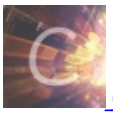
WeChall CTF Writeup（九）

原创

lmn_ 于 2022-02-28 10:00:00 发布 30 收藏

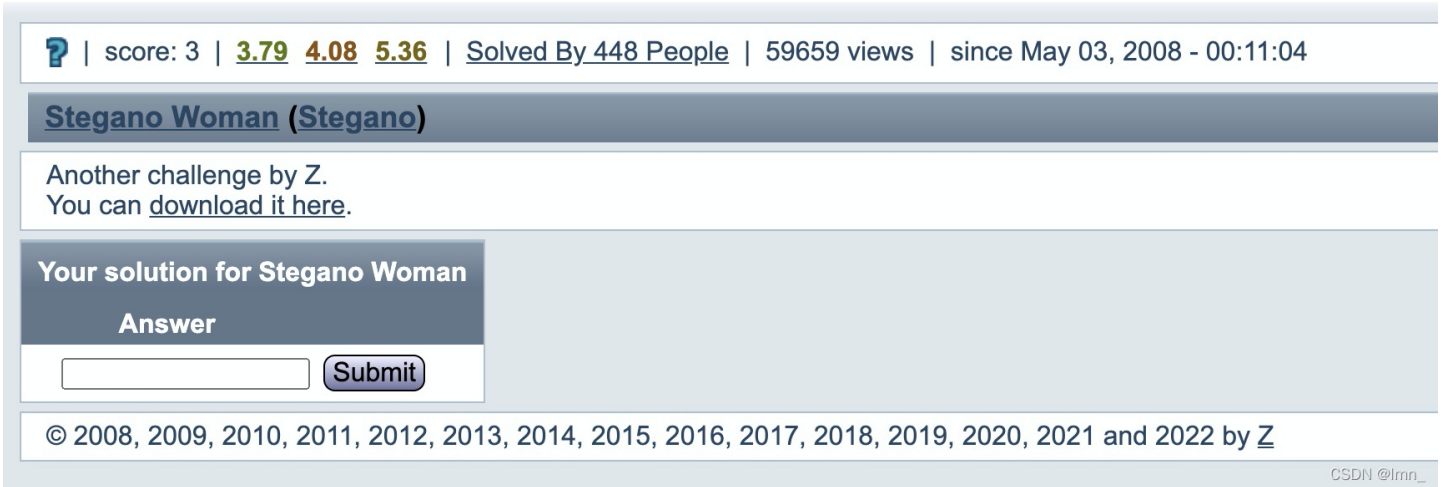
分类专栏： CTF 文章标签： CTF



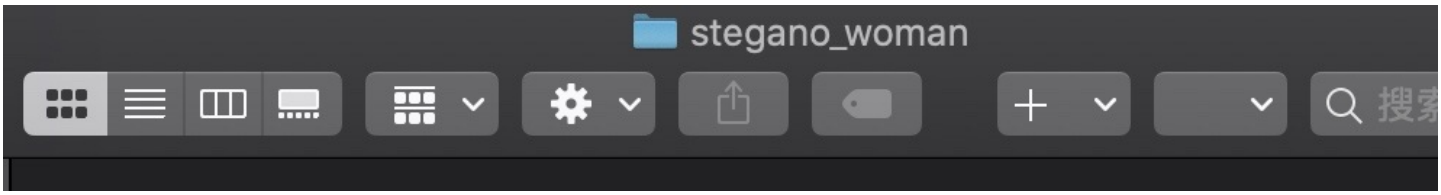
文章目录

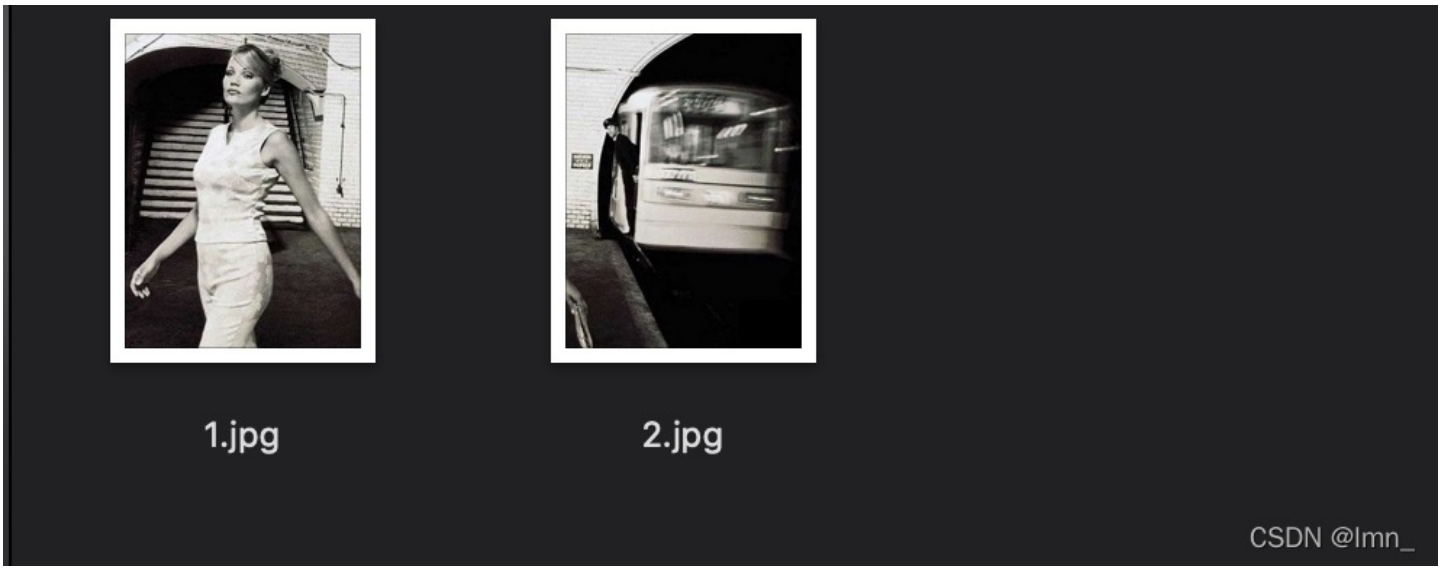
以下题目标题组成：
[Score] [Title] [Author]

0x13 3 Stegano Woman by Z

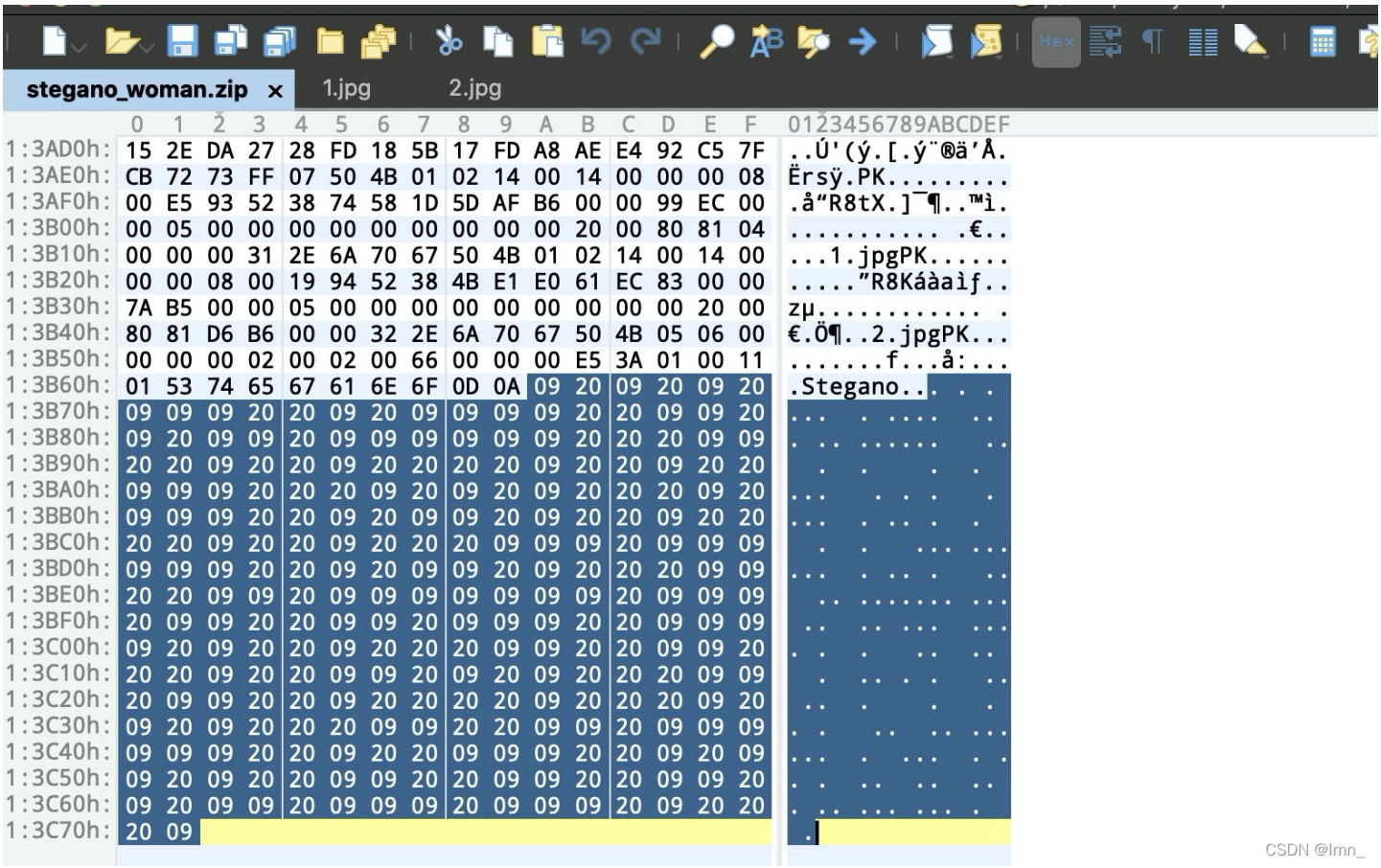


题目意思：
Z 的另一个挑战。
您可以在此处下载。
下载下来的图片：





这道题在图片没有找到什么有用信息，但是在zip里找到这么一段由09 20两种表达的一段字符

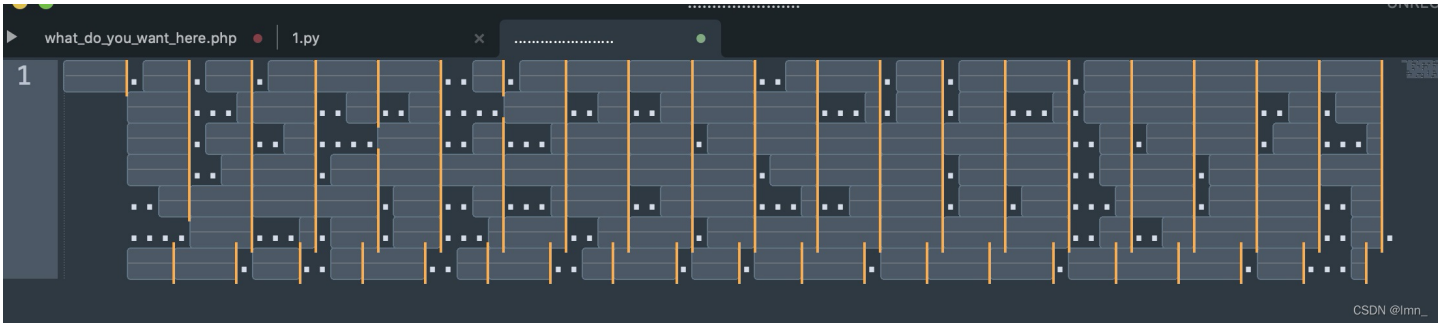


仔细思考一下两种不同的可以考虑什么编码

09 tab

20 space

