Hgame week4 wp

RE

again!

无法pycdc直接导出源代码，dis出来

```
 0      0 RESUME           0

 1      2 LOAD_CONST          0 (0)
        4 LOAD_CONST          1 (None)
        6 IMPORT_NAME         0 (hashlib)
        8 STORE_NAME          0 (hashlib)

 2     10 PUSH_NULL
       12 LOAD_NAME           1 (print)
       14 LOAD_CONST          2 ('you should use this execute file to decrypt "bin2"')
       16 PRECALL             1
       20 CALL                1
       30 POP_TOP

 3     32 PUSH_NULL
       34 LOAD_NAME           1 (print)
       36 LOAD_CONST          3 ('hint:md5')
       38 PRECALL             1
       42 CALL                1
       52 POP_TOP

 4     54 PUSH_NULL
       56 LOAD_NAME           2 (bytearray)
```

逆向出源码

```python
import hashlib

print("you should use this execute file to decrypt 'bin2'")
print("hint:md5")

s = bytearray()

with open("bin1.pyc", "rb") as f:
    f = f.read()

t = list("jkasnwojasd")

f = bytearray(f) # 将 f 转换成 bytearray 类型

for i in range(15):
    f[i] =((ord(t[i%len(t)])+ord(t[i%6]))^(f[i%6]+f[i]))%256
    s.append(f[i])
```

```
    print(hex(f[i]))


print(s)

md5_hash = hashlib.md5(bytes(s)).hexdigest()
print(md5_hash)
```

得到md5和bin2进行循环异或，发现是xxtea算法

```
#include <stdio.h>
#include <stdint.h>
#define DELTA 0x7937B99E
#define MX (((z>>5^y<<2) + (y>>3^z<<4)) ^ ((sum^y) + (key[(p&3)^e] ^ z)))

void btea(uint32_t* v, int n, uint32_t const key[4])
{
    uint32_t y, z, sum;
    unsigned p, rounds, e;
    if (n > 1)            /* Coding Part */
    {
        rounds = 6 + 52 / n;
        sum = 0;
        z = v[n - 1];
        do
        {
            sum += DELTA;
            e = (sum >> 2) & 3;
            for (p = 0; p < n - 1; p++)
            {
                y = v[p + 1];
                z = v[p] += MX;
            }
            y = v[0];
            z = v[n - 1] += MX;
        } while (--rounds);
    }
    else if (n < -1)      /* Decoding Part */
    {
        n = -n;
        rounds = 6 + 52 / n;
        sum = rounds * DELTA;
        y = v[0];
        do
        {
```

```
            e = (sum >> 2) & 3;
            for (p = n - 1; p > 0; p--)
            {
                z = v[p - 1];
                y = v[p] -= MX;
            }
            z = v[n - 1];
            y = v[0] -= MX;
            sum -= DELTA;
        } while (--rounds);
    }
}


int main()
{
    uint32_t v[] = {
0x506FB5C3, 0xB9358F45, 0xC91AE8C7, 0x3820E280, 0xD13ABA83, 0x975CF554,
0x4352036B, 0x1CD20447
    };
    uint32_t const k[4] = { 0x1234,0x2341,0x3412,0x4123 };
    int n = 8;
    printf("加密后的数据: %x %x\n", v[0], v[1]);
    btea(v, -n, k);
