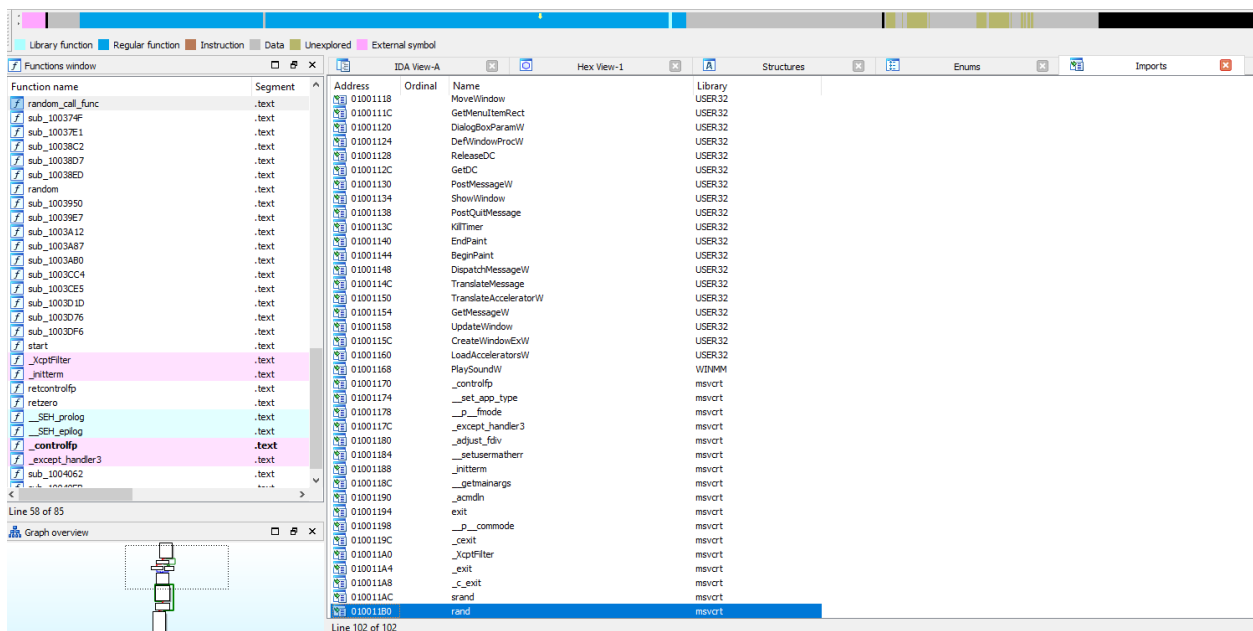


2.1 Task: crack Minesweeper

- when you open winmine.exe the flags should already be present where the bombs are;

Since the bombs are always randomly placed, I deduced that the program is using a randomization function, so I searched for this type of function in IDA. Looking at the imports in IDA (Imports tab), I saw the program calls rand and srand.



Analyzing the code, I noticed that rand is called only by one function.

```

1 int __stdcall random(int a1)
2 {
3     return rand() % a1;
4 }

```

This *random* function (as I renamed it) is only called by another function, renamed *random_call_func*.

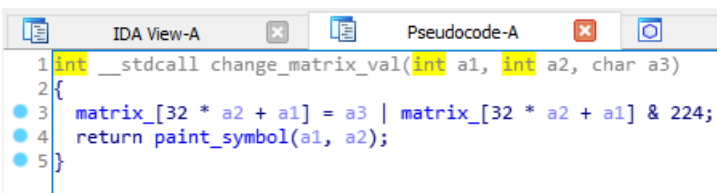
```

1 void random_call_func()
2 {
3     int v0; // ebx
4     int v1; // esi
5     int v2; // eax
6     signed int v3; // [esp-4h] [ebp-10h]
7
8     dword_1005164 = 0;
9     if ( dword_10056AC != dword_1005334 || uValue != dword_1005338 )
10         v3 = 6;
11     else
12         v3 = 4;
13     v0 = v3;
14     dword_1005334 = dword_10056AC;
15     dword_1005338 = uValue;
16     sub_1002ED5();
17     dword_1005160 = 0;
18     nr_bombe_ = dword_10056A4;
19     do
20     {
21         do
22         {
23             v1 = random(dword_1005334) + 1;
24             v2 = random(dword_1005338) + 1;
25         }
26         while ( matrix_[32 * v2 + v1] < 0 );
27         matrix_[32 * v2 + v1] |= 128u;
28         --nr_bombe_;
29     }
30     while ( nr_bombe_ );
31     dword_100579C = 0;
32     nr_bombe_ = dword_10056A4;
33     dword_1005194 = dword_10056A4;
34     dword_10057A4 = 0;
35     dword_10057A0 = dword_1005334 * dword_1005338 - dword_10056A4;
36     dword_1005000 = 1;
37     sub_100346A(0);
38     sub_1001950(v0);
39 }

