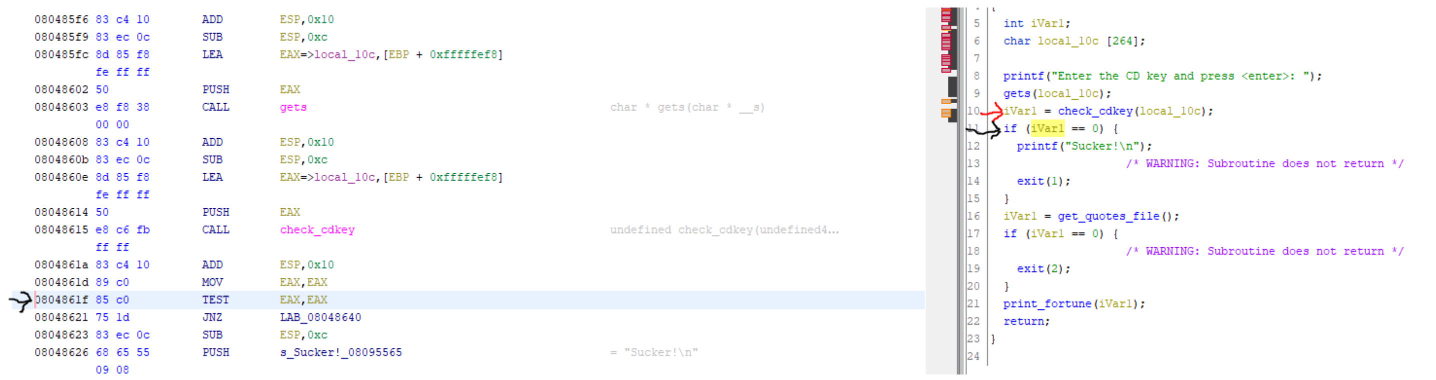
Matt Christensen (mrc621)

November 5, 2021

CS 465 (001) – Clift, Frederic M

Project #8 – Extracting Secrets

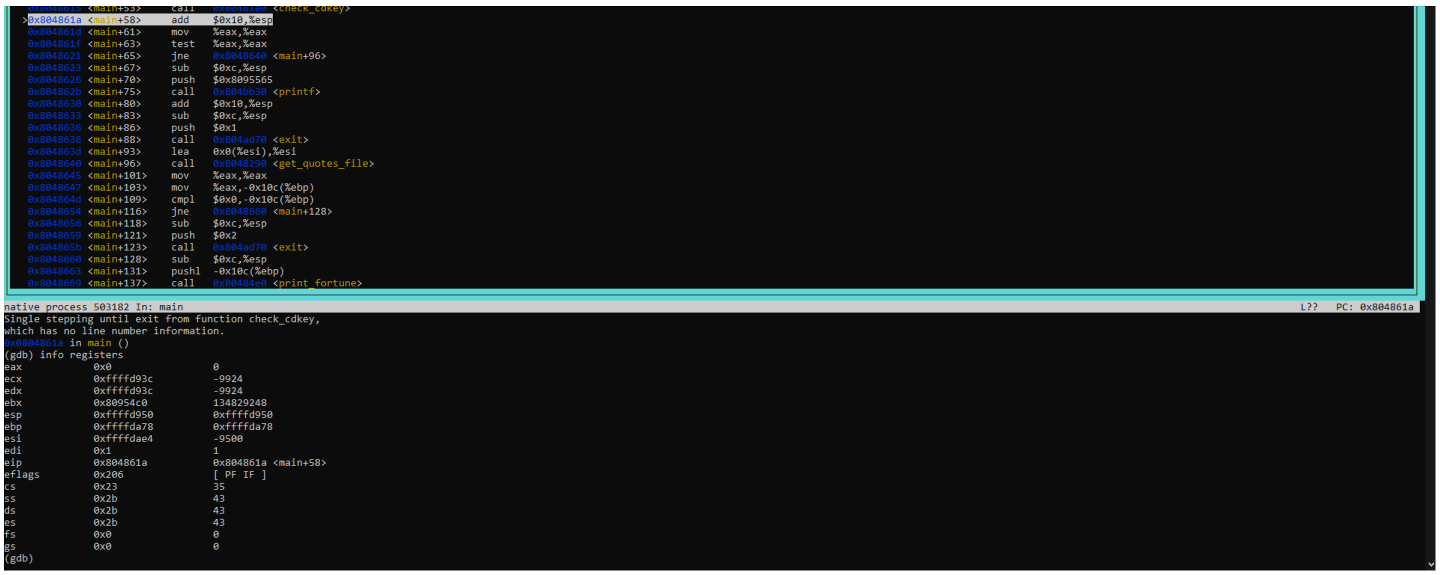
The first step of my process was the look at the program in a RE tool called Ghidra. I saw that the printing of the secret was dependent on the return value of *check_cdkey()*.



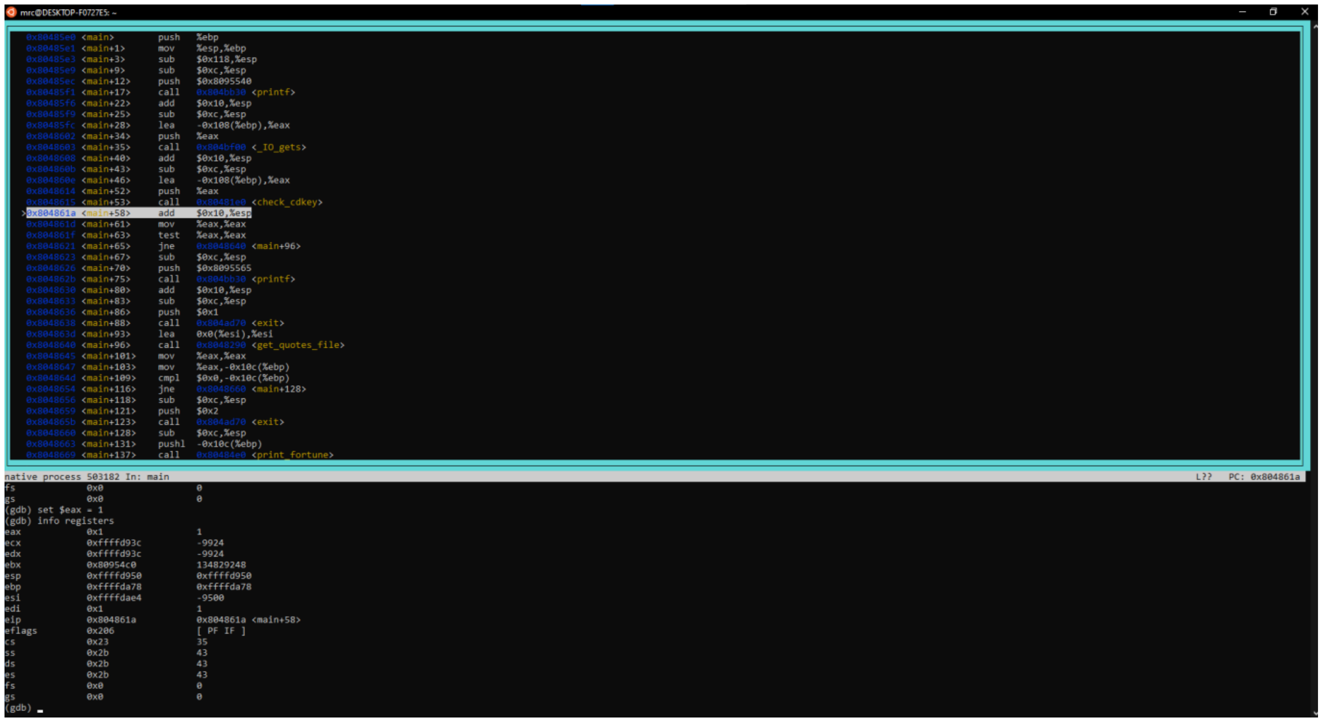
```
080485f6 83 c4 10     ADD     ESP,0x10
080485f9 83 ec 0c     SUB     ESP,0xc
080485fc 8d 85 f8     LEA     EAX=>local_10c,[EBP + 0xfffffef8]
         fe ff ff
08048602 50           PUSH    EAX
08048603 e8 f8 38     CALL    gets                            char * gets(char * __s)
         00 00
08048608 83 c4 10     ADD     ESP,0x10
0804860b 83 ec 0c     SUB     ESP,0xc
0804860e 8d 85 f8     LEA     EAX=>local_10c,[EBP + 0xfffffef8]
         fe ff ff
08048614 50           PUSH    EAX
08048615 e8 c6 fb     CALL    check_cdkey                     undefined check_cdkey(undefined4...
         ff ff
0804861a 83 c4 10     ADD     ESP,0x10
0804861d 89 c0        MOV     EAX,EAX
0804861f 85 c0        TEST    EAX,EAX
08048621 75 1d        JNZ     LAB_08048640
08048623 83 ec 0c     SUB     ESP,0xc
08048626 68 65 55     PUSH    s_Sucker!_08095565              = "Sucker!\n"
         09 08
```

```
  int iVar1;
  char local_10c [264];

  printf("Enter the CD key and press <enter>: ");
  gets(local_10c);
  iVar1 = check_cdkey(local_10c);
  if (iVar1 == 0) {
    printf("Sucker!\n");
                    /* WARNING: Subroutine does not return */
    exit(1);
  }
  iVar1 = get_quotes_file();
  if (iVar1 == 0) {
                    /* WARNING: Subroutine does not return */
    exit(2);
  }
  print_fortune(iVar1);
  return;
}
```