    printf("解密后的数据: %x %x\n", v[0], v[1]);
    return 0;
}
```

change

```
#include<stdio.h>
int main()
{
        unsigned char key[] =
        {
          0x61, 0x6D, 0x32, 0x71, 0x61, 0x73, 0x6C
        };
        unsigned char enc[] =
        {
          0x13, 0x0A, 0x5D, 0x1C, 0x0E, 0x08, 0x23, 0x06, 0x0B, 0x4B,
          0x38, 0x22, 0x0D, 0x1C, 0x48, 0x0C, 0x66, 0x15, 0x48, 0x1B,
          0x0D, 0x0E, 0x10, 0x4F
        };
        /*for (int i = 0; i < 24; i++)
        {
```

```
16                  enc[i] -= i;
17          }*/
18          for (int i = 0; i < 24; i++)
19          {
20                  if (i % 2)
21                  {
22                          enc[i] = enc[i] ^ key[i % 7];
23                  }
24                  else
25                  {
26                          enc[i] -= 10;
27                          enc[i] = enc[i] ^ key[i % 7];
28                  }
29                  printf("%c", enc[i]);
30          }
31
32
33          return 0;
34  }
```

crackme2

发现有异常处理进行patch，发现有smc进行代码处理

```
v39 = a1[17];
v22 = a1[6];
v18 = a1[9];
if ( v18
   + 201 * v24
   + 194 * v10
   + 142 * v20
   + 114 * v39
   + 103 * v11
   + 52 * (v17 + v31)
   + ((v9 + v23) << 6)
   + 14 * (v21 + 4 * v25 + v25)
   + 9 * (v40 + 23 * v27 + v2 + 3 * v1 + 4 * v2 + 4 * v6)
   + 5 * (v16 + 23 * v30 + 2 * (v3 + 2 * v19) + 5 * v5 + 39 * v15 + 51 * v4)
   + 24 * (v8 + 10 * v28 + 4 * (v42 + v7 + 2 * v26))
   + 62 * v22
   + 211 * v41
   + 212 * v29 != 296473 )
  return 0i64;
v38 = 2 * v16;
if ( 207 * v41
   + 195 * v22
   + 151 * v40
   + 57 * v5
   + 118 * v6
```

发现大量的计算流程，直接交给z3

```
1  from z3 import*
```

```
a=[0]*32
s=Solver()
for i in range(32):
    a[i]=z3.Int('a1['+str(i)+']')
v37 = 2 * a[20]
v38 = 2 * a[24]
v33 = 2 * a[5]
v32 = 2 * a[9]
v35 = a[27] + a[14]
v34 = 2 * a[13]
v12 = a[3] + 2 * (a[13] + 4 * (a[10] + a[26])) + a[13] + 4 * (a[10] + a[26])
v36 = 3 * a[19]
v13 = a[23] + a[25] + 8 * a[23] + 4 * (a[28] + 2 * a[11])
s.add( a[9]
+ 201 * a[4]
+ 194 * a[3]
+ 142 * a[15]
+ 114 * a[17]
+ 103 * a[2]
+ 52 * (a[26] + a[13])
+ ((a[12] + a[16]) *64)
+ 14 * (a[19] + 4 * a[27] + a[27])
+ 9 * (a[7] + 23 * a[11] + a[21] + 3 *a[25] + 4 * a[21] + 4 * a[23])
+ 5 * (a[24] + 23 * a[14] + 2 * (a[31] + 2 * a[30]) + 5 * a[0] + 39 * a[18] +
51 * a[29])
+ 24 * (a[28] + 10 * a[1] + 4 * (a[22] + a[8] + 2 * a[20]))
+ 62 * a[6]
+ 211 * a[5]
+ 212 * a[10] == 296473 )
s.add( 207 * a[5]
+ 195 * a[6]
+ 151 * a[7]
+ 57 * a[0]
+ 118 * a[23]
+ 222 * a[22]
+ 103 * a[8]
+ 181 * a[28]
+ 229 * a[12]
+ 142 * a[13]
+ 51 * a[10]
+ 122 * (a[20] + a[15])
+ 91 * (a[21] + 2 * a[24])
+ 107 * (a[11] + a[27])
+ 81 * (a[26] + 2 * a[9] + a[9])
+ 45 * (a[30] + 2 * (a[2] + a[4]) + a[2] + a[4])
+ 4 * (3 * (a[16] + a[19] + 2 * a[16] + 5 * a[29]) + a[17] + 29 * (a[3] +
a[25]) + 25 * a[18])
```

```
47  + 26 * a[1]
48  + 101 * a[14]
49  + 154 * a[31] == 354358 )
50  s.add( 177 * a[7]
51  + 129 * a[20]
52  + 117 * a[22]
53  + 143 * a[1]
54  + 65 * a[28]
55  + 137 * a[27]
56  + 215 * a[19]
57  + 93 * a[13]
58  + 235 * a[17]
59  + 203 * a[2]
60  + 15 * (a[8] + 17 * a[14])
61  + 2
62  * (a[4]
63  + 91 * a[12]
64  + 95 * a[10]
65  + 51 * a[5]
66  + 81 * a[15]
67  + 92 * a[9]
68  + 112 * (a[3] + a[23])
69  + 32 * (a[6] + 2 * (a[25] + a[16]))
70  + 6 * (a[21] + 14 * a[24] + 19 * a[18])
71  + 83 * a[0]
72  + 53 * a[29]
73  + 123 * a[30])
74  + a[26]
75  + 175 * a[11]
76  + 183 * a[31] == 448573)
77  s.add(113 * a[30]
78  + 74 * a[31]
79  + 238 * a[23]
80  + 140 * a[21]
81  + 214 * a[20]
82  + 242 * a[28]
83  + 160 * a[19]
84  + 136 * a[16]
85  + 209 * a[12]
86  + 220 * a[13]
87  + 50 * a[4]
88  + 125 * a[3]
89  + 175 * a[15]
90  + 23 * a[17]
91  + 137 * a[6]
92  + 149 * a[9]
93  + 83 * (a[29] + 2 * a[14])
```

```
 94 + 21 * (9 * a[10] + a[24])
 95 + 59 * (4 * a[11] + a[26])
 96 + 41 * (a[25] + a[5])
 97 + 13 * (a[8] + 11 * (a[7] + a[18]) + 6 * a[22] + 4 * (a[1] + 2 * a[2]) + a[1]
    + 2 * a[2] + 17 * a[0])
 98 + 36 * a[27] == 384306)
 99 s.add(229 * a[19]
100 + 78 * a[25]
101 + a[21]
102 + a[12]
103 + 133 * a[11]
104 + 74 * a[23]
105 + 69 * a[20]
106 + 243 * a[8]
107 + 98 * a[1]
108 + 253 * a[28]
109 + 142 * a[27]
110 + 175 * a[13]
111 + 105 * a[5]
112 + 221 * a[3]
113 + 121 * a[17]
114 + 218 * (a[30] + a[10])
115 + 199 * (a[4] + a[14])
116 + 33 * (a[7] + 7 * a[26])
117 + 4 * (27 * a[15] + 50 * a[2] + 45 * a[9] + 19 * (a[31] + a[22]) + a[24] + 16
    * a[16] + 52 * a[29])
118 + 195 * a[6]
119 + 211 * a[0]
120 + 153 * a[18] == 424240)
121 s.add(181 * a[27]
122 + 61 * a[21]
123 + 65 * a[19]
124 + 58 * a[13]
125 + 170 * a[10]
126 + 143 * a[4]
127 + 185 * a[3]
128 + 86 * a[2]
129 + 97 * a[6]
130 + 235 * (a[16] + a[11])
131 + 3
132 * (53 * a[5]
133 + 74 * (a[28] + a[31])
134 + 13 * (a[22] + 6 * a[12])
135 + 11 * (a[17] + 7 * a[15])
136 + 15 * (a[9] + 4 * a[26])
137 + a[8]
138 + 35 * a[25]
```

```
139 + 29 * a[18])
140 + 4 * (57 * a[23] + 18 * (a[0] + v37) + a[1] + 17 * a[24] + 55 * a[14])
141 + 151 * a[7]
142 + 230 * a[29]
143 + 197 * a[30] == 421974)
144 s.add(
145 209 * a[19]
146 + 249 * a[14]
147 + 195 * a[21]
148 + 219 * a[27]
149 + 201 * a[17]
150 + 85 * a[9]
151 + 213 * (a[26] + a[13])
152 + 119 * (a[2] + 2 * a[5])
153 + 29 * (8 * a[4] + a[7] + 4 * a[11] + a[11])
154 + 2
155 * (a[28]
