# Writeup for Cybersea CTF 2021

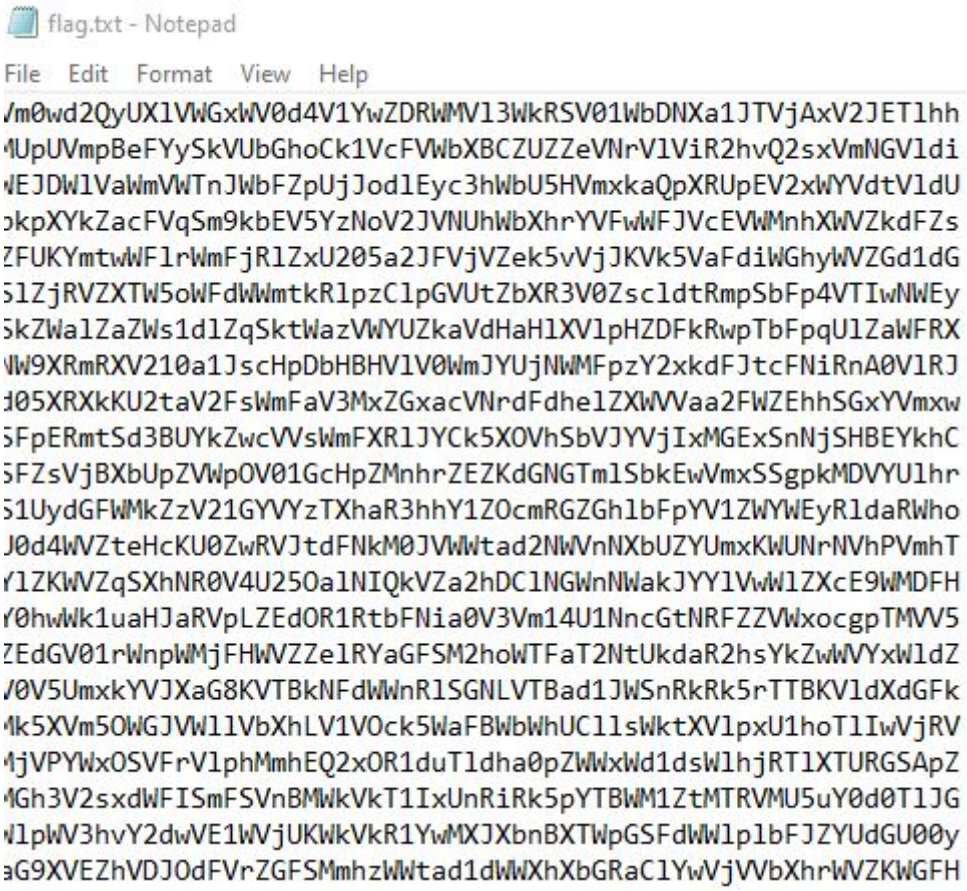
## Writeup for Cybersea CTF 2021

### Category: Cryptography - Matryoshka64 (150)

Given Text File

flag.txt

It in the file contains line of string with a certain fix length then a newline.




From the name of the question, Matryoshka is russian dolls, which is a doll contains another dolls inside, So a Nested dolls. Presumably, 64 in name is Base64, Therefore a nested Base64.

```
import base64

f = open("flag.txt", "r")
flines = f.readlines()

#read line by line and base64 decode
longstr = ""
for line in flines:
```

```
    strline = line.rstrip()
    strmat = base64.b64decode(strline)
    longstr = longstr + str(strmat, "utf-8") #combine them into long
string
    #as i notice that there is "\n" in the output of base64 decode

#after that keep repeating until flag is found
while(True):
    found = False
    splitline = longstr.split("\n") #split back by "\n" to be based64
decode
    longstr = ""
    #print(splitline)
    for x in splitline:
        if "flag" in x: #check if flag string contain in one of the line
            print(x)
            found = True
            break
        output = base64.b64decode(x)
        longstr = longstr + str(output, "utf-8")
        print(output)
    if found:
        break
```


## Category: Cryptography - Prime Number (50) (I don't remember the question name)

Find number of prime number between 0 to 1073741823.