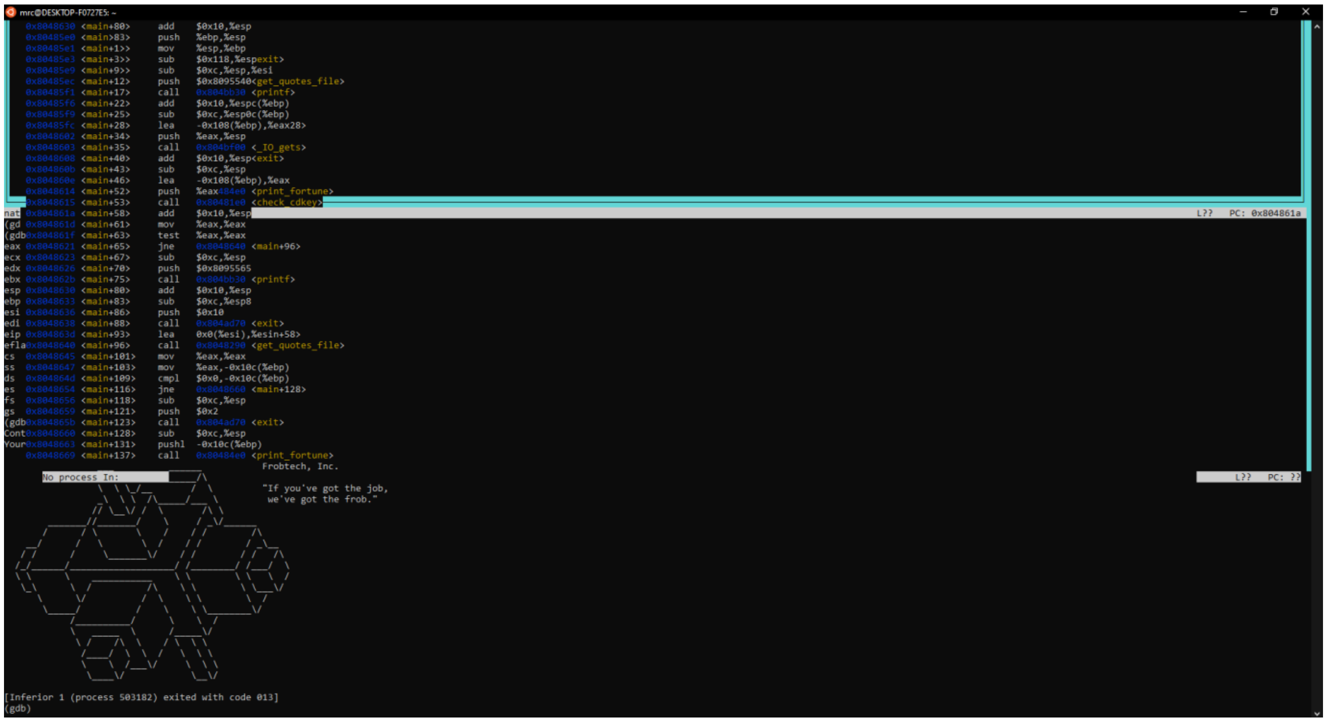
To test this, I ran to program in GDB and broke after the function returned. I then simply set the EAX register to 1 (using the '*set $eax = 1*' command) and the program printed out a fortune (**requirement 1**).



```
   0x8048615 <main+53>     call   0x80481e0 <check_cdkey>
  >0x804861a <main+58>     add    $0x10,%esp
   0x804861d <main+61>     mov    %eax,%eax
   0x804861f <main+63>     test   %eax,%eax
   0x8048621 <main+65>     jne    0x8048640 <main+96>
   0x8048623 <main+67>     sub    $0xc,%esp
   0x8048626 <main+70>     push   $0x8095565
   0x804862b <main+75>     call   0x804bb30 <printf>
   0x8048630 <main+80>     add    $0x10,%esp
   0x8048633 <main+83>     sub    $0xc,%esp
   0x8048636 <main+86>     push   $0x1
   0x8048638 <main+88>     call   0x804ad70 <exit>
   0x804863d <main+93>     lea    0x0(%esi),%esi
   0x8048640 <main+96>     call   0x8048290 <get_quotes_file>
   0x8048645 <main+101>    mov    %eax,%eax
   0x8048647 <main+103>    mov    %eax,-0x10c(%ebp)
   0x804864d <main+109>    cmpl   $0x0,-0x10c(%ebp)
   0x8048654 <main+116>    jne    0x8048660 <main+128>
   0x8048656 <main+118>    sub    $0xc,%esp
   0x8048659 <main+121>    push   $0x2
   0x804865b <main+123>    call   0x804ad70 <exit>
   0x8048660 <main+128>    sub    $0xc,%esp
   0x8048663 <main+131>    pushl  -0x10c(%ebp)
   0x8048669 <main+137>    call   0x80484e0 <print_fortune>
native process 503182 In: main                                           L??   PC: 0x804861a
Single stepping until exit from function check_cdkey,
which has no line number information.
0x0804861a in main ()
(gdb) info registers
eax            0x0                 0
ecx            0xffffd93c          -9924
edx            0xffffd93c          -9924
ebx            0x80954c0           134829248
esp            0xffffd950          0xffffd950
ebp            0xffffda78          0xffffda78
esi            0xffffdae4          -9500
edi            0x1                 1
eip            0x804861a           0x804861a <main+58>
eflags         0x206               [ PF IF ]
cs             0x23                35
ss             0x2b                43
ds             0x2b                43
es             0x2b                43
fs             0x0                 0
gs             0x0                 0
(gdb)
```