156 + 55 * (2 * a[10] + a[30])
157 + 3 * (a[3] + 39 * a[12] + 2 * (a[23] + 20 * a[15]) + 35 * a[8])
158 + 4 * (a[0] + 31 * a[22] + 28 * a[31])
159 + 26 * a[1]
160 + 46 * (v37 + a[24])
161 + 98 * a[25])
162 + 53 * a[16]
163 + 171 * a[18]
164 + 123 * a[29] == 442074)
165 s.add(
166 162 * a[30]
167 + 74 * a[0]
168 + 28 * a[11]
169 + 243 * a[22]
170 + 123 * a[1]
171 + 73 * a[28]
172 + 166 * a[16]
173 + 94 * a[4]
174 + 113 * a[2]
175 + 193 * a[6]
176 + 122 * (a[23] + 2 * a[8])
177 + 211 * (a[3] + a[27])
178 + 21 * (a[26] + 7 * a[5])
179 + 11 * (a[29] + 23 * (a[24] + a[17]) + 2 * (a[7] + 5 * a[14] + 2 * (2 * a[9] +
    a[10]) + 2 * a[9] + a[10]))
180 + 5 * (46 * a[12] + 26 * a[15] + 4 * (a[13] + 2 * a[19]) + a[18] + 27 * a[21]
    + 10 * a[25])
181 + 36 * (a[31] + 5 * a[20]) == 376007)
182 s.add(
183 63 * a[30]
```

```
184  + 143 * a[0]
185  + 250 * a[23]
186  + 136 * a[21]
187  + 214 * a[7]
188  + 62 * a[20]
189  + 221 * a[22]
190  + 226 * a[8]
191  + 171 * a[1]
192  + 178 * a[28]
193  + 244 * a[16]
194  + (a[12]* 128)
195  + 150 * a[13]
196  + 109 * a[10]
197  + 70 * a[5]
198  + 127 * a[15]
199  + 204 * a[17]
200  + 121 * a[6]
201  + 173 * a[9]
202  + 69 * (a[27] + a[14] + a[11])
203  + 74 * (a[24] + 2 * a[18] + a[18])
204  + 22 * (7 * a[4] + a[26] + 10 * a[2])
205  + 40 * (a[25] + 4 * a[19] + a[19])
206  + 81 * a[3]
207  + 94 * a[29]
208  + 84 * a[31] == 411252)
209  s.add(229 * a[18]
210  + 121 * a[29]
211  + 28 * a[14]
212  + 206 * a[24]
213  + 145 * a[11]
214  + 41 * a[25]
215  + 247 * a[23]
216  + 118 * a[20]
217  + 241 * a[1]
218  + 79 * a[28]
219  + 102 * a[27]
220  + 124 * a[16]
221  + 65 * a[12]
222  + 68 * a[13]
223  + 239 * a[26]
224  + 148 * a[4]
225  + 245 * a[17]
226  + 115 * a[2]
227  + 163 * a[6]
228  + 137 * a[9]
229  + 53 * (a[0] + 2 * a[10])
230  + 126 * (a[7] + 2 * a[3])
```

```
231 + 38 * (a[8] + a[19] + 4 * a[8] + 6 * a[5])
232 + 12 * (a[21] + 16 * a[22])
233 + 109 * a[15]
234 + 232 * a[31]
235 + 47 * a[30] == 435012)
236 s.add(209 * a[19]
237 + 233 * a[7]
238 + 93 * a[25]
239 + 241 * a[21]
240 + 137 * a[28]
241 + 249 * a[26]
242 + 188 * a[10]
243 + 86 * a[4]
244 + 246 * a[3]
245 + 149 * a[15]
246 + 99 * a[2]
247 + 37 * a[6]
248 + 219 * a[9]
249 + 17 * (a[23] + 10 * a[27])
250 + 49 * (a[0] + 3 * a[31] + 4 * a[1] + a[1])
251 + 5 * (16 * a[17] + 11 * (a[5] + 2 * a[11] + a[11]) + 12 * a[8] + a[13] + 30 *
    a[24] + 27 * a[30])
252 + 18 * (a[16] + 2 * (a[29] + a[20] + 2 * a[29]) + a[29] + a[20] + 2 * a[29])
253 + 24 * a[12]
254 + 109 * a[22]
255 + 183 * a[14]
256 + 154 * a[18] == 392484)
257 s.add(
258 155 * a[18]
259 + 247 * a[7]
260 + 157 * a[1]