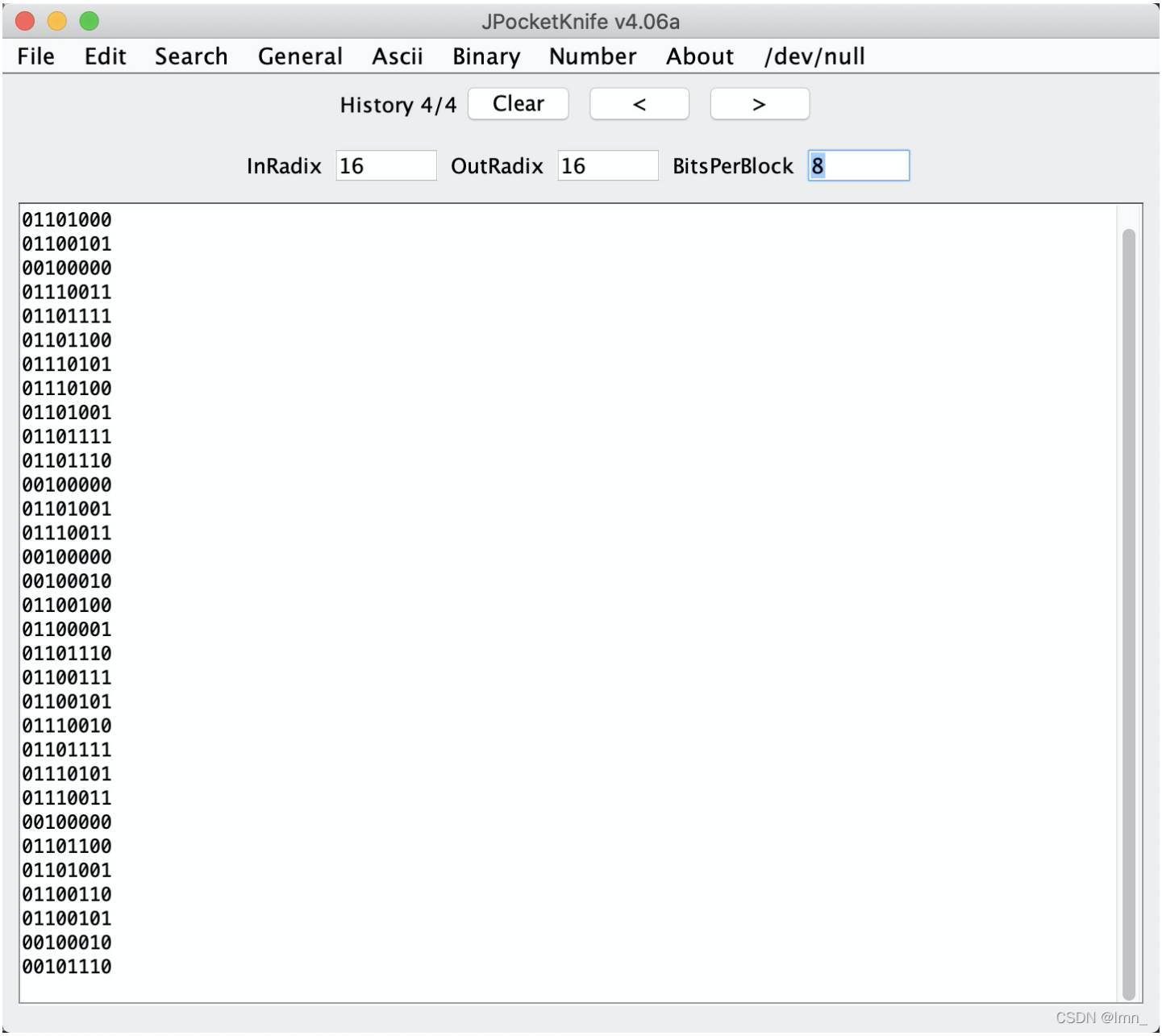
尝试一下摩斯密码



可是怎么分割呢？

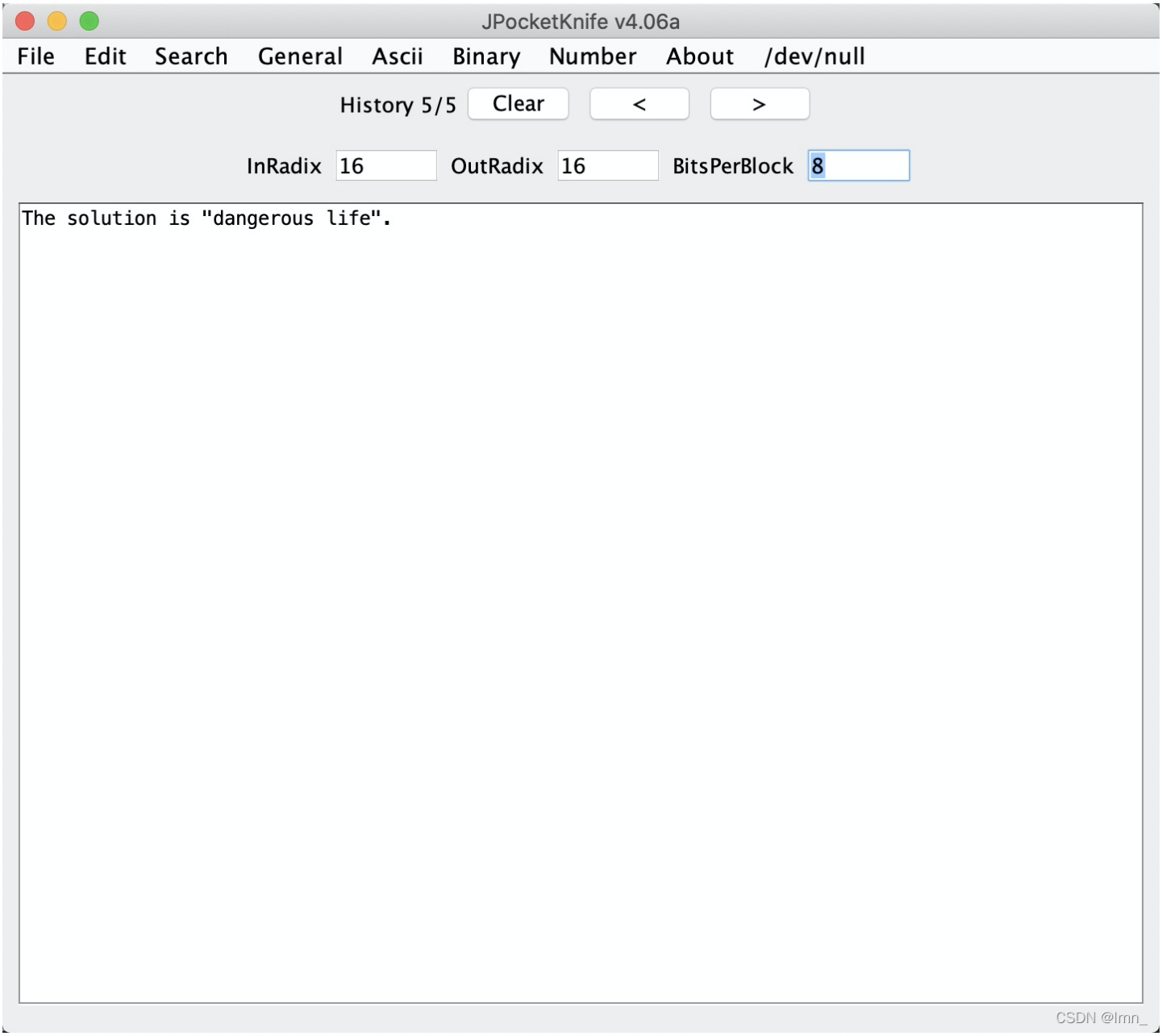
想象一下09是0，20是1

还记得之前有道题提供的工具么



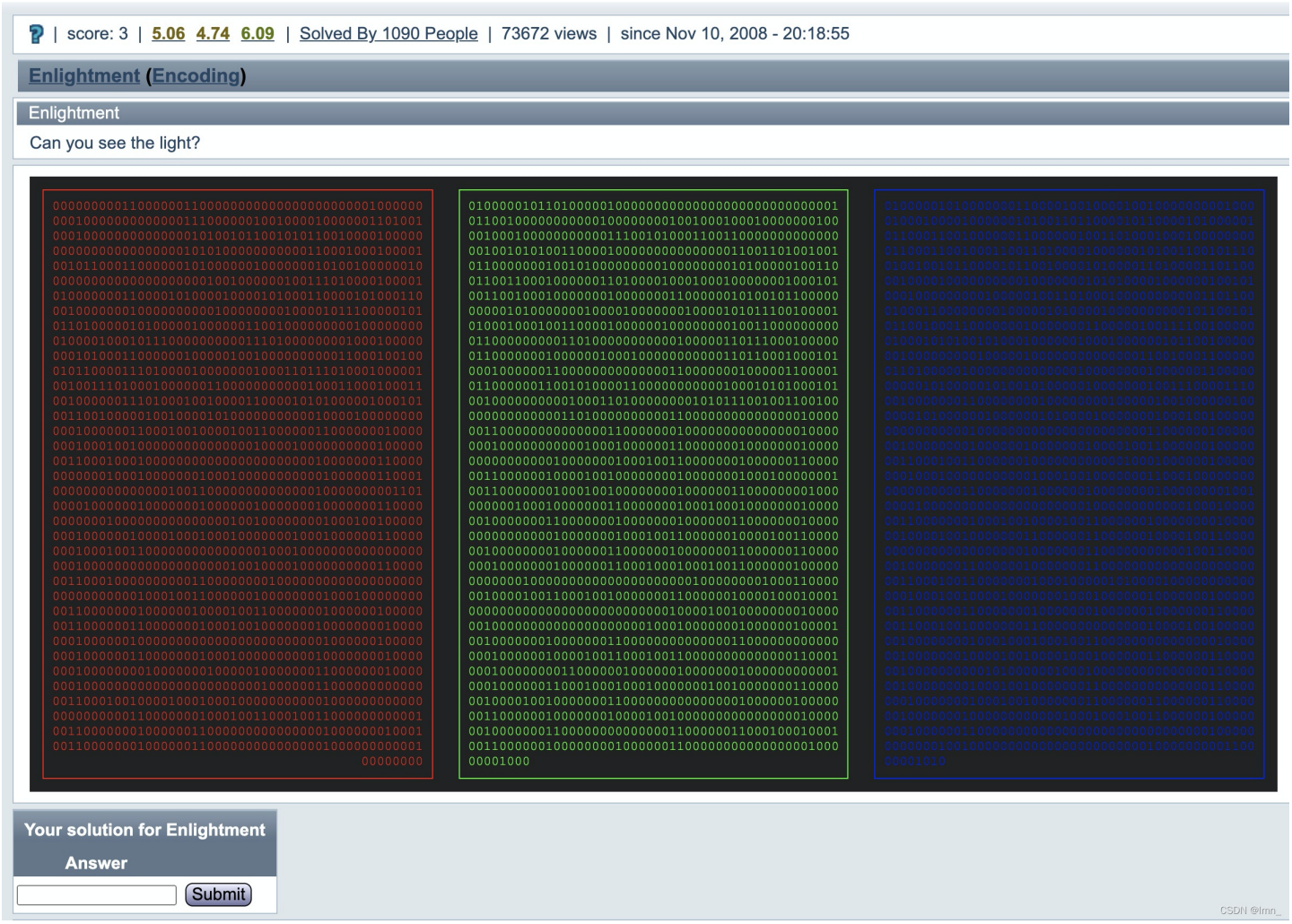
正好可以format成8位为一组

解出结果

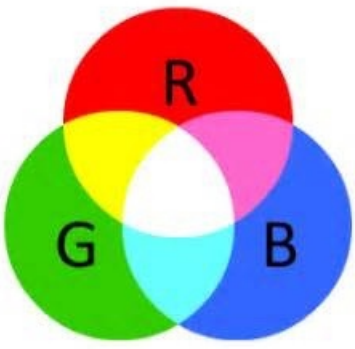


The solution is “dangerous life”.