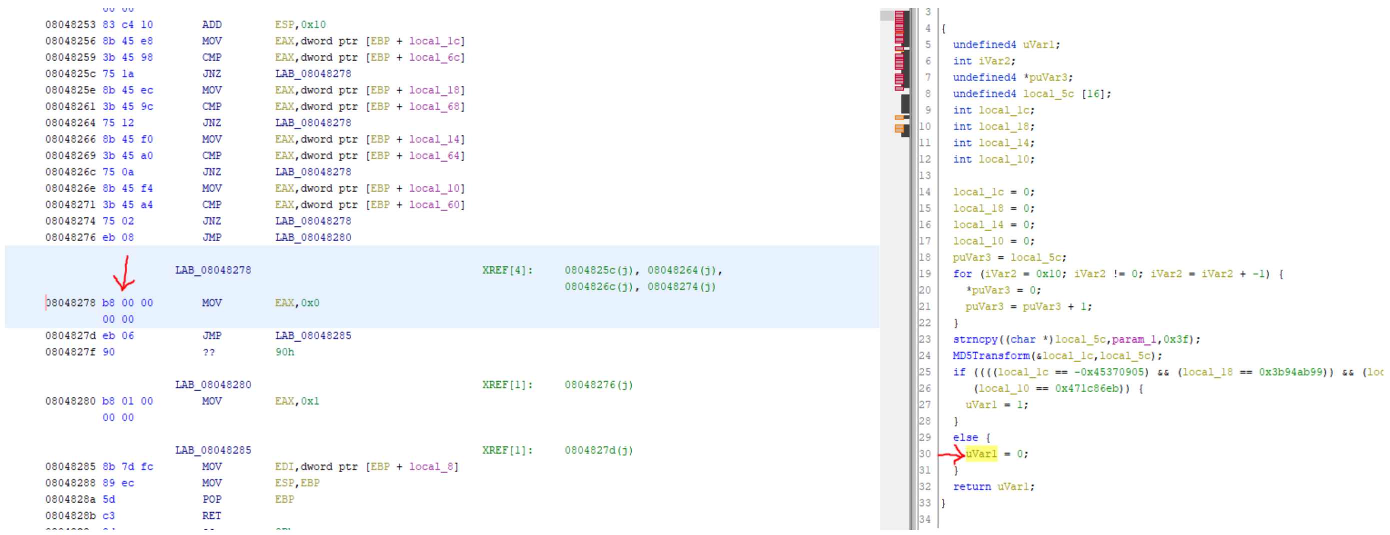


```
mrc@DESKTOP-F0727E5: ~
   0x80485e0 <main>        push   %ebp
   0x80485e1 <main+1>      mov    %esp,%ebp
   0x80485e3 <main+3>      sub    $0x118,%esp
   0x80485e9 <main+9>      sub    $0xc,%esp
   0x80485ec <main+12>     push   $0x8095540
   0x80485f1 <main+17>     call   0x804bb30 <printf>
   0x80485f6 <main+22>     add    $0x10,%esp
   0x80485f9 <main+25>     sub    $0xc,%esp
   0x80485fc <main+28>     lea    -0x108(%ebp),%eax
   0x8048602 <main+34>     push   %eax
   0x8048603 <main+35>     call   0x804bf00 <_IO_gets>
   0x8048608 <main+40>     add    $0x10,%esp
   0x804860b <main+43>     sub    $0xc,%esp
   0x804860e <main+46>     lea    -0x108(%ebp),%eax
   0x8048614 <main+52>     push   %eax
   0x8048615 <main+53>     call   0x80481e0 <check_cdkey>
  >0x804861a <main+58>     add    $0x10,%esp
   0x804861d <main+61>     mov    %eax,%eax
   0x804861f <main+63>     test   %eax,%eax
   0x8048621 <main+65>     jne    0x8048640 <main+96>
   0x8048623 <main+67>     sub    $0xc,%esp
   0x8048626 <main+70>     push   $0x8095565
   0x804862b <main+75>     call   0x804bb30 <printf>
   0x8048630 <main+80>     add    $0x10,%esp
   0x8048633 <main+83>     sub    $0xc,%esp
   0x8048636 <main+86>     push   $0x1
   0x8048638 <main+88>     call   0x804ad70 <exit>
   0x804863d <main+93>     lea    0x0(%esi),%esi
   0x8048640 <main+96>     call   0x8048290 <get_quotes_file>
   0x8048645 <main+101>    mov    %eax,%eax
   0x8048647 <main+103>    mov    %eax,-0x10c(%ebp)
   0x804864d <main+109>    cmpl   $0x0,-0x10c(%ebp)
   0x8048654 <main+116>    jne    0x8048660 <main+128>
   0x8048656 <main+118>    sub    $0xc,%esp
   0x8048659 <main+121>    push   $0x2
   0x804865b <main+123>    call   0x804ad70 <exit>
   0x8048660 <main+128>    sub    $0xc,%esp
   0x8048663 <main+131>    pushl  -0x10c(%ebp)
   0x8048669 <main+137>    call   0x80484e0 <print_fortune>
native process 503182 In: main                                                      L??   PC: 0x804861a
fs             0x0                 0
gs             0x0                 0
(gdb) set $eax = 1
(gdb) info registers
eax            0x1                 1
ecx            0xffffd93c          -9924
edx            0xffffd93c          -9924
ebx            0x80954c0           134829248
esp            0xffffd950          0xffffd950
ebp            0xffffda78          0xffffda78
esi            0xffffdae4          -9500
edi            0x1                 1
eip            0x804861a           0x804861a <main+58>
eflags         0x206               [ PF IF ]
cs             0x23                35