I just use the website to count number of prime number given a range.

https://www.dcode.fr/prime-numbers-search

It is 54400028.

## Category: Miscellaneous - Cols (150)

Given text file:

text.txt

consist of line with fix length and column.

```
#00b5ec #00b5ec #00b5ec #01b6ed #01b6ed #01b6ed #00b7ed #00b7ed #00b8ee #00b8ee #00b8ee #00b8ee #00b8ee #00b8ee
#5cd3f3 #5ed4f5 #60d5f6 #60d4f6 #61d5f4 #63d6f5 #66d7f7 #66d4f5 #67d7f5 #68d7f5 #67d6f4 #67d5f4 #69d6f5 #68d5f4
#41734a #43834d #326834 #6ca071 #89c271 #a3dd94 #85c379 #6caf5c #629e62 #a7d6bd #88c789 #73b760 #94c984 #b6e5ad
#427044 #367c4b #277341 #307248 #41794a #498143 #1c5e2b #56a56e #89cc87 #71a95e #589655 #7baa6d #d3e6a0 #b9d3a8
#192126 #454949 #474344 #595755 #575c54 #100c09 #6c5e51 #8e8079 #3f312a #57463c #261c11 #17140a #271d13 #6a553d
#01b6ed #00b5ec #00b7ed #00b7ed #00b7ed #00b7ed #00b8ee #00b8ee #01b9ef #00b9ef #01baf0 #00b9ef #01baf0
#62d7f6 #62d4f4 #62d5f4 #63d4f4 #63d3f4 #66d6f5 #68d6f5 #69d8f6 #68d7f5 #6bd8f7 #6bd8f7 #6bd7f6 #6bd7f5 #6cd7f5
#4f8559 #88c485 #98ce8e #8ac681 #63ae55 #5aac5e #65a564 #91bf8b #51815a #98d0a3 #6ab364 #83c56f #90cf79 #7fc26d
#5b8557 #2d743d #247741 #266e46 #2c6642 #2c693f #34824f #53b273 #63b874 #579d58 #67b371 #4f9c5a #ade19e #9dce96
#080b10 #353735 #707477 #343a3e #1d1d1a #372c26 #6f554e #63504c #382a25 #52453b #4b3d31 #251b0c #291e0e #31200c
#00b7ed #00b7ed #00b8ee #00b8ee #00b8ee #01b9ee #00baed #00b9ed #00baed #00baed #00b9ed #01bbee #00baee #01bbef
#67d6f5 #65d5f3 #68d7f6 #68d5f5 #69d6f5 #6dd9f9 #6bd7f6 #6bd7f5 #6bd7f5 #6cd8f5 #6cd8f5 #6cd6f5 #6ed7f5 #70d8f5
#71aa6c #a4da98 #94cd88 #80c277 #76c272 #3d8834 #4a8639 #7caa71 #5e8f5d #8ac47c #84c86e #75bf63 #81c066 #8bc470
#89bb85 #2f773f #5ca76e #498a5b #296841 #297449 #27733d #62b46e #439b52 #5cb06d #8cda95 #5da75c #78bc6d #90d288
#091016 #5b5b5b #c4c6c5 #1b1b1c #453933 #5c4334 #5d4139 #3b2721 #231508 #302811 #3b2e20 #44332a #3a2f28 #40302a
#00b7ed #00b8ee #00b8ee #00b8ed #00b9ed #00b9ed #00baed #00b9ec #00baed #01bbee #00bbee #01bcef #01bcef #01bcef
#69d6f4 #6ad7f5 #6bd6f5 #6bd5f4 #6cd6f5 #6fd8f8 #6fd8f5 #70d9f6 #6dd7f4 #6fd8f5 #70d8f6 #70d9f6 #73d9f7 #73d9f5
#8ecb7f #639b52 #7bb066 #98cd98 #3d8b4f #55aa5c #65b664 #6cb16d #8dc68b #7eb76f #99d27a #79bf69 #7db45c #97c36a
#91d094 #60a567 #4f8d49 #97d68d #5e9c59 #468d44 #5b9b50 #97d481 #4d9b4f #3e995b #59aa69 #77c072 #71be64 #78c270
#383d3f #313635 #717471 #0c0e09 #594f49 #81695e #674d44 #47332c #544633 #4f4733 #3f372d #8f827e #a09593 #baaaaa
```

Presumably the code means Color Hex Code since it has fix width and length which make a rectangle therefore we could generate image using the code by creating a pixel image.

Using Python image library (Pillow).

```
pip install Pillow
pip install numpy
```

```
from PIL import Image, ImageColor
import numpy as np

f = open("text.txt", "r")
flines = f.readlines()
count = 0
pixels = []
for line in flines:
    oneline = line.rstrip().split(" ")
    print("Width: ", len(oneline))
    count+=1
    pixellines = []
    for x in oneline:
        # Convert Hex Code to RGB array (XX, XX, XX)
        pixellines.append(ImageColor.getcolor(x, "RGB"))
```

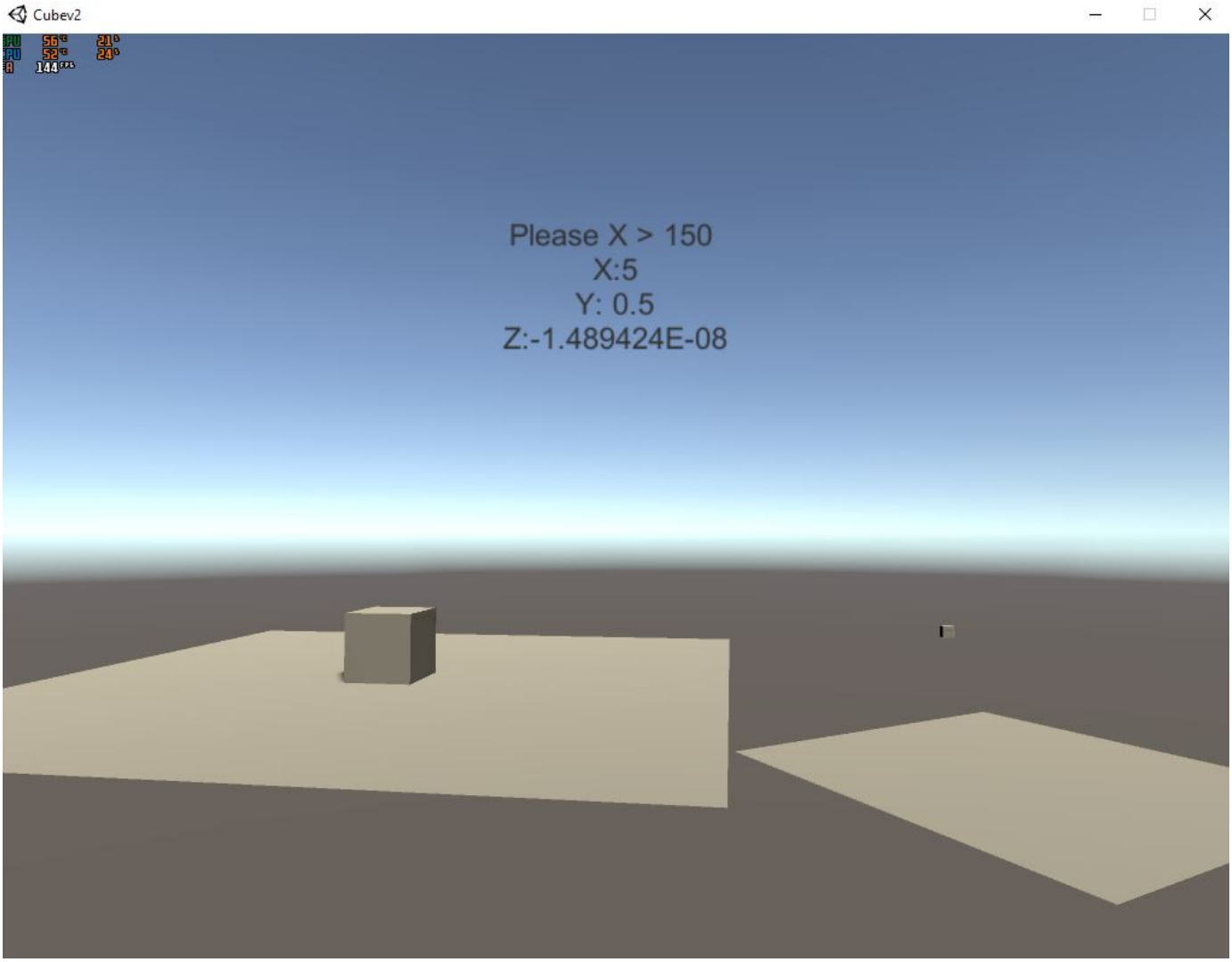
```
    pixels.append(pixellines)
print("Heigth: ", count)

print(pixels)
array = np.array(pixels, dtype=np.uint8)

new_image = Image.fromarray(array)
new_image.save('new.png')
```


## Category: Miscellaneous - Cube (150) (I don't remember the question name)

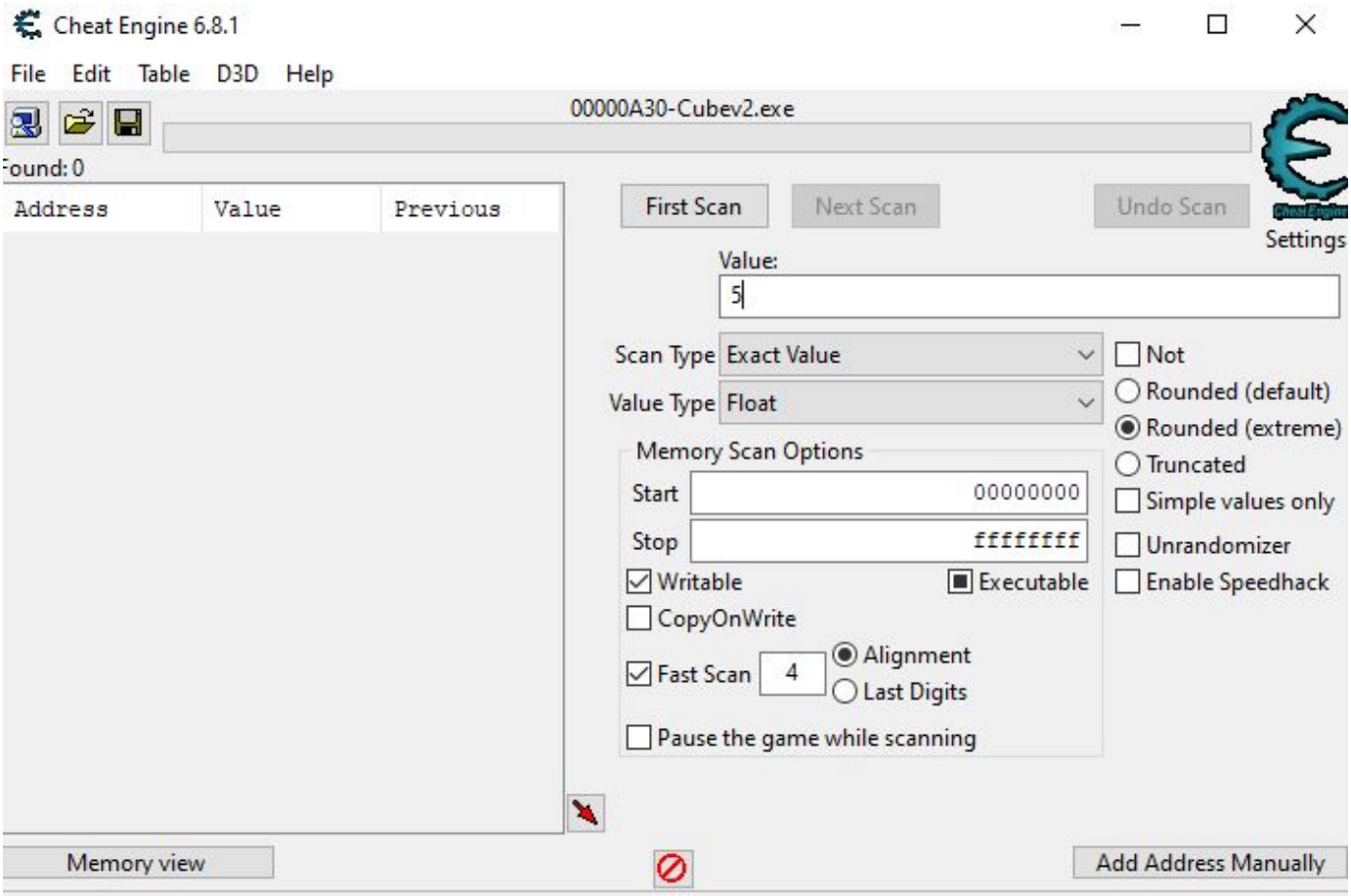
Given zip file containing an application made in Unity.

Instruction Left, Up and Down arrow for movement. Space to reset.

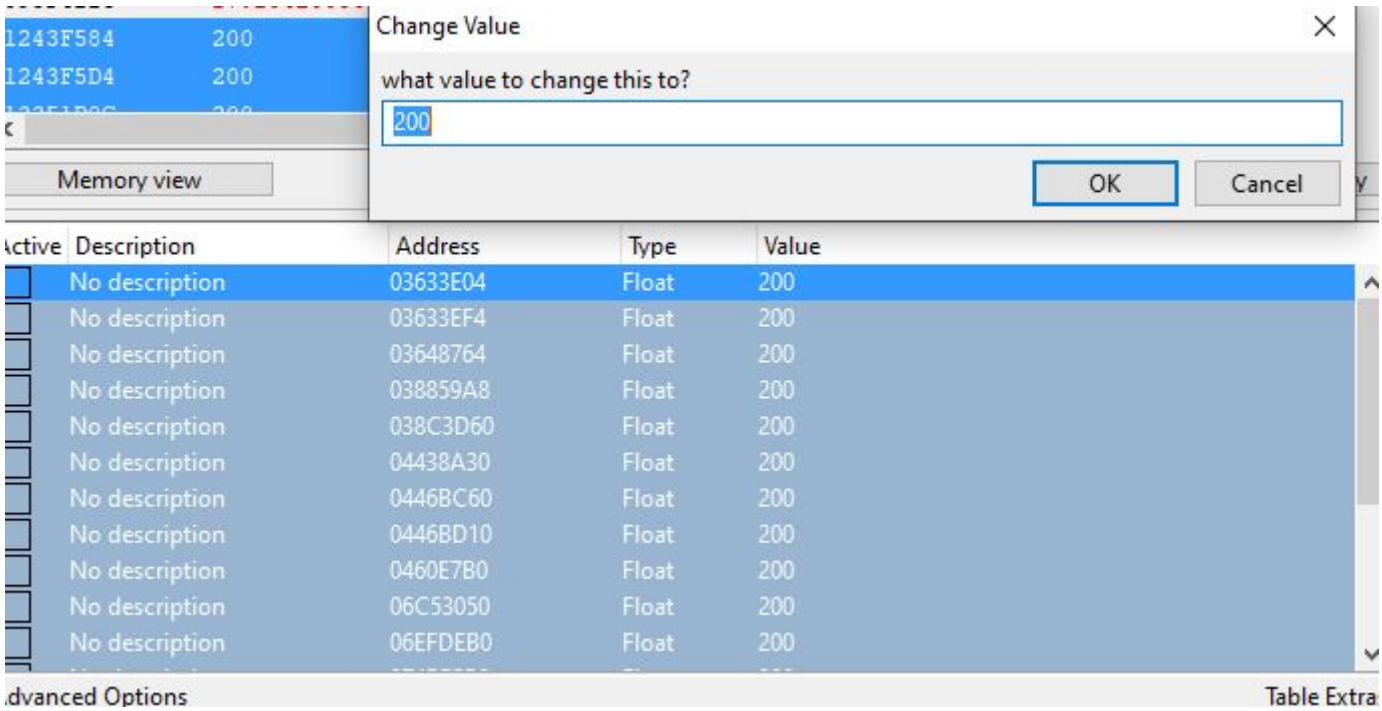
No2_Cubev2_v1_exe.zip



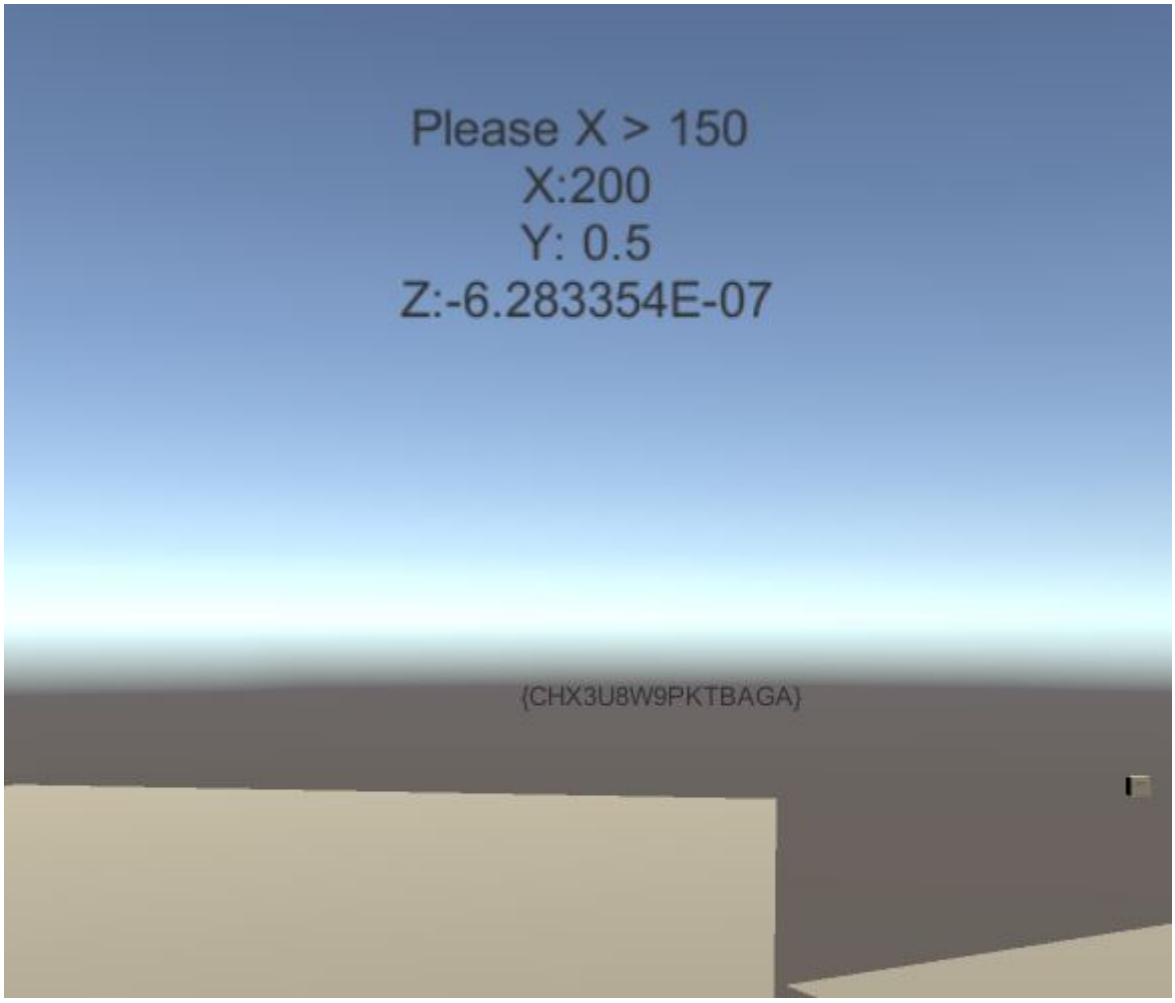

In the game we can only move the box to left, up and down. It seems it want X > 150 and X corresponding to left and right movement but the right is not implemented.




By using cheat engine we can filter and find which address corresponding to X value in memory.




After filtering some more value using next scan, we found a few address responsible for X value. Then we change the value > 150.




Flag is shown after X > 150.

---

## Category: Miscellaneous - Structure (150)

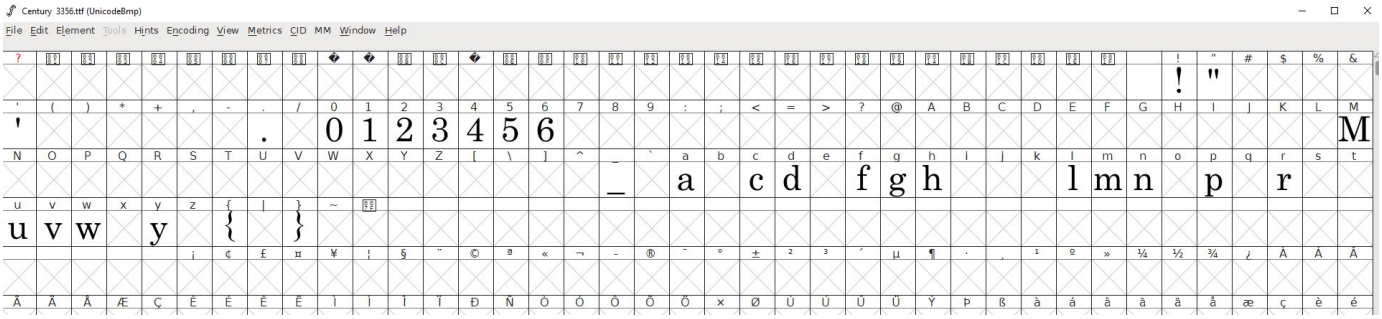
---

Given PDF file:

[structure.pdf]

Searching in HxD Hex Editor and stegoveritas find nothing interesting except for Font File.

Opening it in FontForge reveal certain character missing from the font.




Therefore, the PDF might have been compressed.

Using qpdf we will uncompress the pdf

```
sudo ./bin/qpdf --stream-data=uncompress structure.pdf  uncompress.pdf
```

Open the uncompress.pdf in HxD Hex editor found the flag.

```
0000B50  36 20 30 20 52 20 2F 53 20 2F 53 70 61 6E 20 2F  6 0 R /S /Span /
0000B60  54 79 70 65 20 2F 53 74 72 75 63 74 45 6C 65 6D  Type /StructElem
0000B70  20 3E 3E 0A 3C 3C 20 2F 4B 20 5B 20 33 31 20 30   >>.<< /K [ 31 0
0000B80  20 52 20 33 33 20 30 20 52 20 5D 20 2F 50 20 38   R 33 0 R ] /P 8
0000B90  20 30 20 52 20 2F 50 67 20 33 37 20 30 20 52 20   0 R /Pg 37 0 R
0000BA0  2F 53 20 2F 50 20 2F 54 79 70 65 20 2F 53 74 72  /S /P /Type /Str
0000BB0  75 63 74 45 6C 65 6D 20 3E 3E 0A 3C 3C 20 2F 41  uctElem >>.<< /A
0000BC0  63 74 75 61 6C 54 65 78 74 20 28 66 6C 61 67 7B  ctualText (flag{
0000BD0  34 77 33 35 30 6D 33 21 5F 79 30 75 27 76 33 5F  4w350m3!_y0u'v3_
0000BE0  64 31 35 63 30 76 33 72 33 64 5F 34 5F 68 31 64  dl5c0v3r3d_4_hld
0000BF0  64 33 6E 5F 70 34 36 33 2E 7D 29 20 2F 4B 20 5B  d3n_p463.}) /K [
0000C00  20 30 20 5D 20 2F 50 20 33 30 20 30 20 52 20 2F   0 ] /P 30 0 R /
0000C10  50 67 20 33 37 20 30 20 52 20 2F 53 20 2F 53 70  Pg 37 0 R /S /Sp
0000C20  61 6E 20 2F 54 79 70 65 20 2F 53 74 72 75 63 74  an /Type /Struct
0000C30  45 6C 65 6D 20 3E 3E 0A 5B 20 33 31 20 30 20 52  Elem >>.[ 31 0 R
0000C40  20 33 33 20 30 20 52 20 33 35 20 30 20 52 20 5D   33 0 R 35 0 R ]
```