0x14 3 Enlightment by anto



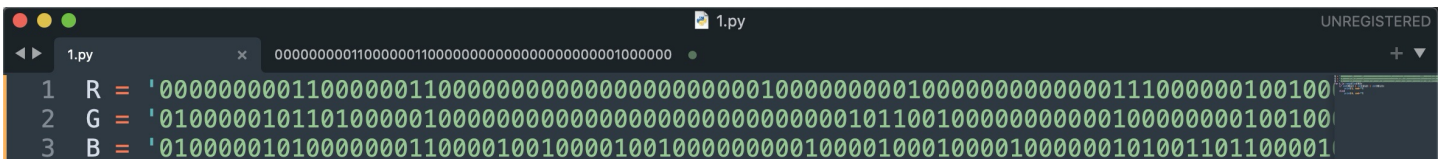
看到光了么？



既然是看到光，那就是三个颜色叠加

代码：

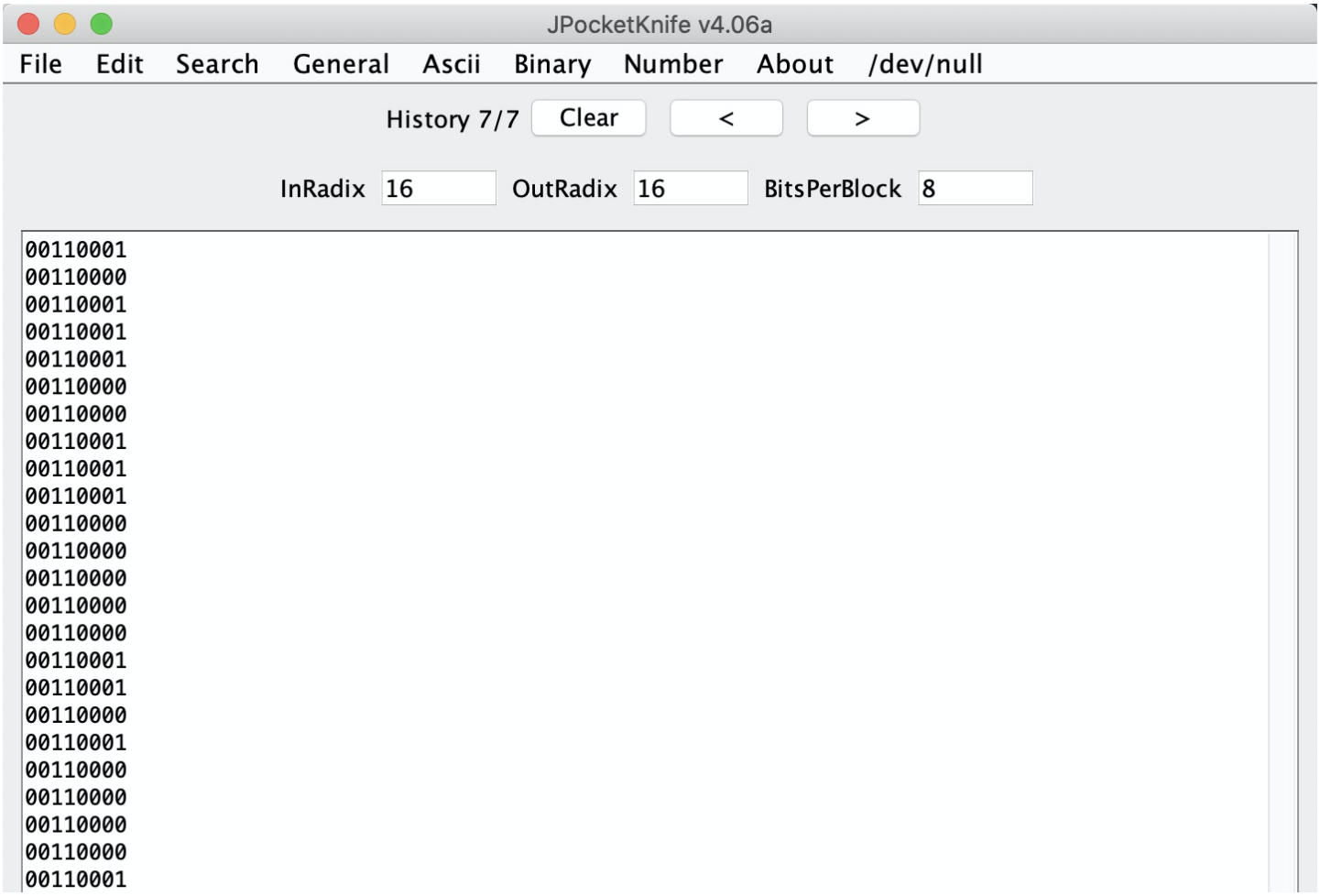
```
R = '0000000001100000011000000000000000000000010000000001000000000000011100000010010000100000011010010001000000
0000000101001011001010110010000100000000000000000000001010100000000000110001000100001001011000110000001010000001
0000000101001000000100000000000000000000100100000010011101000010000101000000011000010100001000010100011000010100
0011000100000001000000000010000000010000101110000010101101000001010000010000001100100000000010000000001000010001
0111000000000011101000000001000100000001010001100000010000010010000000000110001001000101100001110100001000000001
0001101110100010000010010011101000100000011000000000001000110001000110010000011101000100100001100001010100000010
0010100110010000001001000010100000000000100001000000000001000000110001001000010011000001100000001000000010001001
0000000000000000100001000000000010000000110001000100000000000000000000010000000110000000000010001000000010001000
0000000010000001100010000000000000010011000000000000010000000001101000010000001000000010000001000000010000000100000001
1000000000001000000000000000010010000000010001001000000001000000100001000100010000000100010000001100000001000100
1000000000000000100010000000000000000000100000000000000000010010000100000000011000000110001000000000011000000
0100000000000000000000000000000100010011000000100000000100010000000001100000001000000100001001100000001000000
0000000110000001100000001000100100000001000000010000001000000100000000000000000000010000001000000001000001
1000000010001000000000010000000100000001000000010000000100000010000001100000001000000100000000000000000000
1000000110000000000000011000100100001000100010000000000010000000000000000000001100000001000100110001001100000000
0000100110000000100000011000000000000001000000010001001100000001000000110000000000000010000000000010000000'
G = '0100000101101000001000000000000000000000000000010110010000000000010000000100100010001000000010000100010000
0000000111001010001100110000000000000001001010100110000100000000000000011001101001001011000000010010100000000010
0000000101000001001100110011000100000011010000100010001000000010001010011001000100000001000000011000000101001011
0000000000010100000001000010000000100001010111001000010100010001001100001000000100000000100110000000000110000000
0110100000000000100000110111000100000011000000010000001000100000000000110110001000101000100000011000000000000001
1000000010000011000010110000001100101000011000000000001000101010001010010000000001000111010000000010101110010011
0010000000000000011010000000000110000000000000001000000110000000000000011000000010000000000000010000000100000000
0000100010000001100000001000000010000000000000001000000010001001100000001000000110000001100000010000100100000000
1000000010001000000010011000000010001001000000001000000110000000010000000001000100000001100000001000100010000000
1000000100000001100000001000000010000001100000001000000000000001000000010001001100000001000001001100000010000000
1000000110000001000000011000000110000001000000010000001100010001000100110000001000000000000100000000000000000000
0010000000010001100000010000100110001001000000011000000100001000100010000000000000000000000010000100100000000
1000000100000000000000000001000100000001000000100001001000000010000000110000000000000011000000000000001000000001
0000100110001001100000000000000011000100010000000011000000100000010000000100000000010001000000110001000100010000
0000100100000001100000010000100100000001100000000000000100000010000001100000010000000100001001000000000000000000
1000000100000001100000000000000110000001100010001000100110000001000000010000001100000000000000010000000001000'
B = '0100000101000000011000010010000100100000000010000100010000100000010100110110000101100001010000010110001100 1
0000001100000010011010001000100000000011000110010001100110100001000000101001100101110010010010110000101100100001
0100001101000011011000010000100000000001000000010101000010000001001010001000000001000001001110100010000000000011
0110001000110000000010000010100001000000000010110010101100100011000000010000000110000100111100100000010001010100
0101000100000010001000000101100100000001000000000100000100000000000000110010001100000011010000010000000000000010
0000000100000011000000000001010000010100101000001000000010011100001110001000000011000000010000000100000010010000
0010000001010000001000000101000010000000100010010000000000000000100000000000000000000011000001000000010000000000
1000000100000001000010011000000100000001100010011000000100000000000010001000001000000001000100000000010001001
0000000110001000000000000000000110000001000000100000001000000010010000100000000000000000000010000000000001000
1000000110000000100010010000100110000001000000001000000100001001000000011000000110000001000010011000000000000000
0000000100000001100000000000100111000000100000011000000100000011000000000000000000000011000100111000000010001000
0010100001000000000000010001001000010000000100010000001000000010000000110000001100000001000000010000001000000001
1000000110001001000000011000000000000000100001001000000010000000010001000100010011000000000000000100000010000001
0000100100001000100000011000000110000001000000000001010000001000100000000000000110000001000000001000100100000001
1000000000000001100000001000000010001001000000011000000110000011000000100000001000000000000100010001001100000001
0000000010000001100000000000000000000000000000010000000000001001000000000000000000000010000000011000000101 0'
for i in range(len(R)):
    if int(R[i]) | int(G[i]) | int(B[i]):
        print(1, end="")
    else:
        print(0, end="")
```