```

I further assumed that v1 and v2 in the code above are storing the values of the line and the column where a bomb is located within the game matrix. I also determined, from the operation performed at line 27 of the above function, that a cell with a bomb is defined by the 8th bit, which is set to 1 (128 = b 10000000).

Then, I needed to find out how the flag value is defined. To do so, I inspected the matrix references to find where the values of its cells are being modified, and discovered the function that I afterwards renamed *change_matrix_val*:



```

1 int __stdcall change_matrix_val(int a1, int a2, char a3)
2 {
3     matrix_[32 * a2 + a1] = a3 | matrix_[32 * a2 + a1] & 224;
4     return paint_symbol(a1, a2);
5 }

```

The instruction above **matrix_[...] & 224** clears the five least significant bits of the value inside a cell (224 = b 11100000), this meaning that the bits that represent the flag, question mark and detonated bomb symbols will be part of those five bits. The symbol that will be pictured on the cell is determined by the parameter a3 (due to the | operator). To find out what values a3 can take, I examined the calls to the function above, and I observed that *change_matrix_val* is called twice with the constant 76 and once with a computed value:

```

}
change_matrix_val(a1, a2, v4);

```

```

else if ( dword_10057A4 )
{
    change_matrix_val(a1, a2, 76);
    result = sub_100347C(0);
}

```

Thus, it remained to be discovered what 76 represented and how the flag and question mark symbols were depicted. For this task, I used x32dbg to debug the program. By setting a breakpoint at the function *change_matrix_val*, I managed to uncover the following possible values for a3:

- 76 = detonated bomb
- 14 = flag
- 13 = question mark

Finally, to solve the first exercise, I used the x32dbg debugger again to patch the program with changes made to the instruction at line 27 in *random_call_func*.

010036E8	75 07	jne winmine.10036C7
010036EE	C1E0 05	shl eax,5
010036F0	8D8430 40530001	lea eax,dword ptr ds:[eax+esi+1005340]
010036F3	8008 80	or byte ptr ds:[eax],80
010036FA	FF0D 30530001	dec dword ptr ds:[1005330]
010036FD	75 C2	jne winmine.10036C7
01003703	8B0D 38530001	mov ecx,dword ptr ds:[1005338]
01003705	0FAF0D 34530001	imul ecx,dword ptr ds:[1005334]

This was transformed into:

010036E8	75 07	jne cracked_winmine.10036C7
010036EE	C1E0 05	shl eax,5
010036F0	8D8430 40530001	lea eax,dword ptr ds:[eax+esi+1005340]
010036FA	C600 8E	mov byte ptr ds:[eax],8E
010036FD	FF0D 30530001	dec dword ptr ds:[1005330]
01003703	75 C2	jne cracked_winmine.10036C7
01003705	8B0D 38530001	mov ecx,dword ptr ds:[1005338]

8E = 132 = 128 | 14

I replaced the or instruction with mov to make sure that the matrix cell value will be exactly 128 | 14 (equivalent to the five least significant bits being cleared in *change_matrix_val*).





- when you open winmine.exe put the question mark on positions that are blank;

To cover the blank cells with question marks, i needed to identify where the matrix values were initialized. Looking at the matrix XREFs, I spotted the following piece of code, which sets each cell of the matrix (out of a total of 864 cells) to value 15:

```

9 |
10 | v0 = 864;
11 | do
12 |     matrix[--v0] = 15;
13 | while ( v0 );

```

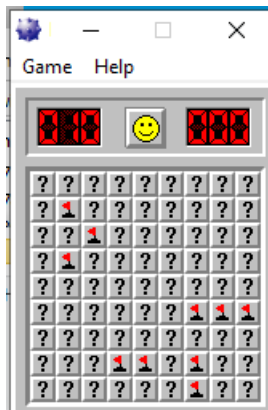
Once again, I manipulated the program file with x32dbg:

•	01002ED5	B8 60030000	mov eax,360
•	01002EDA	48	dec eax
•	01002ED8	C680 40530001 0F	mov byte ptr ds:[eax+1005340],F
•	01002EE2	75 F6	jne winmine.1002EDA
•	01002FF4	8B0D 34530001	mov ecx,dword ptr ds:[1005334]