ss             0x2b                43
ds             0x2b                43
es             0x2b                43
fs             0x0                 0
gs             0x0                 0
(gdb) _
```



```
mrc@DESKTOP-F0727E5: ~
   0x8048630 <main+80>     add    $0x10,%esp
   0x80485e0 <main>83>     push   %ebp,%esp
   0x80485e1 <main+1>>     mov    %esp,%ebp
   0x80485e3 <main+3>>     sub    $0x118,%espexit>
   0x80485e9 <main+9>>     sub    $0xc,%esp,%esi
   0x80485ec <main+12>     push   $0x8095540<get_quotes_file>
   0x80485f1 <main+17>     call   0x804bb30 <printf>
   0x80485f6 <main+22>     add    $0x10,%espc(%ebp)
   0x80485f9 <main+25>     sub    $0xc,%esp0c(%ebp)
   0x80485fc <main+28>     lea    -0x108(%ebp),%eax28>
   0x8048602 <main+34>     push   %eax,%esp
   0x8048603 <main+35>     call   0x804bf00 <_IO_gets>
   0x8048608 <main+40>     add    $0x10,%esp<exit>
   0x804860b <main+43>     sub    $0xc,%esp
   0x804860e <main+46>     lea    -0x108(%ebp),%eax
   0x8048614 <main+52>     push   %eax484e0 <print_fortune>
   0x8048615 <main+53>     call   0x80481e0 <check_cdkey>
nat 0x804861a <main+58>    add    $0x10,%esp                                        L??   PC: 0x804861a
(gd 0x804861d <main+61>    mov    %eax,%eax
(gdb0x804861f <main+63>    test   %eax,%eax
eax 0x8048621 <main+65>    jne    0x8048640 <main+96>
ecx 0x8048623 <main+67>    sub    $0xc,%esp
edx 0x8048626 <main+70>    push   $0x8095565
ebx 0x804862b <main+75>    call   0x804bb30 <printf>
esp 0x8048630 <main+80>    add    $0x10,%esp
ebp 0x8048633 <main+83>    sub    $0xc,%esp8
esi 0x8048636 <main+86>    push   $0x10
edi 0x8048638 <main+88>    call   0x804ad70 <exit>
eip 0x804863d <main+93>    lea    0x0(%esi),%esin+58>
efla0x8048640 <main+96>    call   0x8048290 <get_quotes_file>
cs  0x8048645 <main+101>   mov    %eax,%eax
ss  0x8048647 <main+103>   mov    %eax,-0x10c(%ebp)
ds  0x804864d <main+109>   cmpl   $0x0,-0x10c(%ebp)
es  0x8048654 <main+116>   jne    0x8048660 <main+128>
fs  0x8048656 <main+118>   sub    $0xc,%esp
gs  0x8048659 <main+121>   push   $0x2
(gdb0x804865b <main+123>   call   0x804ad70 <exit>
Cont0x8048660 <main+128>   sub    $0xc,%esp
Your0x8048663 <main+131>   pushl  -0x10c(%ebp)
    0x8048669 <main+137>   call   0x80484e0 <print_fortune>
                                     Frobtech, Inc.
     No process In:                                                                 L??   PC: ??
                                     "If you've got the job,
                                      we've got the frob."

[Inferior 1 (process 503182) exited with code 013]
(gdb)
```

