Challenges solved:
Rev -> baby's first, baby's third, rebug 1
Pwn -> my_first_pwnie
Forensics -> 1black0white

Baby's first

Description: intro-level CTF challenge under rev
Approach: read the python file and find the flag
Solution: prompted ChatGPT to read the provided files and output the flag:
https://chat.openai.com/share/d332faac-0579-41af-919b-546ba170a1da

My_first_pwnie

Description: intro-level CTF challenge under pwn
Approach: print the contents of flag.txt
Solution: prompted ChatGPT to give me a command that prints the contents of a txt file in the current directory:
https://chat.openai.com/share/34901828-27da-414a-84f2-732edcd60a28

Baby's third

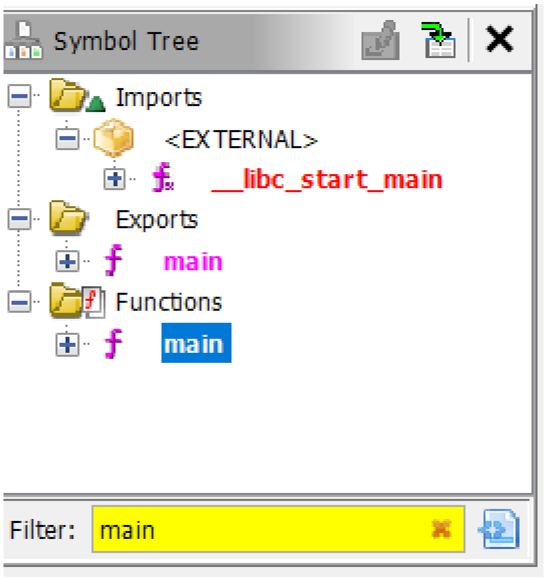
Description: intro-level CTF challenge under rev
Approach: feed readme.txt to chatgpt and perform the commands it suggested
Solution: "strings babysthird" command outputted a potential flag, which turned out to be correct:
https://chat.openai.com/share/5681e82a-7555-424a-aaa1-962dc4753941