F = 15 (the initial value of any cell)

•	01002ED2	C2 0000	ret
•	01002ED5	B8 60030000	mov eax,360
•	01002EDA	48	dec eax
•	01002ED8	C680 40530001 0D	mov byte ptr ds:[eax+1005340],D
•	01002EE2	75 F6	jne cracked_winmine_v2.1002EDA
•	01002EE4	8B0D 34530001	mov ecx,dword ptr ds:[1005334]

D = 13 (the value symbolizing the question mark)



- edit the “Fastest Mine Sweepers” to show your name for all levels of difficulty and set the number of seconds to the minimum;

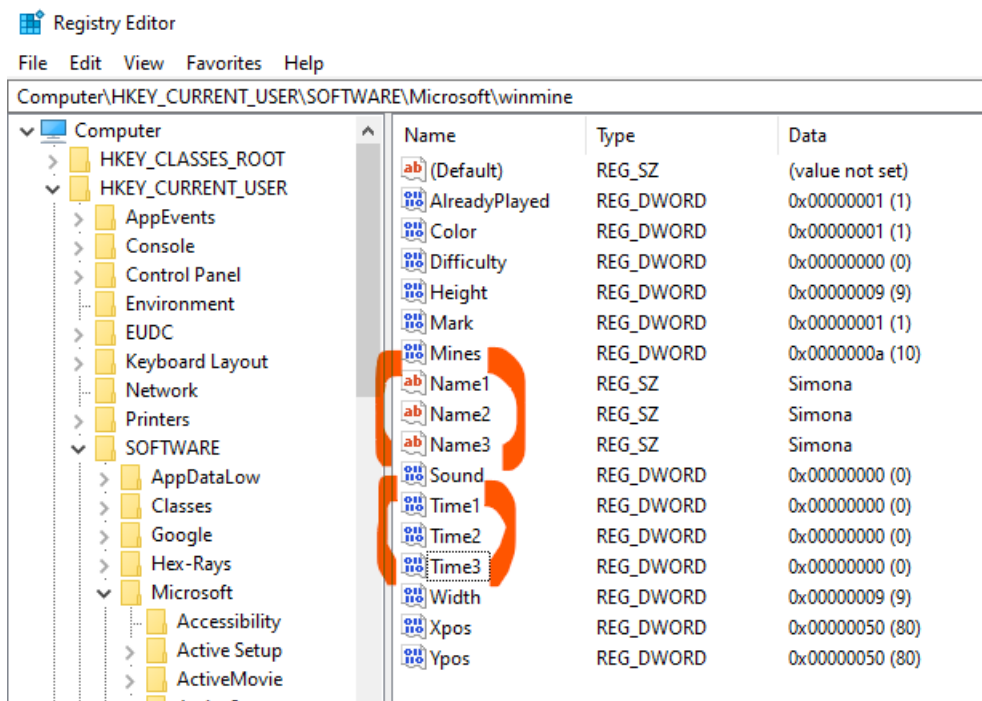
I figured out that the best scores must be saved somewhere outside the executable and that the most obvious solution was the Windows Registry. With this in mind, I verified the Imports tab in IDA to search for win32 APIs that deal with the registry (library ADVAPI32). In this manner, I managed to find the below call to RegCreateKey.

```

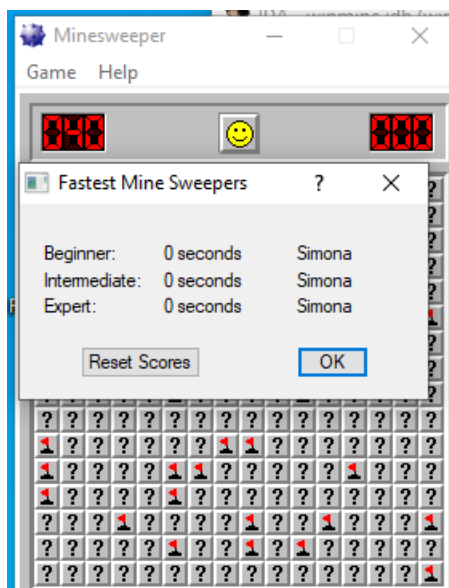
25
26 RegCreateKeyExM(HKEY_CURRENT_USER, L"Software\\Microsoft\\winmine", 0, 0, 0, 0x20019u, 0, &hKey, &dwDisposition);
27 uValue = read_reg(v0, 2u, 9, 9, 25);
28 dword_1005338 = uValue;
29 dword_10056AC = read_reg(v1, 3u, 9, 9, 30);
30 dword_1005334 = dword_10056AC;
31 LOWORD(dword_10056A0) = read_reg(v2, 0, 0, 0, 3);
32 dword_10056A4 = read_reg(v3, 1u, 10, 10, 999);
33 *(_DWORD *)&X = read_reg(v4, 4u, 80, 0, 1024);
34 *(_DWORD *)&Y = read_reg(v5, 5u, 80, 0, 1024);
35 *(_DWORD *)&dword_10056B8 = read_reg(v6, 6u, 0, 0, 3);
36 *(_DWORD *)&dword_10056BC = read_reg(v7, 7u, 1, 0, 1);
37 dword_10056C0 = read_reg(v8, 9u, 0, 0, 1);
38 dword_10056C4 = read_reg(v9, 8u, 0, 0, 2);
39 *(_DWORD *)&dword_10056CC = read_reg(v10, 0xBu, 999, 0, 999);
40 *(_DWORD *)&dword_10056D0 = read_reg(v11, 0xDu, 999, 0, 999);
41 *(_DWORD *)&dword_10056D4 = read_reg(v12, 0xFu, 999, 0, 999);
42 sub_1002B80(v13, 12, (LPBYTE)&ReturnedString);
43 sub_1002B80(v14, 14, (LPBYTE)&word_1005718);
44 sub_1002B80(v15, 16, (LPBYTE)&word_1005758);
45 v16 = GetDesktopWindow();
46 v17 = GetDC(v16);
47 v18 = v17;
48 v19 = GetDeviceCaps(v17, 24);
49 *(_DWORD *)&dword_10056C8 = read_reg((void *) (v19 != 2), 0xAu, v19 != 2, 0, 1);
50 v20 = GetDesktopWindow();
51 ReleaseDC(v20, v18);
52 if ( *(_DWORD *)&dword_10056B8 == 3 )
53     *(_DWORD *)&dword_10056B8 = sub_10038C2();
54 return RegCloseKey(hKey);
55 }

