

第七周周报

学习了brainfuck的解码

看了御网杯2024和XCTF中都有，但是用在线编辑器弄不出来，看了出题人的wp，弄懂了编码原理

```
>+++++++++++[<+++++>--++--]<[-]  
>+++++++++++[<+++++>--++--]<[-]
```

前面的加号数量代表x的值，后面<>包的+数量代表y的值，做 $x*y$ ，得到ASCII码转化字母

如第一行 $x=17$ ， $y=6$ ， $x*y=102$ 为f

用脚本解码的思路是

遍历，越过>和[<，检验是否符合标准，有无固定的--++--]

```
import sympy

def reverse_bf(bf_code):
    flag = []
    i = 0
    n = len(bf_code)
    while i < n:
        if bf_code[i] == '>':
            # Start of a new segment
            i += 1
            # Count x
            x = 0
            while i < n and bf_code[i] == '+':
                x += 1
                i += 1
            # Expect '<'
            if i >= n or bf_code[i] != '[' or bf_code[i+1] != '<':
                raise ValueError("Invalid BF segment")
            i += 2
            # Count y
            y = 0
            while i < n and bf_code[i] == '+':
                y += 1
                i += 1
            # Expect '>--++--]'
            remaining = bf_code[i:i+9]
            if remaining != '>--++--]':
                raise ValueError("Invalid BF segment")
            i += 9
            # Expect '<[-]'
            if bf_code[i:i+4] != '<[-]':
```

```
        raise ValueError("Invalid BF segment")
    i += 4
    # Calculate character
    char = chr(x * y)
    flag.append(char)
else:
    i += 1
return ''.join(flag)
print(reverse_bf('bf代码内容'))
```