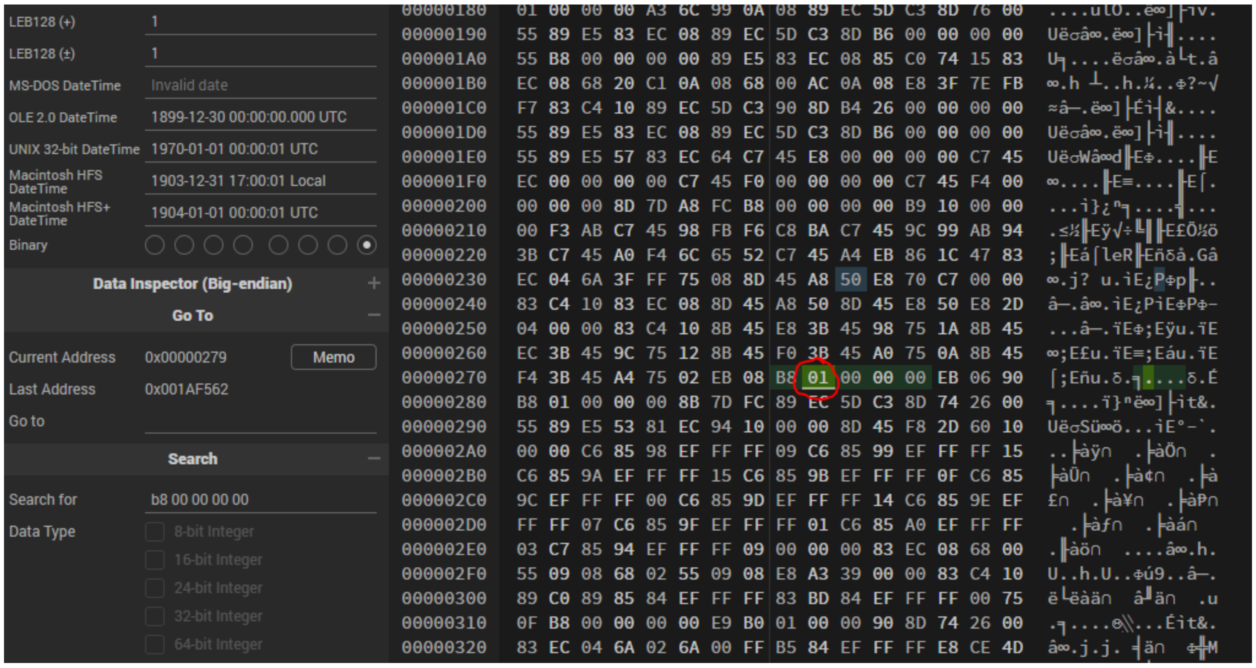
In order edit the program to bypass the cdkey mechanism, I simply found (in Ghidra) that the *check_cdkey* function returned a one if the password was verified and a zero if the password was invalid.



```
08048253 83 c4 10       ADD     ESP,0x10
08048256 8b 45 e8       MOV     EAX,dword ptr [EBP + local_1c]
08048259 3b 45 98       CMP     EAX,dword ptr [EBP + local_6c]
0804825c 75 1a          JNZ     LAB_08048278
0804825e 8b 45 ec       MOV     EAX,dword ptr [EBP + local_18]
08048261 3b 45 9c       CMP     EAX,dword ptr [EBP + local_68]
08048264 75 12          JNZ     LAB_08048278
08048266 8b 45 f0       MOV     EAX,dword ptr [EBP + local_14]
08048269 3b 45 a0       CMP     EAX,dword ptr [EBP + local_64]
0804826c 75 0a          JNZ     LAB_08048278
0804826e 8b 45 f4       MOV     EAX,dword ptr [EBP + local_10]
08048271 3b 45 a4       CMP     EAX,dword ptr [EBP + local_60]
08048274 75 02          JNZ     LAB_08048278
08048276 eb 08          JMP     LAB_08048280

                    LAB_08048278                    XREF[4]:   0804825c(j), 08048264(j),
                                                               0804826c(j), 08048274(j)
08048278 b8 00 00       MOV     EAX,0x0
         00 00
0804827d eb 06          JMP     LAB_08048285
0804827f 90             ??      90h

                    LAB_08048280                    XREF[1]:   08048276(j)
08048280 b8 01 00       MOV     EAX,0x1
         00 00

                    LAB_08048285                    XREF[1]:   0804827d(j)
08048285 8b 7d fc       MOV     EDI,dword ptr [EBP + local_8]
08048288 89 ec          MOV     ESP,EBP
0804828a 5d             POP     EBP
0804828b c3             RET
```

```
{
  undefined4 uVar1;
  int iVar2;
  undefined4 *puVar3;
  undefined4 local_5c [16];
  int local_1c;
  int local_18;
  int local_14;
  int local_10;

  local_1c = 0;
  local_18 = 0;
  local_14 = 0;
  local_10 = 0;
  puVar3 = local_5c;
  for (iVar2 = 0x10; iVar2 != 0; iVar2 = iVar2 + -1) {
    *puVar3 = 0;
    puVar3 = puVar3 + 1;
  }
  strncpy((char *)local_5c,param_1,0x3f);
  MD5Transform(&local_1c,local_5c);
  if ((((local_1c == -0x45370905) && (local_18 == 0x3b94ab99)) && (loc
     (local_10 == 0x471c86eb)) {
    uVar1 = 1;
  }
  else {
    uVar1 = 0;
  }
  return uVar1;
}
```

So I simply found the instruction responsible for setting the return value to zero (*b8 00 00 00 00* – which moves zero into the EAX register), opened the *fortune_static* binary in a hex editor, found the instruction, and changed it to move a one into the EAX register (with the instruction – *B8 01 00 00 00*).



LEB128 (+)	1
LEB128 (±)	1
MS-DOS DateTime	Invalid date
OLE 2.0 DateTime	1899-12-30 00:00:00.000 UTC
UNIX 32-bit DateTime	1970-01-01 00:00:01 UTC
Macintosh HFS DateTime	1903-12-31 17:00:01 Local
Macintosh HFS+ DateTime	1904-01-01 00:00:01 UTC
Binary	○ ○ ○ ○ ○ ○ ○ ●

Data Inspector (Big-endian)

Go To

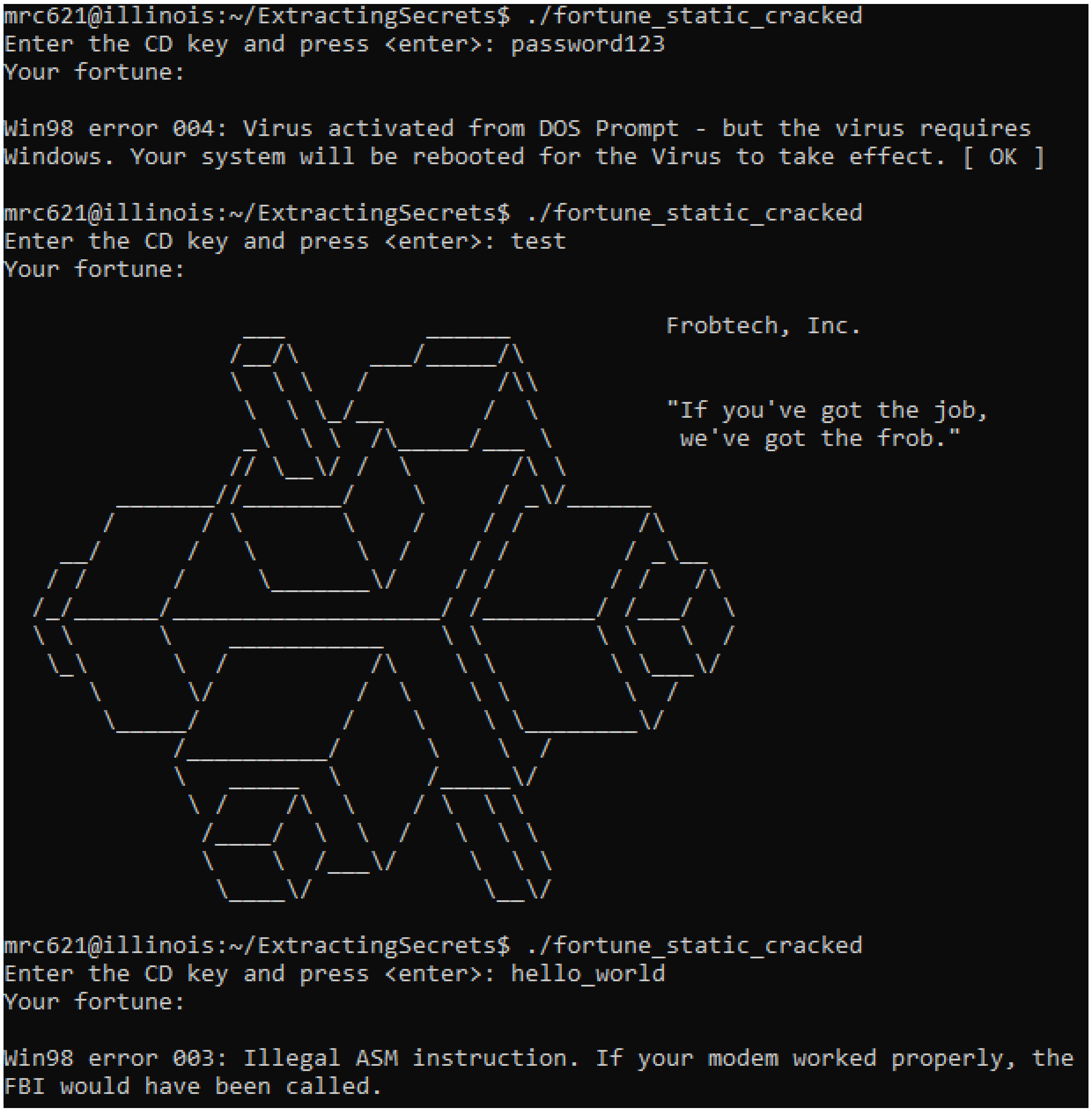
Current Address	0x00000279
Last Address	0x001AF562
Go to	

Search

Search for	b8 00 00 00 00
Data Type	☐ 8-bit Integer
	☐ 16-bit Integer
	☐ 24-bit Integer
	☐ 32-bit Integer
	☐ 64-bit Integer