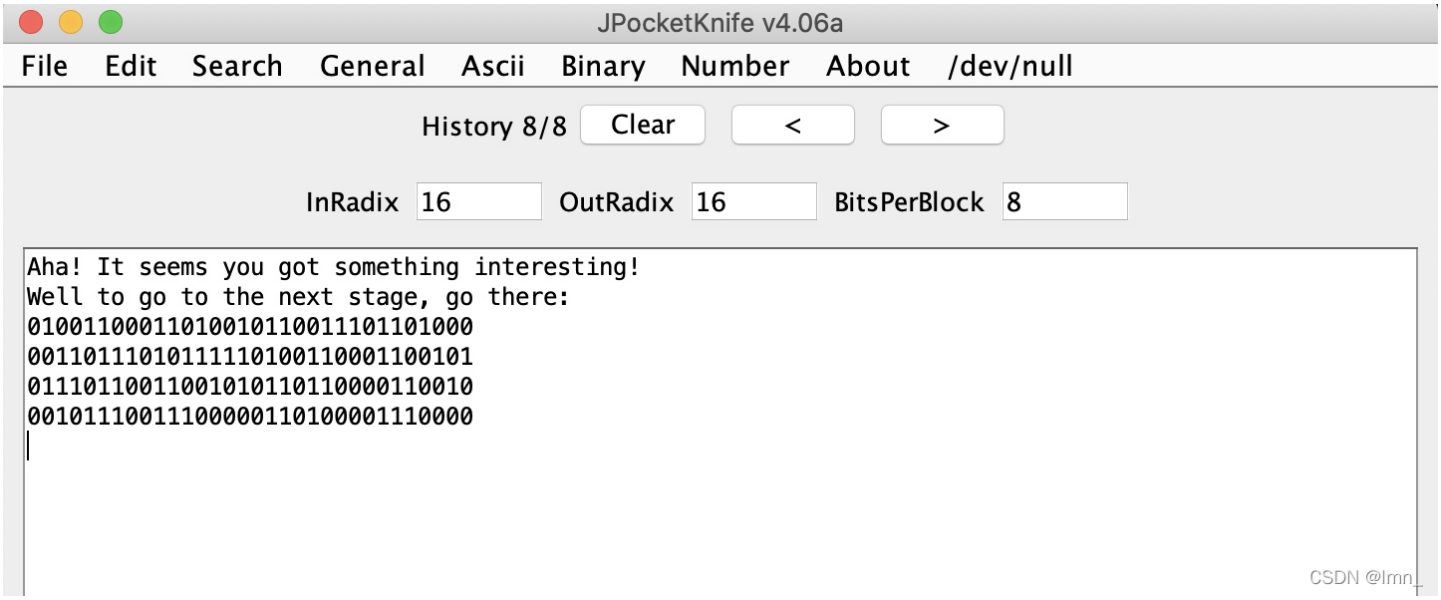
```python
for i in range(len(R)):
    if int(R[i]) | int(G[i]) | int(B[i]):
        print(1, end="")
    else:
        print(0, end="")
```

```
0100000101101000011000010010000100100000010010010111010000100000011100110110010101100101011011010
1110011001000000111100101101111011101010010000001100111011011110111010000100000011100110110111101
1011010110010101110100011010000110100101101110011001110010000001101001011011100111010001100101011
1001001100101011100110111010001101001011011100110011100100000100001101000010100101011101100101011
1100011011000010000001110100011011110010000001100111011011110010000001110100011011110010000001110
1000110100001100101001000000110111001100101011110000111010000100000011100110111010001100001011001
1101100101001011000010000001100111011011110010000001110100011010000110010101110010011001010011101
0000011010000101000110000001100010011000000110000001100010011000100110000001100000011000000110001
0011000100110000001100010011000000110000001100010011000000110001001100010011000000110000001100010
0110001001100010011000000110001001100010011000000110001001100000011000000110000000011010000101000
1100000011000000110001001100010011000000110001001100010011000100110000001100010011000000110001001
1000100110001001100010011000100110000001100010011000000110000001100010011000100110000001100000011
0000001100010011000100110000001100000011000100110000001100010000110100001010001100000011000100110
0010011000100110000001100010011000100110000001100000011000100110001001100000011000000110001001100
0000110001001100000011000100110001001100000011000100110001001100000011000000110000001100000011000
1001100010011000000110000001100010011000000001101000010100011000000110000001100010011000000110001
0011000100110001001100000011000000110001001100010011000100110000001100000011000000110000001100000
0110001001100010011000000110001001100000011000000110000001100000011000100110001001100010011000000
1100000011000000110000000011010000101 0[Finished in 64ms]
```