261 + 119 * a[16]
262 + 161 * a[26]
263 + 133 * a[15]
264 + 85 * a[6]
265 + 229 * (a[8] + a[4])
266 + 123 * (2 * a[13] + a[22])
267 + 21 * (a[5] + 12 * a[14])
268 + 55 * (a[12] + a[0] + a[9] + 2 * a[0])
269 + 15 * (a[31] + 16 * a[3] + 9 * a[19])
270 + 2
271 * (a[21]
272 + 115 * a[10]
273 + 111 * a[24]
274 + 26 * a[23]
275 + 88 * a[28]
276 + 73 * a[17]
```

```
277 + 71 * a[2]
278 + 28 * (a[20] + 2 * (a[27] + 2 * a[25]))
279 + 51 * a[11]
280 + 99 * a[29]
281 + 125 * a[30]) == 437910)
282 s.add(220 * a[31]
283 + 200 * a[29]
284 + 139 * a[18]
285 + 33 * a[0]
286 + 212 * a[14]
287 + 191 * a[24]
288 + 30 * a[11]
289 + 233 *a[25]
290 + 246 * a[23]
291 + 89 * a[21]
292 + 252 * a[7]
293 + 223 * a[22]
294 + 19 * a[27]
295 + 141 * a[19]
296 + 163 * a[12]
297 + 185 * a[26]
298 + 136 * a[13]
299 + 46 * a[4]
300 + 109 * a[3]
301 + 217 * a[17]
302 + 75 * a[6]
303 + 157 * a[9]
304 + 125 * (a[2] + a[30])
305 + 104 * (v33 + a[15])
306 + 43 * (a[1] + 2 * a[10] + a[10])
307 + 32 * (a[28] + a[8] + 2 * a[28] + 2 * (a[16] + a[20])) == 421905)
308 s.add(211 * a[4]
309 + 63 * a[18]
310 + 176 * a[0]
311 + 169 * a[24]
312 + 129 * a[11]
313 + 146 * a[7]
314 + 111 * a[20]
315 + 68 * a[22]
316 + 39 * a[27]
317 + 188 * a[16]
318 + 130 * a[12]
319 + (a[13]*64)
320 + 91 * a[5]
321 + 208 * a[15]
322 + 145 * a[17]
323 + 247 * a[9]
```

```
 + 93 * (a[6] + a[26])
 + 71 * (a[23] + 2 * a[2])
 + 103 * (a[28] + 2 * a[14])
 + 6 * (a[19] + 10 * a[1] + 28 * a[8] + 9 * a[10] + 19 * a[21] + 24 * a[25] + 22
     * a[31])
 + 81 * a[3]
 + 70 * a[29]
 + 23 * a[30] == 356282)
s.add(
94 * a[22]
 + 101 * a[21]
 + 152 * a[7]
 + 200 * a[8]
 + 226 * a[28]
 + 211 * a[16]
 + 121 * a[4]
 + 74 * a[2]
 + 166 * a[9]
 + ((a[23] + 3 * a[1]) * 64)
 + 41 * (4 * a[12] + a[19])
 + 23 * (a[17] + 11 * a[5])
 + 7 * (a[15] + 10 * a[27] + 2 * v12 + v12)
 + 3 * (78 * a[14] + 81 * a[24] + 55 * a[11] + 73 *a[25] + 4 * a[20] + a[18] +
    85 * a[31] + 65 * a[30])
 + 62 * a[6]
 + 88 * a[0]
 + 110 * a[29] == 423091)
s.add(133 * a[6]
 + 175 * a[18]
 + 181 * a[14]
 + 199 * a[24]
 + 123 * a[11]
 + 242 * a[25]
 + 75 * a[23]
 + 69 * a[21]
 + 153 * a[7]
 + 33 * a[20]
 + 100 * a[22]
 + 229 * a[8]
 + 177 * a[28]
 + 134 * a[13]
 + 179 * a[10]
 + 129 * a[5]
 + 14 * a[3]
 + 247 * a[4]
 + 228 * a[15]
 + 92 * a[2]
```

```
369 + 86 * (a[12] + v32)
370 + 94 * (a[16] + a[19])
371 + 37 * (a[26] + 4 * a[31])
372 + 79 * (a[27] + 2 * a[1])
373 + 72 * a[0]
374 + 93 * a[17]
375 + 152 * a[29]
376 + 214 * a[30] == 391869)
377 s.add( 211 * a[4]
378 + 213 * a[9]
379 + 197 * a[7]
380 + 159 * a[27]
381 + 117 * a[19]
382 + 119 * a[12]
383 + 98 * a[26]
384 + 218 * a[5]
