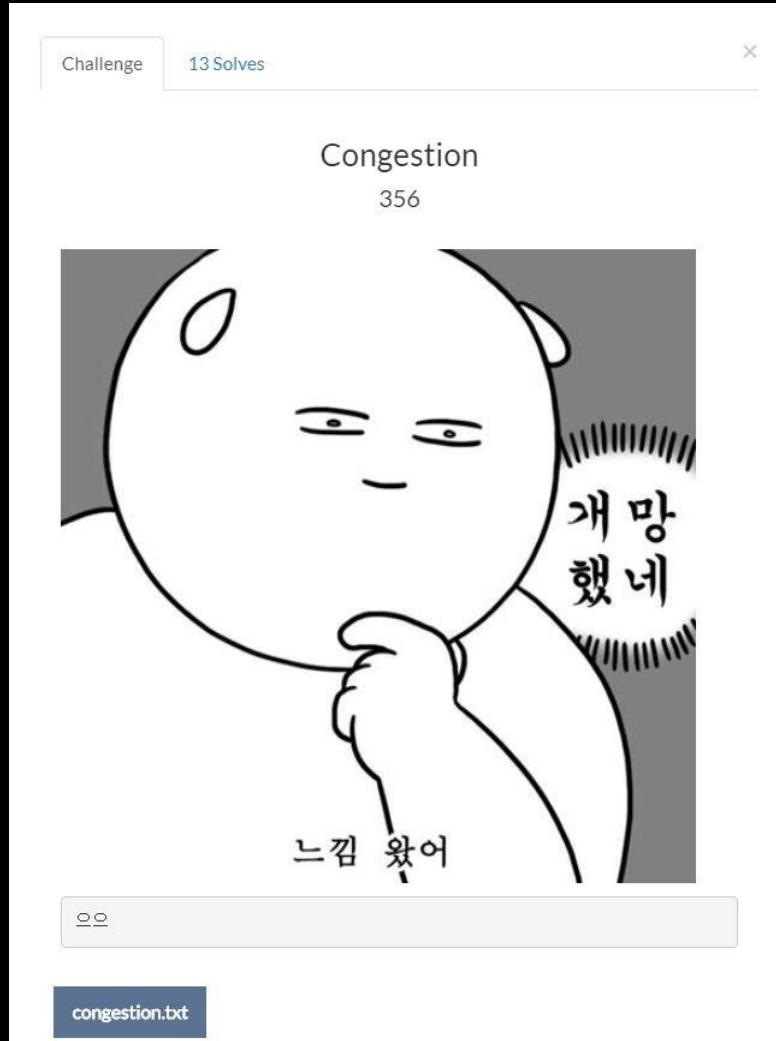
소스코드를 더 보면

대문자, 소문자에 대응하는

변환을 하고 있다.

변환하는 풀어쓰기 보인다.

{ Crypto } > Congestion <



구성을 보니 Brain F*** 언어로 이루어져 있다.

브레인퍽(Brainfuck)은 우어반 뮐러(Urban Müller)가 1993년 경에 만든 최소주의 컴퓨터 프로그래밍 언어이다. 이름에 포함된 뮐러는 그의 이름을 뜻하는 독일어이다.

목차 [숨기기]

- 1 언어의 설계
 - 1.1 명령어들
- 2 해설
- 3 예제
 - 3.1 헬로 월드 프로그램
 - 3.2 ROT13
- 4 관련 항목
- 5 외부 링크

언어의 설계 [편집]

뮐러는 가장 작은 컴파일러로 구현할 수 있는 간단하면서도 티링 완전한 프로그래밍 언어를 만드는 것이 목적이었다. 이 언어는 최소화된 프로그래밍 언어이자, 컴파일러 크기가 1024바이트인 False의 영향을 받았다.

이름이 말해 주듯이, 브레인퍽 프로그램은 이해하기 어려운 경향이 있다. 하지만 티링 기계는 컴퓨터가 할 수 있는 모든 연산을 표현하는 프로그램 외에, 0으로 초기화된 바이트 단위의 배열과, 처음에 배열의 맨 첫 바이트를 가리키는 포인터, 그리고

명령어들 [편집]

여덟 개의 명령어들은 각각 한 개의 문자로 구성되어 있으며 다음과 같다:

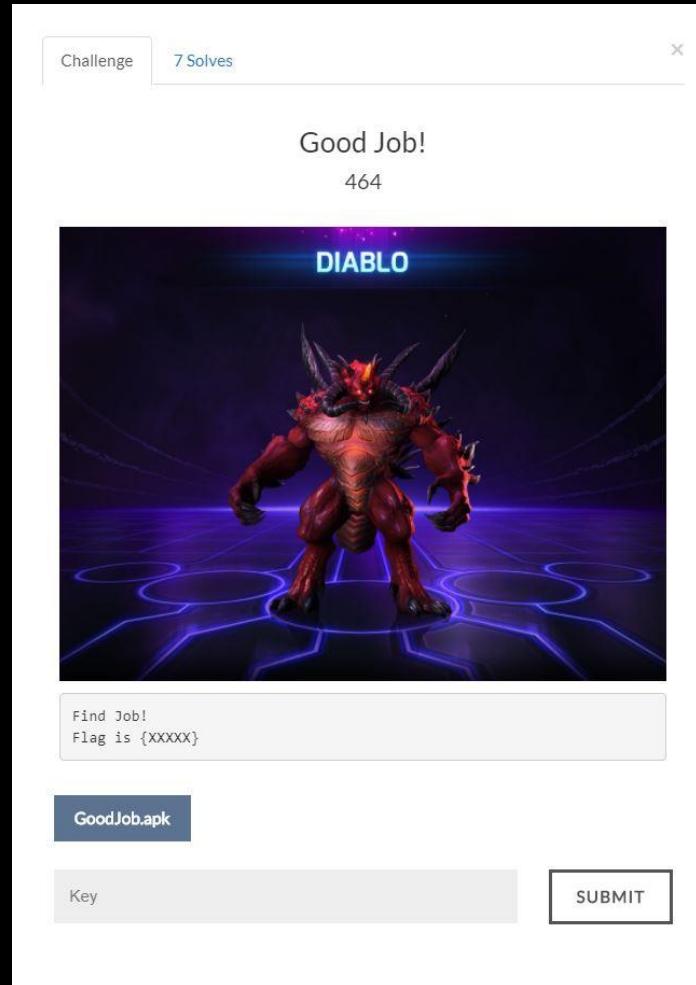
문자	의미
>	포인터를 증가시킨다.
<	포인터를 감소시킨다.
+	포인터가 가리키는 바이트의 값을 증가시킨다.
-	포인터가 가리키는 바이트의 값을 감소시킨다.
.	포인터가 가리키는 바이트의 값을 ASCII 문자로 출력한다.
,	포인터가 가리키는 바이트에 입력받은 문자의 ASCII 값을 넣는다.

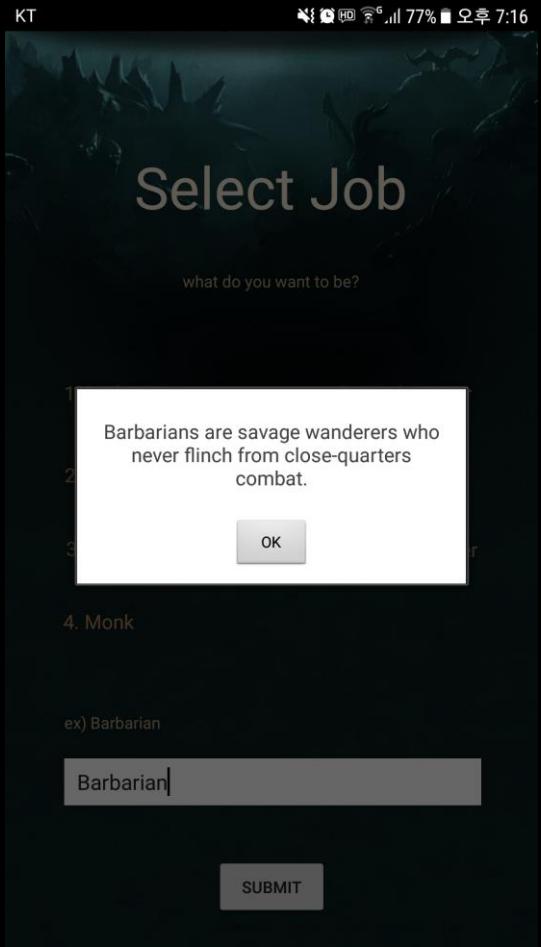
The screenshot shows a Brainfuck interpreter window. At the top, there are several configuration options: 'Large variables:' (unchecked), 'Prompt for input:' (unchecked), 'functions:' (containing 'add', 'dup', 'swap', 'mul', 'if'), 'Alert when finished:' (unchecked), and 'code ^' (checkbox). Below these are 'execute' and 'clear' buttons. The main area has an 'input:' field containing the text: "The impacts of tourism on the environment are evident to scientists, but not all residents attribute environmental damage to tourism. Residents commonly have positive views on the economic and some sociocultural influences of tourism on quality of life, but their reactions to environmental impacts are mixed. Some residents feel tourism provides more parks and recreation areas, improves the quality of the roads and public facilities, and does not contribute to ecological decline. Many do not blame tourism for traffic problems, overcrowded outdoor recreation, or the disturbance of peace and tranquility of parks. Alternatively, some residents express concern that tourists overcrowd the local fishing, hunting, and other recreation areas or may cause traffic and pedestrian congestion. Flag is {Brain_FXXk_Language_Is_Real_FXXk} Some studies suggest that variations in residents' feelings about tourism's relationship to environmental damage are related to the type of tourism, the extent to which residents feel the natural environment needs to be protected, and the distance residents live from the tourist attractions." An 'output:' field at the bottom also contains this text and has a 'clear' button.