再一次放到那个软件中

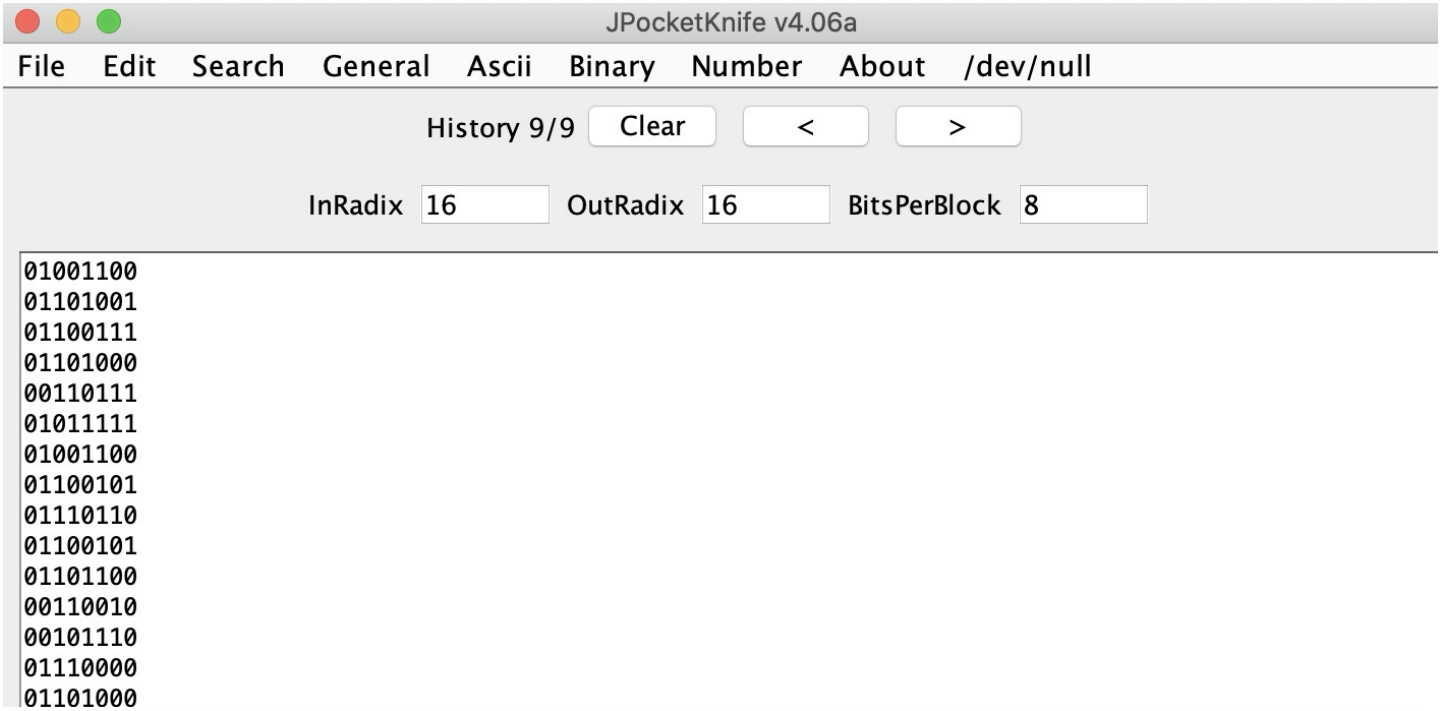


```
00110001
00110001
00110000
00110000
00110000
00110000
00001101
00001010
```

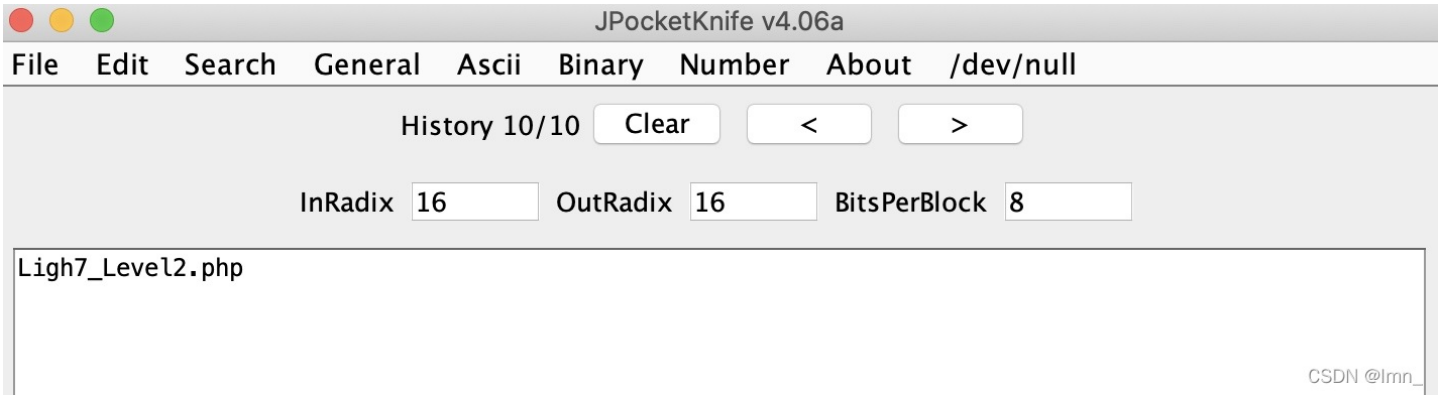

```
Aha! It seems you got something interesting!
Well to go to the next stage, go there:
01001100011010010110011101101000
00110111010111110100110001100101
01110110011001010110110000110010
00101110011100000110100001110000
```

again!