```

Following the path indicated by the call (Computer\HKEY_CURRENT_USER\SOFTWARE\Microsoft\winmine), I was able to confirm that my intuition was correct and get to the best scores location, which I further altered as required.



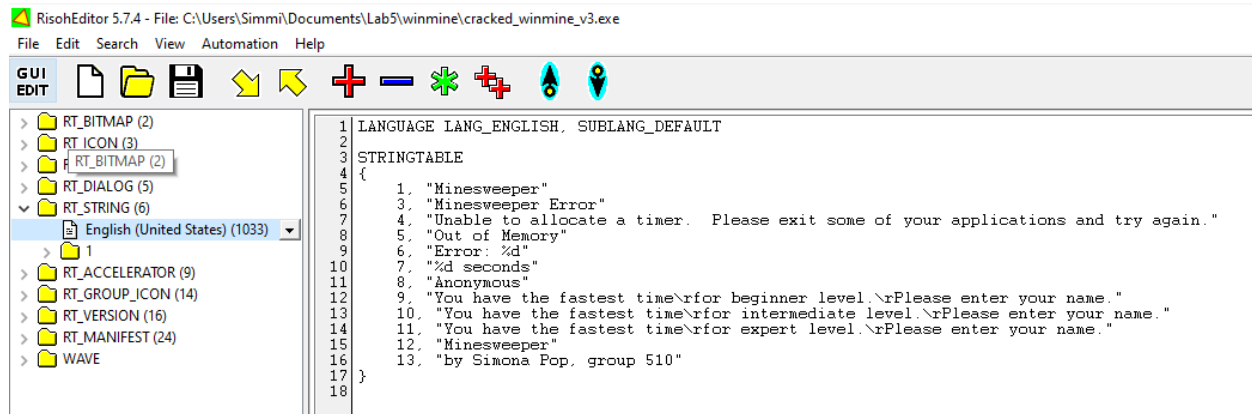
The result:



- edit the “About MineSweeper” window to show your name as the creator of the game.

Since the About window contains the executable icon, I assumed that the string including the authors of the game is stored in the same place as the icon. Per further research, I concluded that this place is equivalent to the rodata segment of an ELF file. In Windows, this segment is called Resources.

Therefore, I downloaded a resource editor named RisoHEditor and modified the resources of the winmine exec.



The result:

