Overview

Description:

```
I built a maze using goto statements! Navigate through it successfully to
get the flag.

Note: the provided file contains a dummy flag.
```

Use `file angr_management_test`

```
angr_management_test: ELF 64-bit LSB pie executable, x86-64, version 1
(SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2,
BuildID[sha1]=18a2a8b995afb6b68b7be69bc6a65a21f1f9d67c, for GNU/Linux 3.2.0,
not stripped
```

Use `checksec`:

```
minhduc@minhduc-Lecoo-N155A:~/CTF/ByuCTF/rev-angr-managemnet/challenge$
checksec --file=angr_management_test
RELRO           STACK CANARY      NX            PIE             RPATH
RUNPATH Symbols     FORTIFY Fortified   Fortifiable FILE
Full RELRO      Canary found      NX enabled    PIE enabled     No RPATH
No RUNPATH   48 Symbols   No    0       2           angr_management_test
```

Analyze and Solve

Use Ghidra to analyze ELF file `angr_management_test`, source code is messed up with many `if` and `while`

Based on comments in source code, I realized that I have to type each point on the map to get out the maze and flag will print

After this line, all lines used for confusing player with many `if (.. != ...)`

```
if (iVar1 == 0x270) {
printf("Arrived at %d\n",
        0x270);
puts("byuctf{test_flag}");
```

```
return 0;
}
```

So I can deleta all and the source code is now down to about 2000 lines - which can use ChatGPT or other AI to read and get the list of correct points to get out maze

List off points:

```
['256', '423', '495', '307', '39', '250', '391', '119', '105', '499', '123',
'104', '536', '257', '608', '253', '74', '365', '543', '300', '571', '506',
'595', '192', '383', '112', '17', '556', '93', '318', '114', '276', '18',
'216', '449', '414', '124', '503', '71', '407', '78', '285', '481', '66',
'381', '531', '82', '337', '600', '86', '230', '327', '472', '393', '348',
'331', '14', '207', '402', '548', '528', '168', '530', '490', '378', '408',
'518', '202', '87', '342', '329', '624']
```

Use Python to connect server and send all of them:

```
from pwn import *

HOST, PORT = 'chals.cyberjousting.com', 1368

p = remote(HOST, PORT)

with open('code.txt', 'r', encoding='utf-8') as f:
lines = f.read().splitlines()

for i in range (72):

p.recvuntil(b'Arrived')
p.sendline(lines[i])
print('Send: ', i)
p.interactive()
```

Flag: `byuctf{g3t_w1th_th3_c0ntr01_fl0w}`