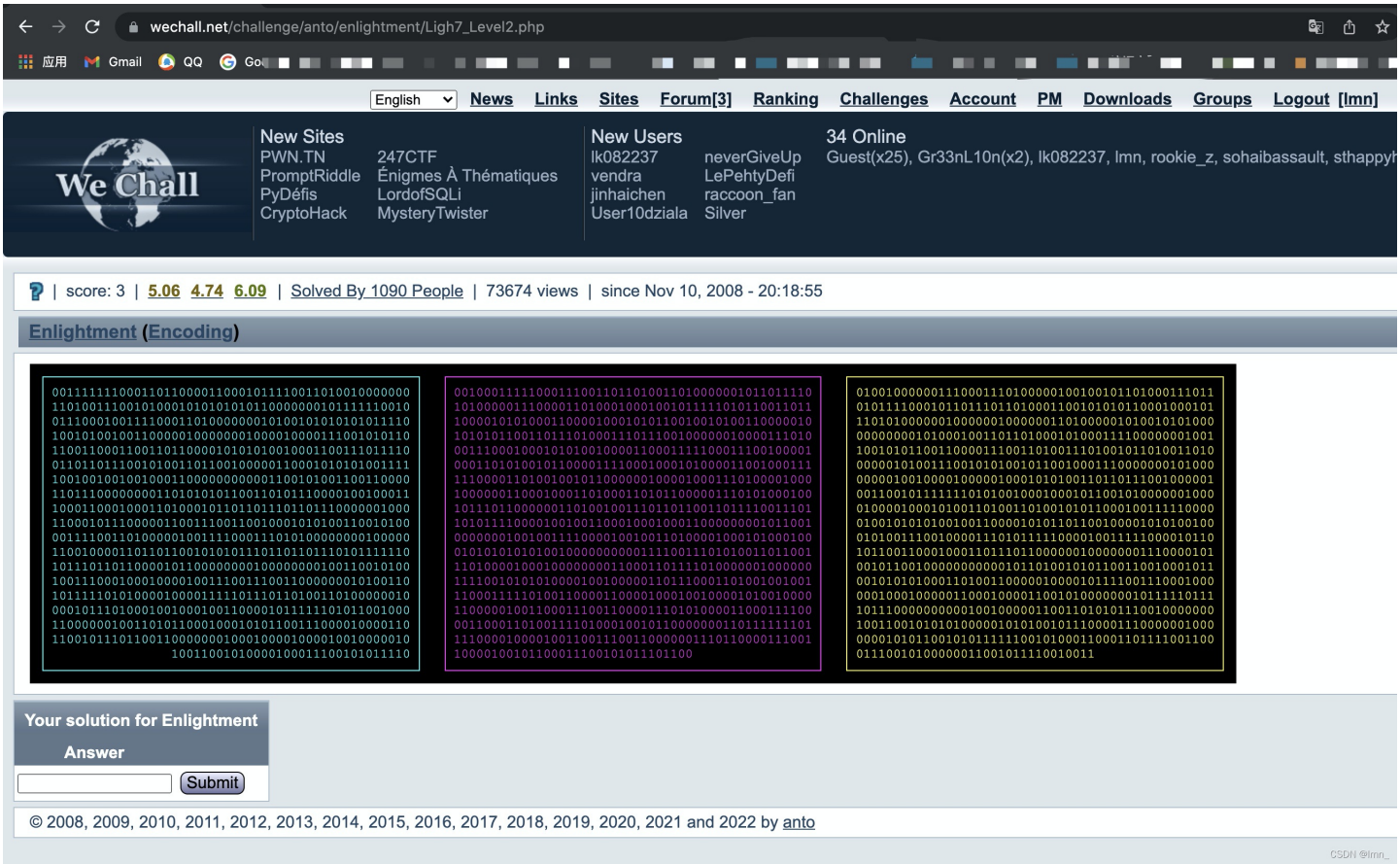


```
01110000
```

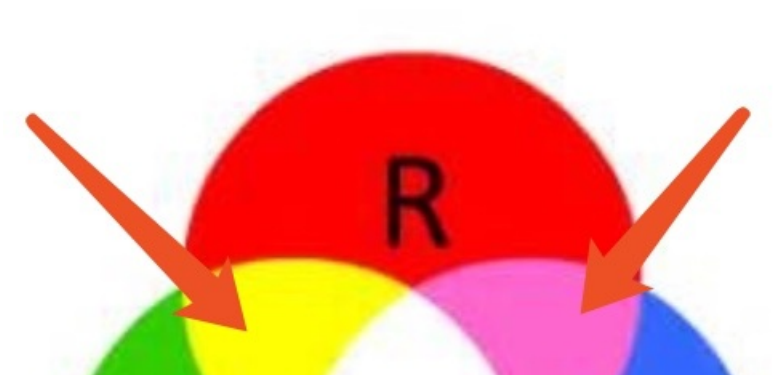

快看到答案了

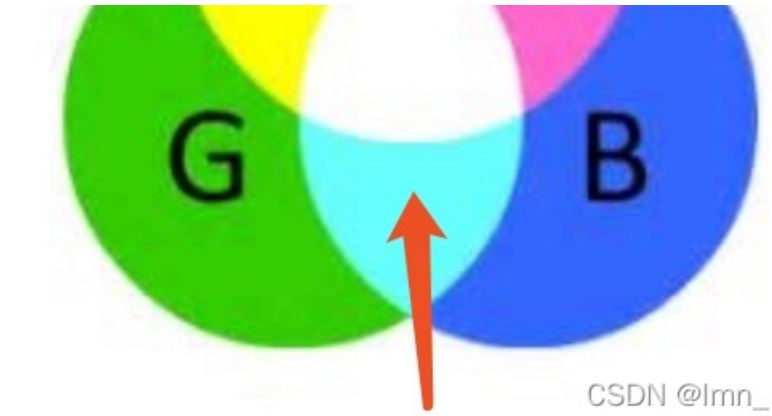


这就很有意思了



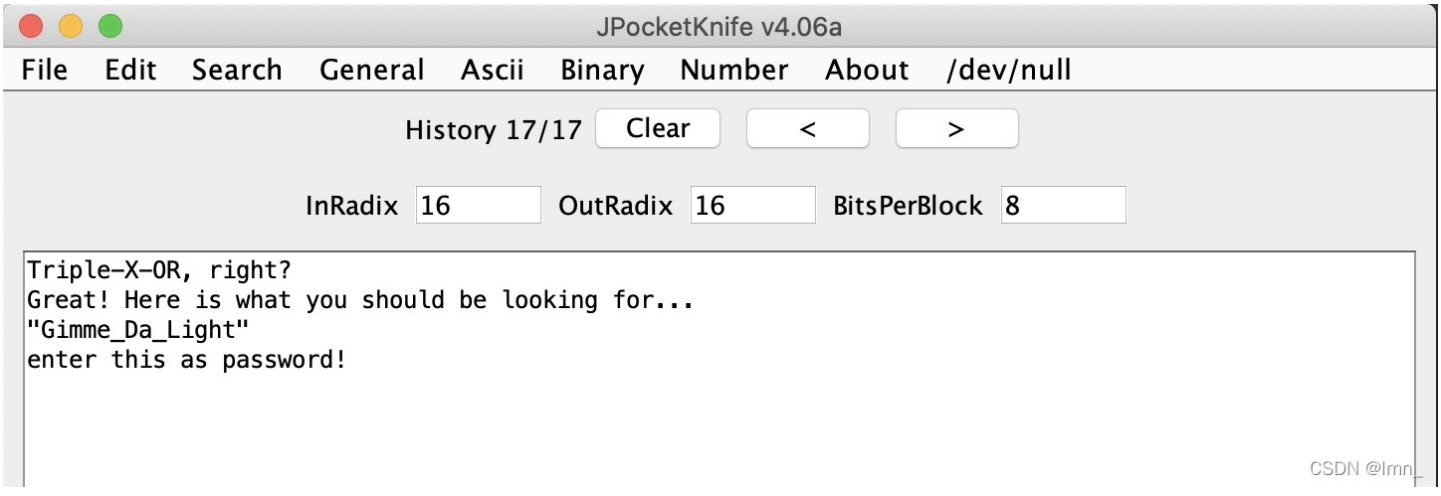
是这个颜色对吧






```python
R = '00100011111000111001101101001101000000101101111010100000111000011010001000100101
G = '01001000000111000111010000010010010110100011101101011110001011011101101000110010
B = '00111111100011011000011000101111001101001000000011010011100101000101010101011000
for i in range(len(R)):
    if int(R[i]) ^ int(G[i]) ^ int(B[i]):
        print(1, end="")
    else:
        print(0, end="")
```

```
0101010001110010011010010111000001101100011001010010110101011000001011010100111101010010001011000
0100000011100100110100101100111011010000111010000111111000011010000101001000111011100100110010101
1000010111010000100001001000000100100001100101011100100110010100100000011010010111001100100000011
1011101101000011000010111010000100000011110010110111101110101001000000111001101101000011011110111
0101011011000110010000100000011000100110010100100000011011000110111101101111011010110100101101
1100110011100100000011001100110111101110010001011100010111000101110000011010000101000100010010001
1101101001011011010110110101100101010111110100010001100001010111110100110001101001011001110110100
0011101000010001000001101000010100110010101101110011101000110010101110010001000000111010001101000
0110100101110011001000000110000101110011001000000111000001100001011100110111001101110111011011110
1110010011001000010000[Finished in 43ms]
```



Triple-X-OR, right?

Great! Here is what you should be looking for…

“Gimme_Da_Light”

enter this as password!

0x15 3 Simply Red by anto

| score: 3 | 5.15 5.45 6.48 | Solved By 311 People | 59285 views | since Nov 10, 2008 - 20:19:00

Simply Red (Stegano, Image)

Find the sentence hidden in me or I'll have to destroy you.

"No sacrifice is too great in the service of freedom."

Your solution for Simply Red

Answer

Submit

题目意思：

找出隐藏在我身上的那句话，否则我将不得不摧毁你。

“在为自由服务的过程中，任何牺牲都不算大。”

既然从根本开始做，那就假设我不知道这是谁，既然是饮用那就是他说出的话，搜索一下

Animated character biography

Optimus Prime (formerly known as **Orion Pax**) is constantly, if not always, depicted as having strong moral character, excellent leadership, and sound decision-making skills, and possesses brilliant military tactics, powerful martial arts, and advanced extraterrestrial weaponry. Optimus Prime has a strong sense of honor and justice, being dedicated to building peaceful and mutually beneficial co-existence with humans, the protection of life and liberty of all sentient species.[3] As the current Matrix of Leadership bearer, Optimus Prime is the de facto leader of the Autobots, a faction of a transforming species of synthetic intelligence from the planet Cybertron. The Autobots are constantly waging civil war against a rival faction of transforming robots called Decepticons. According to Bob Budiansky, co-writer of the Transformers series, Dennis O'Neil was responsible for his name.