해석해보니
장문의 글 속에 Flag가 보인다.

{ Digital APP }

> Good Job <





여기서 대로
Barbarian 입체화
Barbarian에 대한
정보가 뜬다.

Dex 2 jar 프로그램과
Java Decompiler を 이용하여
프로그램 미디 앤드偏偏의
자바 코드를 확인해보았다.

```
for (;;)
{
    switch (i)
    {
        default:
            Toast.makeText(MainActivity.this.getApplicationContext(), "Incorrect answer.",  
                return;
        if (paramAnonymousView.equals("Barbarian"))
        {
            i = 0;
            continue;
        if (paramAnonymousView.equals("Crusader"))
        {
            i = 1;
            continue;
        if (paramAnonymousView.equals("Demon Hunter"))
        {
            i = 2;
            continue;
        if (paramAnonymousView.equals("Monk"))
        {
            i = 3;
            continue;
        if (paramAnonymousView.equals("Witch Doctor"))
        {
            i = 4;
            continue;
        if (paramAnonymousView.equals("Wizard"))
        {
            i = 5;
            continue;
        if (paramAnonymousView.equals("Necromancer"))
        {
            i = 6;
            continue;
        if (paramAnonymousView.equals("Security Person of K.knock!!!"))
        {
            i = 7;
        }
    }
}
}
}
}
break;
```

입력값을 Equals 함수로
이용하여 비교하고
값에 대해
판단하는 것 같다.

```
:sswitch_0
const-string v11, "Barbarian"

invoke-virtual {v0, v11}, Ljava/lang/String;.>equals(Ljava/lang/Object;)Z
move-result v11

if-eqz v11, :cond_1

const/4 v9, 0x0

goto :goto_1

:sswitch_1
const-string v11, "Crusader"

invoke-virtual {v0, v11}, Ljava/lang/String;.>equals(Ljava/lang/Object;)Z
move-result v11

if-eqz v11, :cond_1

move v9, v10

goto :goto_1
```

Smali로
보니
switch 구조가
확인된다.

두 가지 풀이가 가능했다.

1. 모든 경우에 짐이 70이 되게 만든다
2. 해당 구문을 입증한다.

```
:sswitch_0
const-string v11, "Barbarian"

invoke-virtual {v0, v11}, Ljava/lang/String;.>equals(Ljava/lang/Object;)Z
move-result v11

if-eqz v11, :cond_1

const/4 v9, 0x0

goto :goto_1

:sswitch_1
const-string v11, "Crusader"

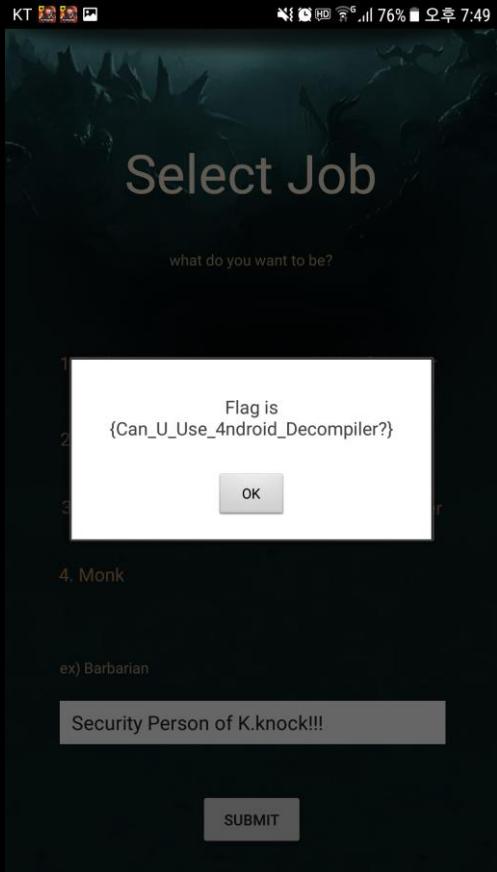
invoke-virtual {v0, v11}, Ljava/lang/String;.>equals(Ljava/lang/Object;)Z
move-result v11

if-eqz v11, :cond_1

move v9, v10

goto :goto_1
```