Rebug 1

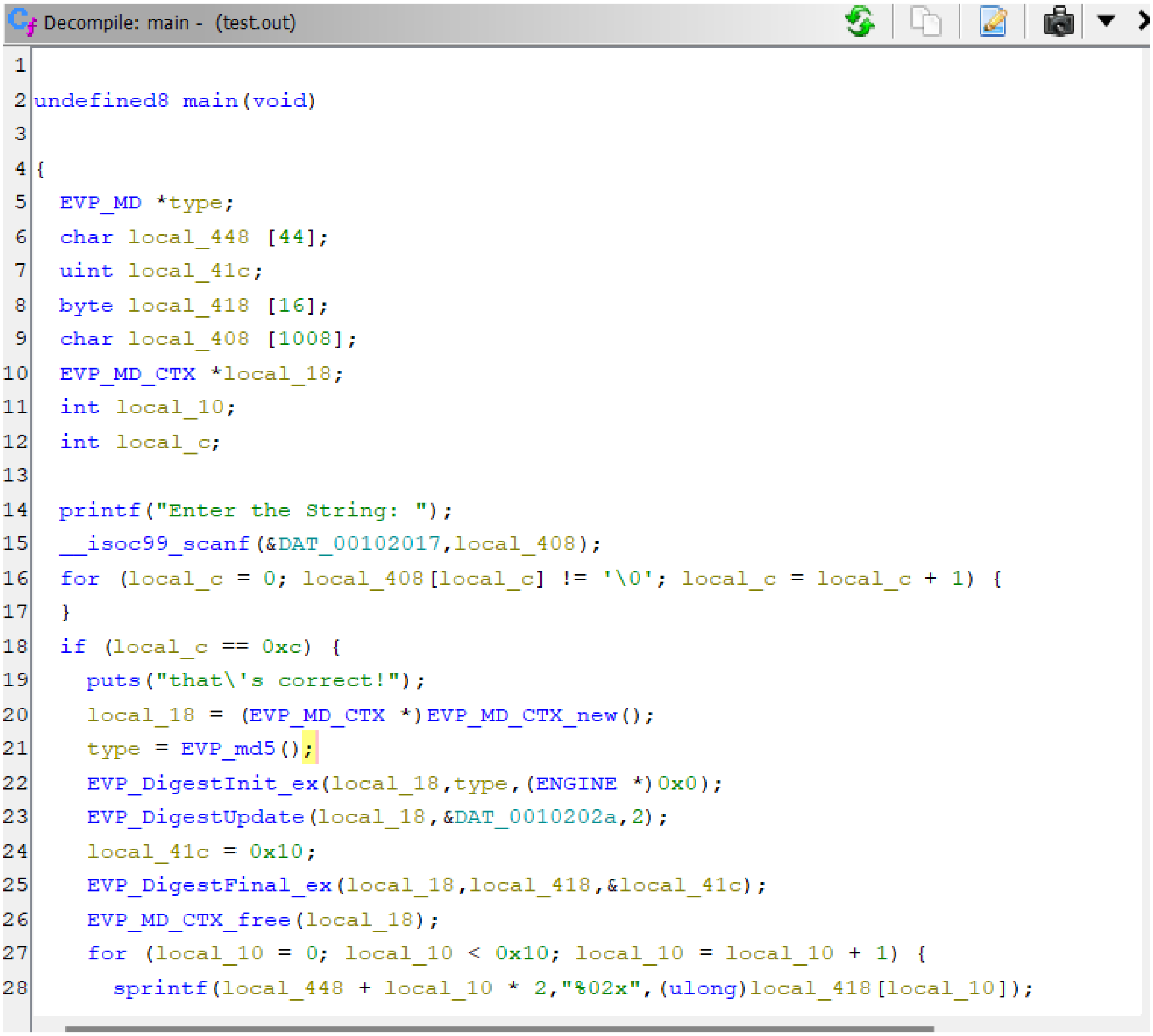
Description: intro-level CTF challenge under rev
Approach: use Ghidra to reverse engineer the binary file, then inspect the decompiled main function to guess the input requirements
Solution: any string of length 12 turned out to be the correct input, which ChatGPT found by using a code to make sure and give us an exemplary valid input:
https://chat.openai.com/share/13e53995-dbd2-4412-bd83-1027e9930dc5

Ghidra screenshots:

Locating the main function:



Inspecting the main function:



```
 1
 2 undefined8 main(void)
 3
 4 {
 5   EVP_MD *type;
 6   char local_448 [44];
 7   uint local_41c;
 8   byte local_418 [16];
 9   char local_408 [1008];
10   EVP_MD_CTX *local_18;
11   int local_10;
12   int local_c;
13
14   printf("Enter the String: ");
15   __isoc99_scanf(&DAT_00102017,local_408);
16   for (local_c = 0; local_408[local_c] != '\0'; local_c = local_c + 1) {
17   }
18   if (local_c == 0xc) {
19     puts("that\'s correct!");
20     local_18 = (EVP_MD_CTX *)EVP_MD_CTX_new();
21     type = EVP_md5();
22     EVP_DigestInit_ex(local_18,type,(ENGINE *)0x0);
23     EVP_DigestUpdate(local_18,&DAT_0010202a,2);
24     local_41c = 0x10;
25     EVP_DigestFinal_ex(local_18,local_418,&local_41c);
26     EVP_MD_CTX_free(local_18);
27     for (local_10 = 0; local_10 < 0x10; local_10 = local_10 + 1) {
28       sprintf(local_448 + local_10 * 2,"%02x",(ulong)local_418[local_10]);
```

Here, we can see from line 18 that our input must be of length 12 (0xc in hex).

1black0white

Description: easy/medium CTF challenge under forensics

Approach: generate a QR code based on the random numbers. To do that, turn decimal to binary, then make them equal lengths, and generate an image based on the hint (1black0white).

Solution: the solution was not achieved. Below are my ChatGPT prompt attempts to arrive at the solution:

https://chat.openai.com/share/3b2ac9d4-0eeb-444e-9bb2-e77e50027a10