385 + 106 * a[17]
386 + 69 * a[2]
387 + 43 * (a[21] + a[10] + 2 * a[21])
388 + 116 * (a[29] + a[3] + v37)
389 + 5 * (a[22] + 9 * a[16] + 35 * a[15] + 37 * a[13])
390 + 11 * (a[24] + 13 * a[11] + 5 * a[0] + 8 * a[14])
391 + 6 * (29 * a[1] + 25 * a[28] + 38 * a[6] + a[18] + 13 * a[25] + 10 * a[31])
392 + 136 * a[8]
393 + 142 * a[23]
394 + 141 * a[30] == 376566)
395 s.add(173 * a[31]
396 + 109 * a[18]
397 + 61 * a[14]
398 + 187 * a[25]
399 + 79 * a[23]
400 + 53 * a[7]
401 + 184 * a[19]
402 + 43 * a[16]
403 + 41 * a[12]
404 + 166 * a[13]
405 + 193 * a[5]
406 + 58 * a[4]
407 + 146 * a[3]
408 + (a[15] *64)
409 + 89 * a[17]
410 + 121 * a[2]
411 + 5 * (a[26] + 23 * a[28])
412 + 7 * (29 * a[9] + a[10] + 4 * a[8])
413 + 13 * (3 * a[22] + a[24] + 7 * a[20] + 13 * a[21])
414 + 3 * (a[29] + 83 * a[0] + 51 * a[11] + 33 * a[6] + 8 * (a[30] + 4 * a[1]) + 18
     * a[27]) == 300934)
```

```
s.add (
78 * a[25]
+ 131 * a[0]
+ 185 * a[24]
+ 250 * a[7]
+ 90 * a[20]
+ 129 * a[22]
+ 255 * a[1]
+ 206 * a[28]
+ 239 * a[27]
+ 150 * a[3]
+ 253 * a[17]
+ 104 * a[6]
+ 58 * (a[21] + 2 * a[8])
+ 96 * (a[18] + a[13])
+ 117 * (a[12] + 2 * a[29])
+ 27 * (a[26] + 8 * a[9] + a[9])
+ 19 * (a[16] + 3 * a[19] + 4 * a[10] + a[10])
+ 7 * (22 * a[5] + 3 * (a[2] + 11 * a[4]) + a[31] + 29 * a[23] + 14 * a[11])
+ 109 * a[15]
+ 102 * a[14]
+ 100 * a[30] == 401351)
s.add (233 * a[30]
+ 71 * a[0]
+ 209 * a[11]
+ 82 * a[23]
+ 58 * a[20]
+ 53 * a[27]
+ 113 * a[16]
+ 206 * a[13]
+ 39 * a[5]
+ 163 * a[15]
+ 222 * a[2]
+ 191 * a[9]
+ 123 * (a[8] + a[7])
+ 69 * (a[12] + 2 * a[6] + a[6])
+ 9 * (a[31] + 8 * a[4] + 7 * (3 *a[25] + a[1]) + 5 * a[24] + 19 * a[14])
+ 4 * (a[18] + 26 * a[26] + 61 * a[10] + 43 * a[22] + 49 * a[21] + 32 * a[29])
+ 10 * (7 * (a[28] + v36) + a[17] + 12 * a[3]) == 368427)
s.add (139 * a[14]
+ 53 * a[0]
+ 158 * a[24]
+ 225 *a[25]
+ 119 * a[23]
+ 67 * a[21]
+ 213 * a[7]
+ 188 * a[1]
```

```
462  + 152 * a[28]
463  + 187 * a[19]
464  + 129 * a[16]
465  + 54 * a[12]
466  + 125 * a[26]
467  + 170 * a[4]
468  + 184 * a[2]
469  + 226 * a[6]
470  + 253 * a[9]
471  + 26 * (a[10] + a[5])
472  + 97 * (a[29] + 2 * a[27])
473  + 39 * (5 * a[20] + a[11])
474  + 21 * (a[17] + 8 * a[22])
475  + 12 * (17 * a[3] + a[13] + 15 * a[8] + 12 * a[30])
476  + 165 * a[15]
477  + 88 * a[18]
478  + 157 * a[31] == 403881)
479  s.add( 114 * a[31]
480  + 61 * a[11]
481  + 134 * a[7]
482  + 62 * a[22]
483  + 89 * a[12]
484  + 211 * a[26]
485  + 163 * a[5]
486  + 66 * a[4]
487  + 201 * (a[8] + a[9])
488  + 47 * (5 * a[24] + a[6])
489  + 74 * (a[29] + a[13])
490  + 142 * (a[21] + a[1])
491  + 35 * (a[15] + 6 * a[20])
492  + 39 * (a[18] + 6 * a[14])
493  + 27 * (a[27] + 9 * a[16] + 8 * a[23])
494  + 4 * (a[19] + 63 * a[30] + 2 * (a[25] + 12 * (a[3] + a[0]) + 8 * a[2] + 26 *
     a[10]))
495  + 10 * (a[28] + 4 * a[17] + a[17]) == 382979)
496  s.add (122 * a[27]
497  + 225 * a[19]
498  + 52 * a[16]
499  + 253 * a[12]
500  + 197 * a[26]
501  + 187 * a[13]
502  + 181 * a[10]
503  + 183 * a[5]
504  + 47 * a[15]
505  + 229 * a[17]
506  + 88 * a[6]
507  + 127 * (a[3] + v32)
```

```
508 + 37 * (a[8] + 3 * a[31])
509 + ((a[2] + 2 * a[14] + a[14]) *64)
510 + 7 * (21 * a[28] + a[11] + 18 * (a[29] + a[25] + v38))
511 + 6 * (23 * a[4] + a[20] + 17 * a[21] + 39 * a[23])
512 + 10 * (a[0] + 11 * a[1] + 21 * a[22])
513 + 149 * a[30]
514 + 165 * a[7]
515 + 121 * a[18] == 435695)
516 s.add (165 * a[15]
517 + 223 * a[29]
518 + 249 * a[0]
519 + 199 *a[25]
520 + 135 * a[21]
521 + 133 * a[20]
522 + 254 * a[22]
523 + 111 * a[8]
524 + 189 * a[1]
525 + 221 * a[27]
526 + 115 * a[19]
527 + 186 * a[12]
528 + 79 * a[5]
529 + 217 * a[4]
530 + 122 * a[2]
531 + 38 * a[9]
532 + 109 * (v34 + a[10])
533 + 14 * (a[28] + 17 * a[7] + 8 * (a[23] + v38))
534 + 4 * (11 * (5 * a[14] + a[17]) + 6 * (a[3] + 2 * a[6]) + a[11] + 52 * a[26] +
    50 * a[16])
535 + 229 * a[18]
536 + 86 * a[31]
537 + 234 * a[30] == 453748)
538 s.add (181 * a[27]
539 + 94 * a[22]
540 + 125 *a[25]
541 + 226 * a[20]
542 + 155 * a[8]
543 + 95 * a[19]
544 + 212 * a[26]
545 + 91 * a[13]
546 + 194 * a[10]
547 + 98 * a[4]
548 + 166 * a[2]
549 + 120 * a[6]
550 + 59 * a[9]
551 + 32 * (a[12] + a[28])
552 + 158 * (a[23] + a[0])
553 + 101 * (a[5] + a[30])
```

```
554 + 63 * (a[29] + 2 * a[16])
555 + 67 * (a[1] + 2 * a[15])
556 + 11 * (a[17] + 10 * a[24] + 11 * a[3])
557 + 39 * (a[14] + 4 * (a[21] + a[18]))
558 + 233 * a[7]
559 + 56 * a[11]
560 + 225 * a[31] == 358321)
561 s.add(229 * a[19]
562 + 135 * a[29]
563 + 197 * a[18]
564 + 118 * a[0]
565 + 143 * a[24]
566 + 134 * a[23]
567 + 204 * a[7]
568 + 173 * a[20]
569 + 81 * a[8]
570 + 60 * a[1]
571 + 58 * a[28]
572 + 179 * a[16]
573 + 142 * a[12]
574 + 178 * a[26]
575 + 230 * a[13]
576 + 148 * a[10]
577 + 224 * a[5]
578 + 194 * a[4]
579 + 223 * a[3]
580 + 87 * a[15]
581 + 200 * a[17]
582 + 233 * a[2]
583 + 49 * a[6]
584 + 127 * v35
585 + 31 * (4 * a[11] + a[9])
586 + 42 * (a[25] + 6 * a[21])
587 + 109 * a[22]
588 + 75 * a[31]
589 + 165 * a[30] == 456073)
590 s.add (41 * a[29]
591 + 253 * a[31]
592 + 163 * a[18]
593 + 193 * a[14]
594 + 155 * a[24]
595 + 113 * a[11]
596 + 131 * a[23]
597 + 55 * a[21]
598 + 21 * a[7]
599 + 53 * a[20]
600 + 13 * a[28]
```

```
601 + 201 * a[27]
602 + 237 * a[12]
603 + 223 * a[13]
604 + 95 * a[4]
605 + 194 * a[15]
606 + 62 * a[17]
607 + 119 * a[2]
608 + 171 * a[6]
609 + 135 * a[9]
610 + 69 * (a[3] + 3 * a[1])
611 + 211 * (a[25] + a[10])
612 + 4 * (43 * a[8] + a[22] + 40 * a[26])
613 + 6 * (a[0] + 33 * a[5] + 20 * (2 * a[30] + a[19]) + 24 * a[16]) == 407135)
614 s.add (
615 111 * a[30]
616 + 190 * a[31]
617 + 149 * a[29]
618 + 173 * a[1]
619 + 118 * a[16]
620 + 146 * a[10]
621 + 179 * a[3]
622 + 51 * a[15]
623 + 49 * a[17]
624 + 61 * a[2]
625 + 125 * a[6]
626 + 162 * a[9]
627 + 214 * v35
628 + 14 * (v34 + a[4])
629 + 178 * (a[5] + a[24])
630 + 11 * (4 * a[12] + a[19] + 17 * a[22])
631 + 65 * (a[20] + a[26] + 2 * a[20] + 2 * a[0])
632 + 4 * (a[8] + 38 * a[18] + 4 * v13 + v13 + 8 * a[7] + 43 * a[21]) == 369835)
633 s.add(27 * a[11]
634 + 223 * a[23]
635 + 147 * a[20]
636 + 13 * a[19]
637 + 35 * (a[26] + 7 * a[29])
638 + 57 * (a[30] + v32 + 3 * a[2])
639 + 11 * (a[25] + 17 * (a[12] + a[0]) + 10 * a[24] + 3 * a[13])
640 + 2
641 * (53 * a[16]
642 + a[27]
643 + 38 * a[18]
644 + 43 * a[22]
645 + 115 * a[10]
646 + 61 * a[6]
647 + 111 * (a[3] + a[7])
```

```
+ 14 * (a[15] + a[8] + 2 * a[8] + 8 * a[1])
+ 109 * a[21]
+ 100 * a[5]
+ 63 * a[28])
+ 93 * a[17]
+ 251 * a[14]
+ 131 * a[31] == 393303)
s.add( 116 * a[12]
+ 152 * a[10]
+ 235 * a[15]
+ 202 * a[9]
+ 85 * (a[28] + 3 * a[2])
+ 221 * (a[24] + a[7])
+ 125 * (v33 + a[4])
+ 7 * (19 * a[29] + 9 * (a[3] + 2 * a[27]) + a[21] + 33 * a[31] + 32 * a[30])
+ 3
* (71 * a[17] + 43 * a[6] + 32 * (a[26] + a[20]) + 15 * (a[0] + a[23] + 2 *
    a[16]) + a[1] + 74 * a[13] + 48 * a[22])
+ 10 * (a[19] + 11 * a[14] + 16 * a[18])
+ 136 * a[8]
+ 106 *a[25]
+ 41 * a[11] == 403661)
s.add( 127 * a[29]
+ 106 * a[18]
+ 182 * a[14]
+ 142 * a[0]
+ 159 * a[24]
+ 17 *a[25]
+ 211 * a[23]
+ 134 * a[21]
+ 199 * a[8]
+ 103 * a[1]
+ 247 * a[16]
+ 122 * a[12]
+ 95 * a[5]
+ 62 * a[3]
+ 203 * a[17]
+ 16 * a[2]
+ 41 * (6 * a[22] + a[27])
+ 9 * (22 * a[4] + a[15] + 27 * a[13] + 28 * a[7])
+ 10 * (a[28] + a[6] + v36 + 8 * a[26] + 2 * (a[6] + v36 + 8 * a[26]) + 13 *
    a[10])
+ 6 * (23 * a[11] + a[20])
+ 213 * a[9]
+ 179 * a[31]
+ 43 * a[30] == 418596 )
s.add(
```

```
149 * a[30]
+ a[25]
+ 133 * a[6]
+ 207 * a[5]
+ 182 * a[20]
+ 234 * a[8]
+ 199 * a[28]
+ 168 * a[19]
+ 58 * a[3]
+ 108 * a[15]
+ 142 * a[9]
+ 156 * (a[12] + a[27])
+ 16 * (a[10] + 6 * a[13])
+ 126 * (a[26] + 2 * a[17])
+ 127 * (a[29] + 2 * a[11] + a[7])
+ 49 * (a[14] + 4 * a[24])
+ 11 * (a[0] + 22 * a[2])
+ 5 * (a[18] + a[22] + 45 * a[4] + 50 * a[1])
+ 109 * a[21]
+ 124 * a[23]
+ 123 * a[31] == 418697)
if s.check() == z3.sat:
    print(s.model())
else:
    raise Exception("NO SOLUTION!")

```