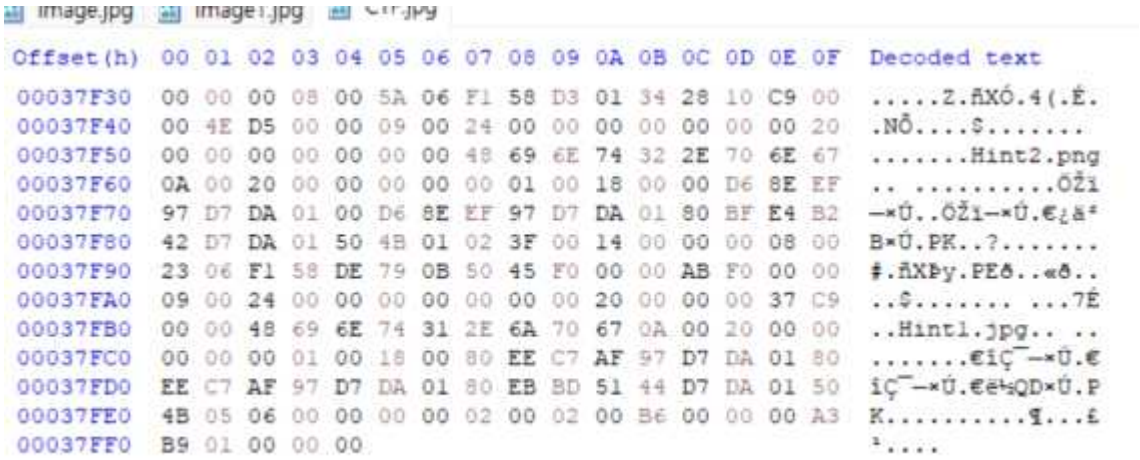
Broken hearted 문제 풀이

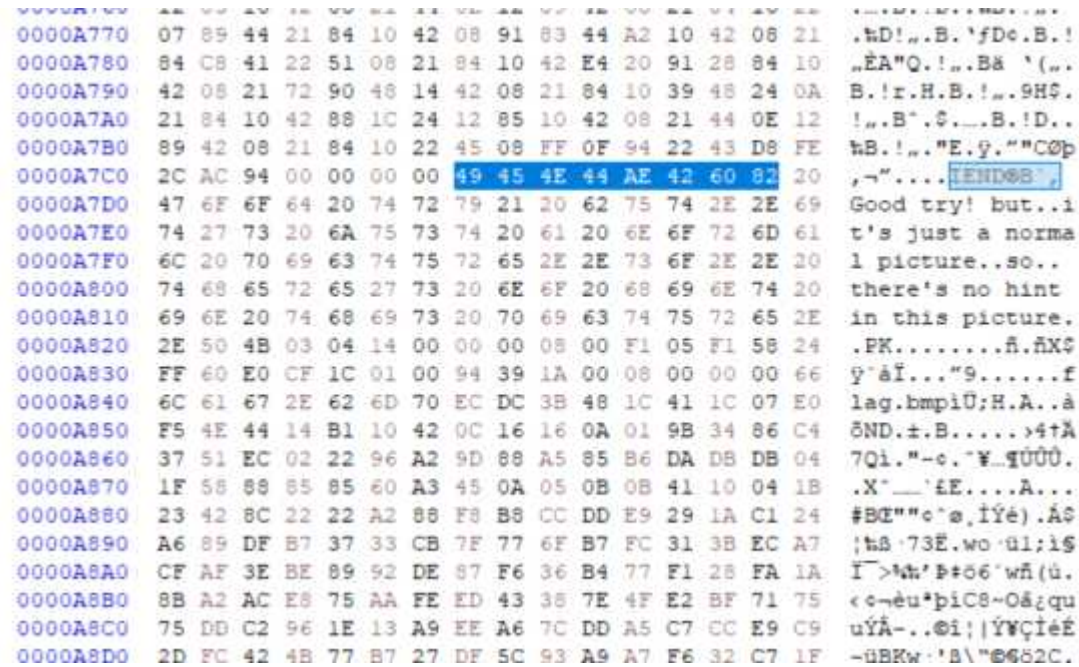


Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00037F30	00	00	00	08	00	5A	06	F1	58	D3	01	34	28	10	C9	00	.....Z.ñXÓ.4(.É.
00037F40	00	4E	D5	00	00	09	00	24	00	00	00	00	00	00	00	20	.NÕ....$.....
00037F50	00	00	00	00	00	00	00	48	69	6E	74	32	2E	70	6E	67	.......Hint2.png
00037F60	0A	00	20	00	00	00	00	00	01	00	18	00	00	D6	8E	EF	.. ..ÖŽi
00037F70	97	D7	DA	01	00	D6	8E	EF	97	D7	DA	01	80	BF	E4	B2	—×Ú..ÖŽi—×Ú.€¿ä²
00037F80	42	D7	DA	01	50	4B	01	02	3F	00	14	00	00	00	08	00	B×Ú.PK..?.....
00037F90	23	06	F1	58	DE	79	0B	50	45	F0	00	00	AB	F0	00	00	#.ñXÞy.PEð..«ð..
00037FA0	09	00	24	00	00	00	00	00	00	00	20	00	00	00	37	C9	..$..... ...7É
00037FB0	00	00	48	69	6E	74	31	2E	6A	70	67	0A	00	20	00	00	..Hint1.jpg.. ..
00037FC0	00	00	00	01	00	18	00	80	EE	C7	AF	97	D7	DA	01	80	.......€îÇ¯—×Ú.€
00037FD0	EE	C7	AF	97	D7	DA	01	80	EB	BD	51	44	D7	DA	01	50	îÇ¯—×Ú.€ë½QD×Ú.P
00037FE0	4B	05	06	00	00	00	00	02	00	02	00	B6	00	00	00	A3	K..........¶...£
00037FF0	B9	01	00	00	00												¹....

맨 밑에 뭔가 힌트가 잔뜩 쓰여있었다. jpg 어쩌고 png 어쩌고
Maybe..you can search for LSB Steganography..and..Always check the end carefully.

진짜 어쩌라는 건지 모르겠는데 일단 파일 형식이 png니까 푸터 넘버를 찾아봤다.

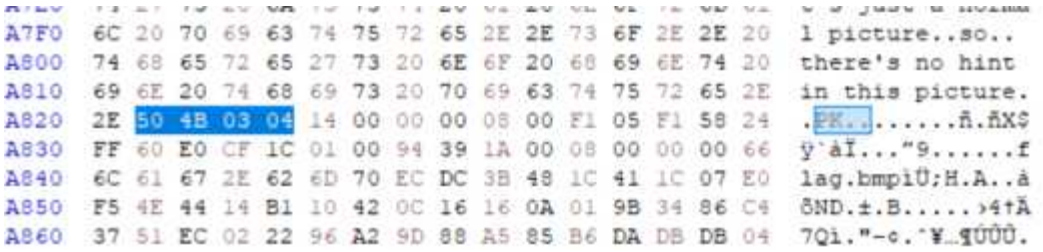
검색 결과



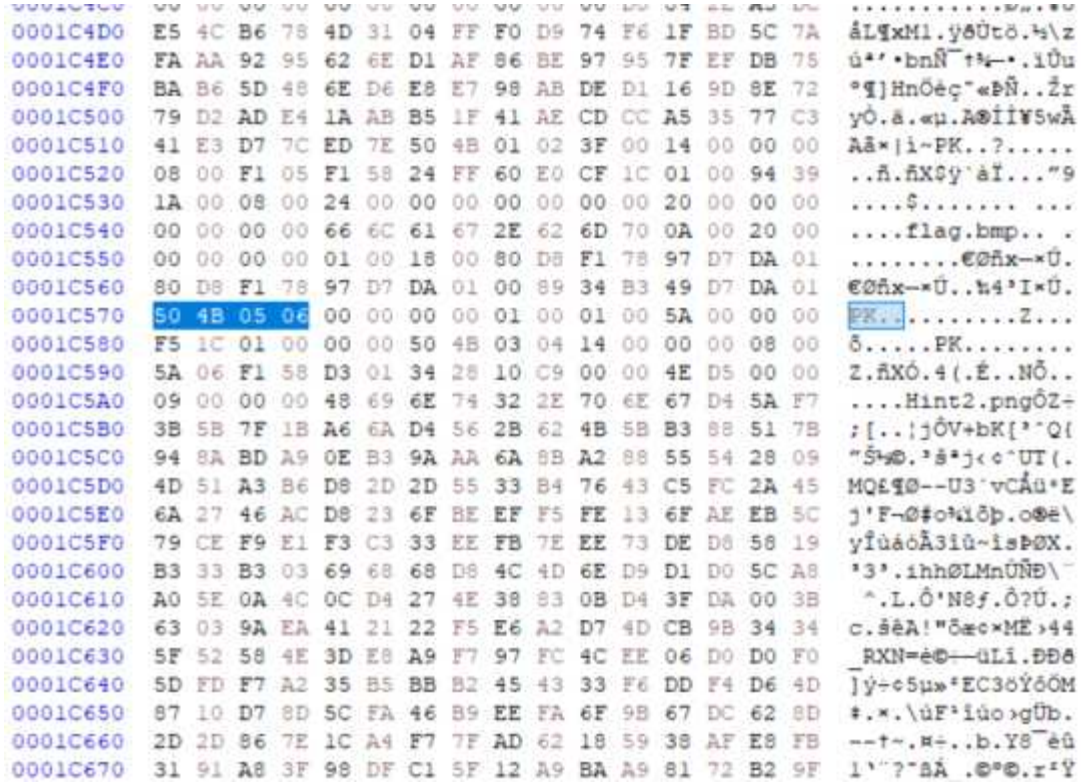
0000A770	07	89	44	21	84	10	42	08	91	83	44	A2	10	42	08	21	.‰D!„.B.'ƒD¢.B.!
0000A780	84	C8	41	22	51	08	21	84	10	42	E4	20	91	28	84	10	„ÈA"Q.!„.Bä '(„.
0000A790	42	08	21	72	90	48	14	42	08	21	84	10	39	48	24	0A	B.!r.H.B.!„.9H$.
0000A7A0	21	84	10	42	88	1C	24	12	85	10	42	08	21	44	0E	12	!„.Bˆ.$.…B.!D..
0000A7B0	89	42	08	21	84	10	22	45	08	FF	0F	94	22	43	D8	FE	‰B.!„."E.ÿ.""CØþ
0000A7C0	2C	AC	94	00	00	00	00	49	45	4E	44	AE	42	60	82	20	,¬"....IEND®B`‚
0000A7D0	47	6F	6F	64	20	74	72	79	21	20	62	75	74	2E	2E	69	Good try! but..i
0000A7E0	74	27	73	20	6A	75	73	74	20	61	20	6E	6F	72	6D	61	t's just a norma
0000A7F0	6C	20	70	69	63	74	75	72	65	2E	2E	73	6F	2E	2E	20	l picture..so..
0000A800	74	68	65	72	65	27	73	20	6E	6F	20	68	69	6E	74	20	there's no hint
0000A810	69	6E	20	74	68	69	73	20	70	69	63	74	75	72	65	2E	in this picture.
0000A820	2E	50	4B	03	04	14	00	00	00	08	00	F1	05	F1	58	24	.PK.....ñ.ñX$
0000A830	FF	60	E0	CF	1C	01	00	94	39	1A	00	08	00	00	00	66	ÿ`àÏ...”9.....f
0000A840	6C	61	67	2E	62	6D	70	EC	DC	3B	48	1C	41	1C	07	E0	lag.bmpìÜ;H.A..à
0000A850	F5	4E	44	14	B1	10	42	0C	16	16	0A	01	9B	34	86	C4	õND.±.B.....›4†Ä
0000A860	37	51	EC	02	22	96	A2	9D	88	A5	85	B6	DA	DB	DB	04	7Qì."–¢.ˆ¥…¶ÚÛÛ.
0000A870	1F	58	88	85	85	60	A3	45	0A	05	0B	0B	41	10	04	1B	.Xˆ……`£E....A...
0000A880	23	42	8C	22	22	A2	88	F8	B8	CC	DD	E9	29	1A	C1	24	#BŒ""¢ˆø¸ÌÝé).Á$
0000A890	A6	89	DF	B7	37	33	CB	7F	77	6F	B7	FC	31	3B	EC	A7	¦‰ß·73Ë.wo·ü1;ì§
0000A8A0	CF	AF	3E	BE	89	92	DE	87	F6	36	B4	77	F1	28	FA	1A	Ï¯>¾‰'Þ‡ö6´wñ(ú.
0000A8B0	8B	A2	AC	E8	75	AA	FE	ED	43	38	7E	4F	E2	BF	71	75	‹¢¬èuªþíC8~Oâ¿qu
0000A8C0	75	DD	C2	96	1E	13	A9	EE	A6	7C	DD	A5	C7	CC	E9	C9	uÝÂ–..©î¦|Ý¥ÇÌéÉ
0000A8D0	2D	FC	42	4B	77	B7	27	DF	5C	93	A9	A7	F6	32	C7	1F	-üBKw·'ß\"©§ö2Ç.

좋은 시도였다며 칭찬해준다. 근데 그냥 보통의 사진이라 이 사진에 힌트가 없다고 써있어서 또 다시 막혔다.

다음으로는 여러 파일 형식을 검색해보았다. 푸터 넘버는 여러 결과가 나오는 걸 보니 우연히 그 형식이 나올 수 있는 건가 보다. 헤더 넘버가 더 중요한 것 같다. 그리고 zip 파일 넘버를 검색해봤다. 아까 시도는 좋았다고 칭찬해준 그 문장 바로 뒤에서 시작됐다. 왜 zip 헤더 넘버가 있는 걸까? 푸터 넘버도 있을까?

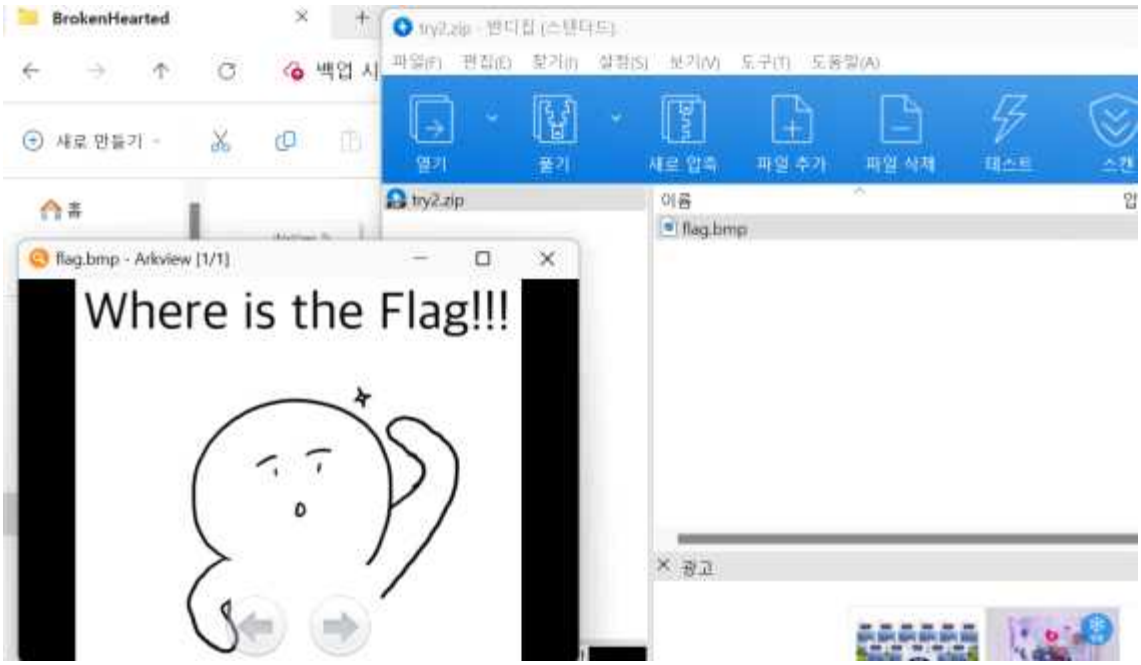


zip 파일 푸터 넘버도 존재했다. 그러면 위아래가 다 있는 건데...
한 번 이 부분을 전부 선택해서 따로 새 파일에 붙여넣기 해봤다.



그리고 .zip 파일로 저장을 했더니

압축 파일이 생겼고 그 안에 bmp 파일이라는 처음 보는 파일 형식이 들어있었다.
드디어 깃발을 찾은 걸까??????????????????????????????

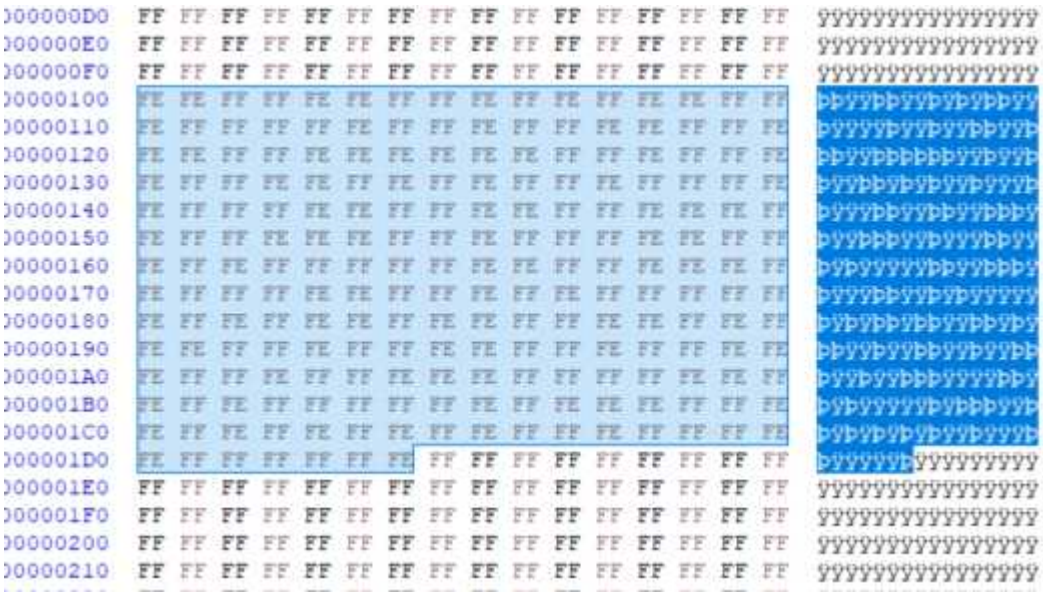


아니었다. '플래그 어딨어!!!' 하면서 또 뭔가 해야한다고 알려주고 있었다.
어떻게 해야 하지?
일단 이 파일을 열어봤다.

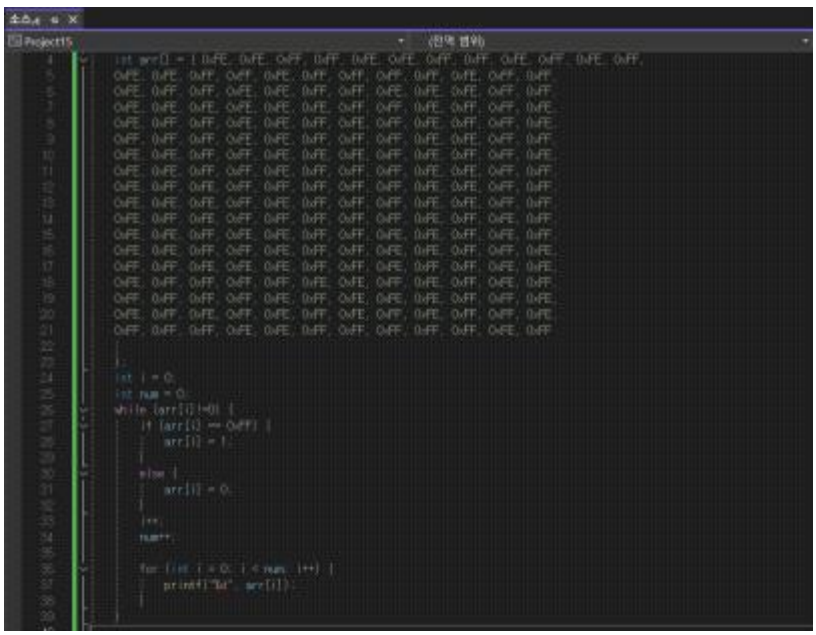
Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
001A3900	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	yyyyyyyyyyyyyyyy
001A3910	FF	FF	FF	FF	FF	FF	20	4C	6F	6F	6B	20	61	74	20	74	yyyyyy Look at t
001A3920	68	65	20	64	69	66	66	65	72	65	6E	74	20	48	65	78	he different Hex
001A3930	20	76	61	6C	75	65	73	20	6F	6E	20	6C	69	6E	65	73	values on lines
001A3940	20	30	30	30	30	30	31	30	30	20	74	6F	20	30	30	30	00000100 to 000
001A3950	30	35	30	30	30	2E	20	54	68	65	20	66	6C	61	67	20	05000. The flag
001A3960	62	65	67	69	6E	73	20	77	69	74	68	20	46	45	20	61	begins with FE a
001A3970	6E	64	20	63	6F	6E	73	69	73	74	73	20	6F	66	20	61	nd consists of a
001A3980	20	74	6F	74	61	6C	20	6F	66	20	32	31	36	20	62	79	total of 216 by
001A3990	74	65	73	2E													tes.

맨 밑에 힌트가 있었다.
100부터 5000사이에 있는 값들을 살펴봐야 할 것 같았다. 뭔가 혼자만 다른 부분이고 FE로 시작되고 216 바이트 부분이랜다.

그거는 100부터 시작되는 이 216 바이트 부분을 말하는 것 같았다.
5000 부분까지 나머지는 죄다 FF 밖에 없으니까..



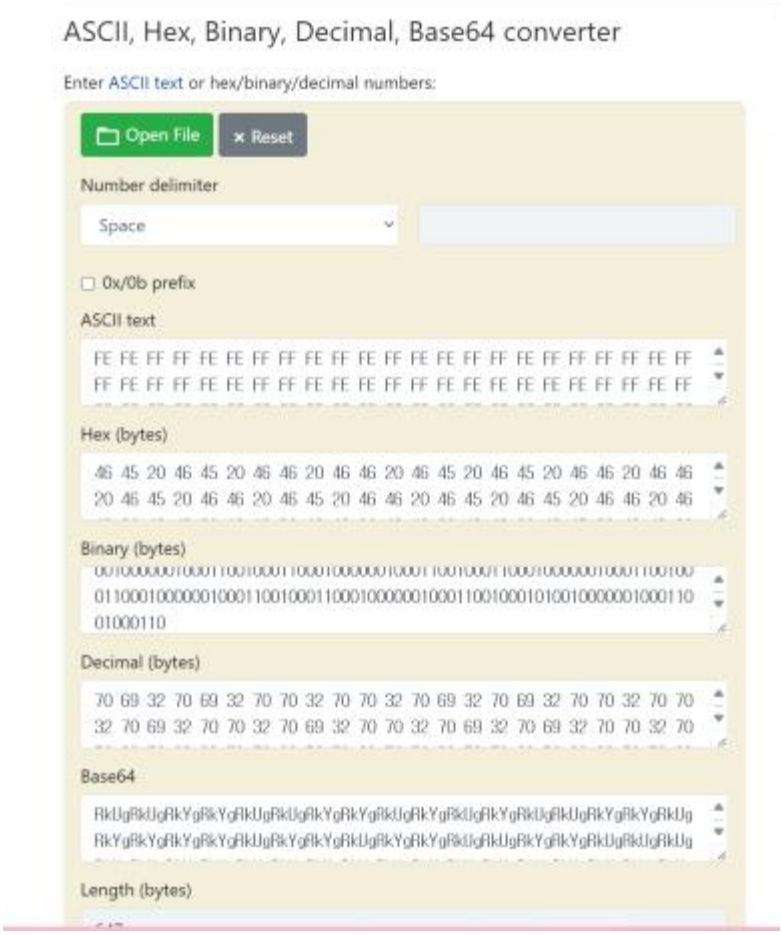
근데 여기까지 오는 것도 오래 걸렸는데 이 부분에서 한 시간 반 동안 계속 해멘 것 같다.
지금까지 두 문제 열어보고 다 실패하고 이게 세 번째였는데 이것마저 못 풀면 플래그(학생회)로서 체면도 안 서고 또 팀에 민폐일 것 같아서 계속 포기 안 하고 서치에 서치에 서치를 거듭했다.
이걸 무슨 2진수로 변환을 하고 디코딩을 하고 난리난리를 쳐봤는데도 나오지 않았다.



남이 비슷한 ctf 문제를 풀면서 FE FF 이런 값들을 변환을 해줘야 한다고 하던데
그래서 그 코드를 따라해봤는데 망했다.
C 공부 더 해야겠다.

아스키코드든 2진수든 디코딩이든 그걸 정확히 차이점을 모르니 그냥 계속 상호변환 하고 있었는데
문득 갑자기 저 C언어를 코딩한 사람은 왜 했더라 하고 생각해보았다.