Optimus Prime is usually depicted as being a member of an ancient Transformers race called the Dynasty of Primes, often receiving the title "The Last Prime" in many stories, in which he is depicted as being the last of the Primes. In the *Transformers: Covenant of Primus*, it was established that Optimus Prime was the last born of the original Thirteen Transformers created by the creator Primus. It was his unique spark and his inspiring reassurance that "All are One" that allowed the Primes to rally and succeed in their battle against the Chaos Bringer Unicron. When tragedy at last ended the era of the Primes and brought forth the new race of lesser descendant Transformers he alone chose to be reborn in the Well of All Sparks as one of them, that he might know them and their needs more completely. All memory of his past life gone, he took the name "Orion Pax" and sought his way like any other robot on the new world becoming Optimus Prime once more when receiving the Matrix of Leadership when Cybertron faced a new enemy in his former friend, Megatron and his army of followers, the Decepticons. This brings a Great War to their planet of Cybertron. Optimus's origins and personality can vary depending on which "universe" he's seen in. This origin is the most consistent between the various incarnations. Further differences are listed in the respective sections below.

Generation 1

In-universe information	
Affiliation	Autobot/Maximal
Japanese name	Inochi, Convoy
Sub-group	Action Masters, Autorollers, Combat Heroes, Deluxe Vehicles, Go-Bots, Masterpiece, Primes, Powermasters, Voyagers
Function	Autobot Leader, Supreme Commander, Chief Commander
Rank	10
Partner	Roller, Hi-Q
Motto	"Freedom is the right of all sentient beings." (Generation 1) "No sacrifice is too great in the service of freedom." (Classics)
Alternate modes	Freightliner Cab-over-engine Class 8 truck, Cybertron truck, COBRA Sentry & Missile System tank, 1920s style truck, Peterbilt Truck, Mid-90s Peterbilt 4964EX Tanker Truck, Lamborghini Diablo, Dump truck, Dodge Ram SRT-10, Nissan GT-R, Bat

CSDN @lmn_

我们知道原来它叫 Optimus Prime

第一眼竟然认成了 Option Prime

分析一定与素数有关，并且是红色为底色，之前有一道题是提图片的HSB，既然是红色，那就分析像素的R为素数尝试一下

在搜索代码的时候发现已经有大佬写出了这个代码
地址：https://www.tuziang.com/index.php/combat/1985.html

```
from PIL import Image
from math import sqrt

def is_prime(n):
    if n==1:
        return False
    for i in range(2, int(sqrt(n) + 1)):
        if n % i == 0:
            return False
    return True
img = Image.open("op.png")
width = img.size[0]
height = img.size[1]

for x in range(width):
    for y in range(height):
        r,g,b = img.getpixel((x,y))
        if is_prime(r):
            continue
        else:
            img.putpixel((x,y),(255,255,255))

img.show()
```

In [1]:

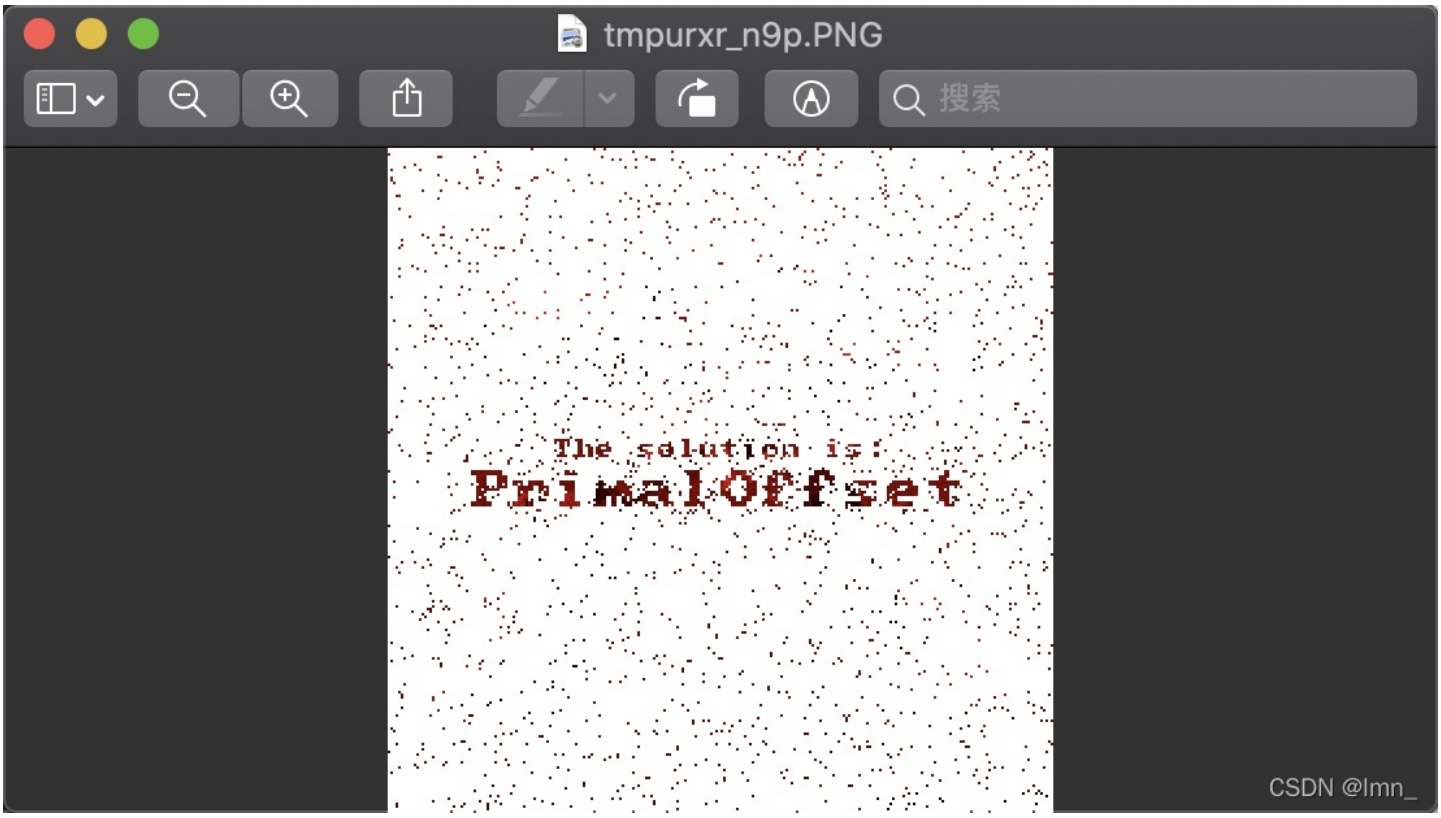
```
from PIL import Image
from math import sqrt

def is_prime(n):
    if n==1:
        return False
    for i in range(2, int(sqrt(n) + 1)):
        if n % i == 0:
            return False
    return True

img = Image.open("op.png")
width = img.size[0]
height = img.size[1]

for x in range(width):
    for y in range(height):
        r,g,b = img.getpixel((x,y))
        if is_prime(r):
            continue
        else:
            img.putpixel((x,y),(255,255,255))

img.show()
```



参考链接：

http://www.chiange.com/

http://www.ruanyifeng.com/blog/2013/07/gpg.html

https://www.tuziang.com/index.php/combat/1855.html