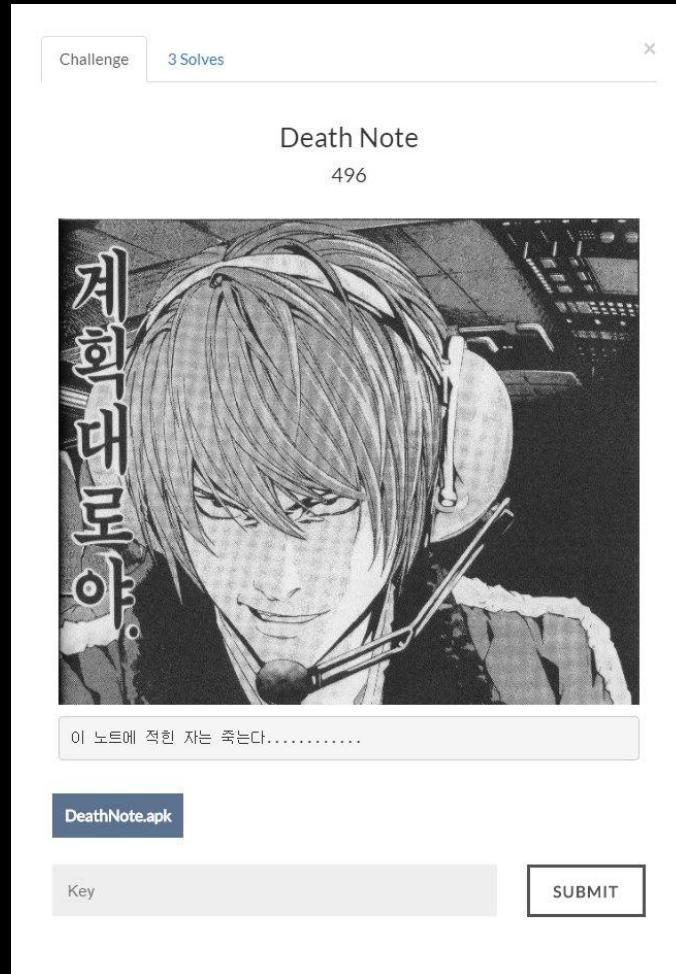
다른문제 풀이는
const/4 v9, 0x0부분을
0x7로 바꾸면된다.



어떤 방법을 써도
풀꺼그를 볼 수 있다.

{ Digital APP }

> Death Note <



DEATHNOTE

DEATH NOTE

The human whose name is written in this note shall die.

This note will not take effect unless the writer has the person's face in their mind when writing his/her name.

Therefore, people sharing the same name will not be affected.

If the cause of death is written within the next 40 seconds of writing the person's name, it will happen.

If the cause of death is not specified, the person will simply die of a heart attack.

학점

SUBMIT

I can't kill T^T

안죽이려
다행이다.

```
public void onClick(View paramAnonymousView)
{
    paramAnonymousView = localEditText.getText().toString();
    if (MainActivity.this.str.equals(paramAnonymousView))
    {
        localTextView.setText("Flag is {" + paramBundle.substring(7, 19) + "}");
        return;
    }
    Toast.makeText(MainActivity.this.getApplicationContext(), "I can't kill T^T", 0).show();
}
});

public native String stringFromJNI();
```

if 조건을 만족하면
flag를 볼 수 있으니
강제로 만족하도록
smali 코드를 손대자

```
.line 41
.local v0, "name":Ljava/lang/String;
iget-object v1, p0, Lcom/apple/deathnote/MainActivity$1;->this$0:Lcom/apple/deathnote/MainActivity;
iget-object v1, v1, Lcom/apple/deathnote/MainActivity;->str:Ljava/lang/String;
invoke-virtual {v1, v0}, Ljava/lang/String;->equals(Ljava/lang/Object;)Z
move-result v1
if-nez v1, :cond_0
```

```
.line 42
iget-object v1, p0, Lcom/apple/deathnote/MainActivity$1;->val$text1:Landroid/widget/TextView;
new-instance v2, Ljava/lang/StringBuilder;
invoke-direct {v2}, Ljava/lang/StringBuilder;-><init>()V
const-string v3, "Flag is {"
```

eqz (== 0) 와 같고
nez (!= 0) 으로 수정

```
C:\Users\Admin\Android>java -jar apktool_2.3.0.jar b DeathNote -o dn.apk
I: Using Apktool 2.3.0
I: Checking whether sources has changed...
I: Smaling smali folder into classes.dex...
I: Checking whether resources has changed...
I: Building apk file...
I: Copying unknown files/dir...

C:\Users\Admin\Android>java -jar sign.jar dn.apk
C:\Users\Admin\Android>java -jar sign.jar dn.apk --override
```

수정후 빌드하고 sign해주었다.

DEATHNOTE

DEATH NOTE

Flag is {zNDM4ZjNiZDV}

○ㄹ

SUBMIT

Flag를 얻었다.