그러니까 FE가 0이고 FF가 1이라는 건데...
어떤 사람들은 노가다로 바꿨다고 하고, 어떤 사람은 저렇게 코드를 짜서 변환했다고 하고,,,

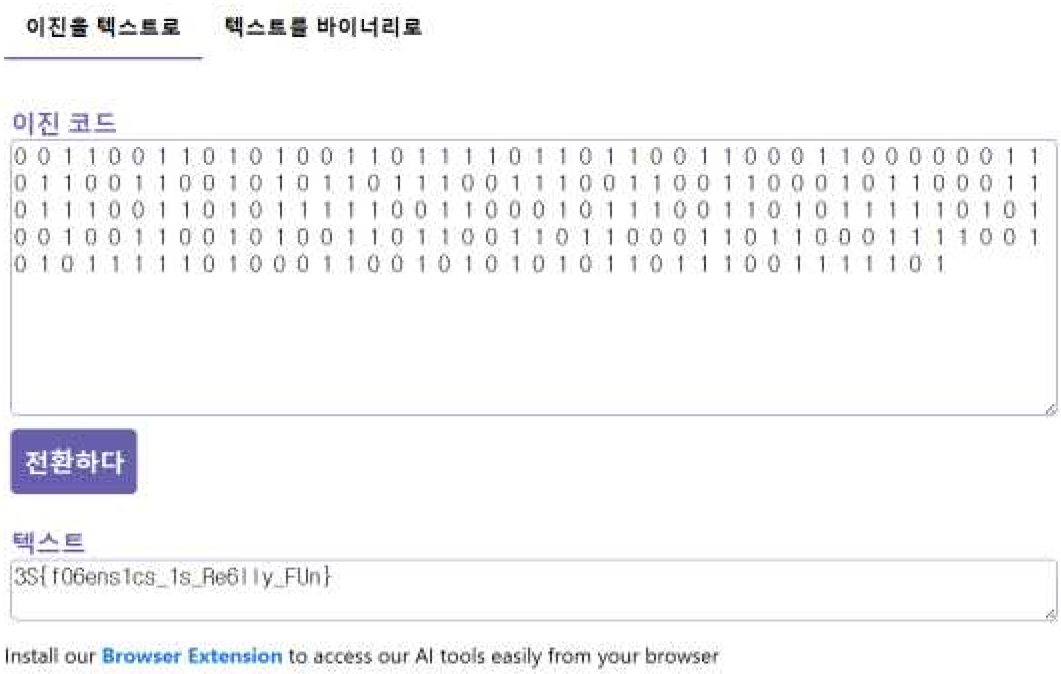


그러면 이 FF FE 이런 숫자들을 여기서 변환하는 게 아니었던 건가??????????
2진수 변환 잘 해주길래 이런 사이트 이용해서 변환해야 하는 줄 알았는데... 아니었나 보다.
C 코딩 실패한 김에 박후린 교수님께 죄송한 마음을 담아 노가다 변환으로 속죄하려다가...
나는 그것보단 좀 더 머리를 굴릴 줄 알았다.

한글로 갈 것도 없이 그냥 메모장에도 글자 변환 기능이 있었다. 그래서 FE는 0으로 FF는 1로 변환을 해주었다.

0 0 1 1 0 0 1 1 0 1 0 1 0 0 1 1 0 1 1 1 1 0 1 1 0 1 1 0 0 1 1 0 0 0 1 1 0 0 0 0
0 0 1 1 0 1 1 0 0 1 1 0 0 1 0 1 0 1 1 0 1 1 1 0 0 1 1 1 0 0 1 1 0 0 1 1 0 0 0 1
0 1 1 0 0 0 1 1 0 1 1 1 0 0 1 1 0 1 0 1 1 1 1 1 0 0 1 1 0 0 0 1 0 1 1 1 0 0 1 1
0 1 0 1 1 1 1 1 0 1 0 1 0 0 1 0 0 1 1 0 0 1 0 1 0 0 1 1 0 1 1 0 0 1 1 0 1 1 0 0
0 1 1 0 1 1 0 0 0 1 1 1 1 0 0 1 0 1 0 1 1 1 1 1 0 1 0 0 0 1 1 0 0 1 0 1 0 1 0 1
0 1 1 0 1 1 1 0 0 1 1 1 1 1 0 1

아까 사이트에서 변환하던 값이랑 뭔가 다른 게 나왔고



이진수를 텍스트로 변환했더니 나왔다!!!
내 첫 플래그다