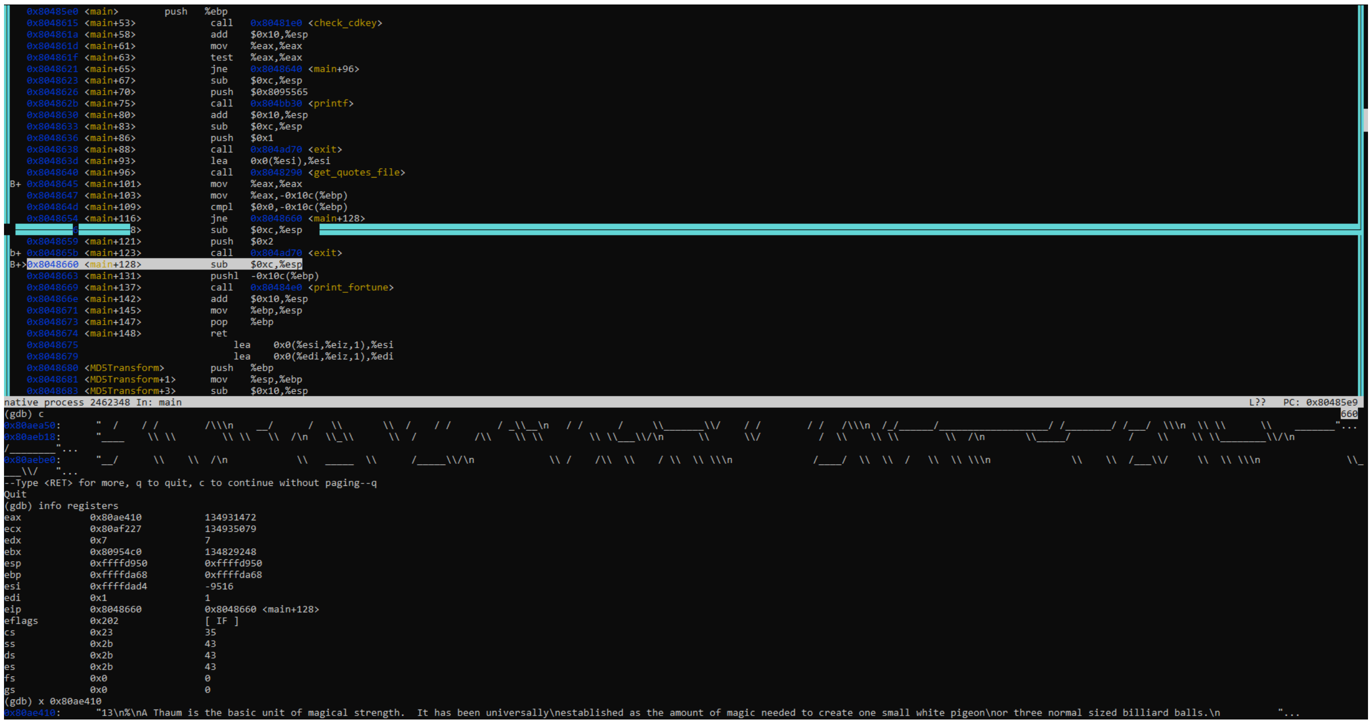
```
00000180  01 00 00 00 A3 6C 99 0A 08 89 EC 5D C3 8D 76 00   ....ulO..ë∞]├ìv.
00000190  55 89 E5 83 EC 08 89 EC 5D C3 8D B6 00 00 00 00   Uëσâ∞.ë∞]├ì╢....
000001A0  55 B8 00 00 00 00 89 E5 83 EC 08 85 C0 74 15 83   U╕....ëσâ∞.à└t.â
000001B0  EC 08 68 20 C1 0A 08 68 00 AC 0A 08 E8 3F 7E FB   ∞.h ┴..h.¼..Φ?~√
000001C0  F7 83 C4 10 89 EC 5D C3 90 8D B4 26 00 00 00 00   ≈â─.ë∞]├Éì┤&....
000001D0  55 89 E5 83 EC 08 89 EC 5D C3 8D B6 00 00 00 00   Uëσâ∞.ë∞]├ì╢....
000001E0  55 89 E5 57 83 EC 64 C7 45 E8 00 00 00 00 C7 45   UëσWâ∞d╟EΦ....╟E
000001F0  EC 00 00 00 00 C7 45 F0 00 00 00 00 C7 45 F4 00   ∞....╟E≡....╟E⌠.
00000200  00 00 00 8D 7D A8 FC B8 00 00 00 00 B9 10 00 00   ...ì}¿ⁿ╕....╣...
00000210  00 F3 AB C7 45 98 FB F6 C8 BA C7 45 9C 99 AB 94   .≤½╟Eÿ√÷╚║╟E£Ö½ö
00000220  3B C7 45 A0 F4 6C 65 52 C7 45 A4 EB 86 1C 47 83   ;╟Eá⌠leR╟Eñδå.Gâ
00000230  EC 04 6A 3F FF 75 08 8D 45 A8 50 E8 70 C7 00 00   ∞.j? u.ìE¿PΦp╟..
00000240  83 C4 10 83 EC 08 8D 45 A8 50 8D 45 E8 50 E8 2D   â─.â∞.ìE¿PìEΦPΦ-
00000250  04 00 00 83 C4 10 8B 45 E8 3B 45 98 75 1A 8B 45   ...â─.ïEΦ;Eÿu.ïE
00000260  EC 3B 45 9C 75 12 8B 45 F0 3B 45 A0 75 0A 8B 45   ∞;E£u.ïE≡;Eáu.ïE
00000270  F4 3B 45 A4 75 02 EB 08 B8 01 00 00 00 EB 06 90   ⌠;Eñu.δ.╕....δ.É
00000280  B8 01 00 00 00 8B 7D FC 89 EC 5D C3 8D 74 26 00   ╕....ï}ⁿë∞]├ìt&.
00000290  55 89 E5 53 81 EC 94 10 00 00 8D 45 F8 2D 60 10   UëσSü∞ö...ìE°-`.
000002A0  00 00 C6 85 98 EF FF FF 09 C6 85 99 EF FF FF 15   ..╞àÿ∩  .╞àÖ∩  .
000002B0  C6 85 9A EF FF FF 15 C6 85 9B EF FF FF 0F C6 85   ╞àÜ∩  .╞à¢∩  .╞à
000002C0  9C EF FF FF 00 C6 85 9D EF FF FF 14 C6 85 9E EF   £∩  .╞à¥∩  .╞à₧∩
000002D0  FF FF 07 C6 85 9F EF FF FF 01 C6 85 A0 EF FF FF     .╞àƒ∩  .╞àá∩
000002E0  03 C7 85 94 EF FF FF 09 00 00 00 83 EC 08 68 00   .╟àö∩  ....â∞.h.
000002F0  55 09 08 68 02 55 09 08 E8 A3 39 00 00 83 C4 10   U..h.U..Φú9..â─.
00000300  89 C0 89 85 84 EF FF FF 83 BD 84 EF FF FF 00 75   ë└ëàä∩  â╜ä∩  .u
00000310  0F B8 00 00 00 00 E9 B0 01 00 00 90 8D 74 26 00   .╕....Θ░...Éìt&.
00000320  83 EC 04 6A 02 6A 00 FF B5 84 EF FF FF E8 CE 4D   â∞.j.j. ╡ä∩  Φ╬M
```

This new binary, when ran, would accept all passwords given and will always print a fortune (**requirement 2**).



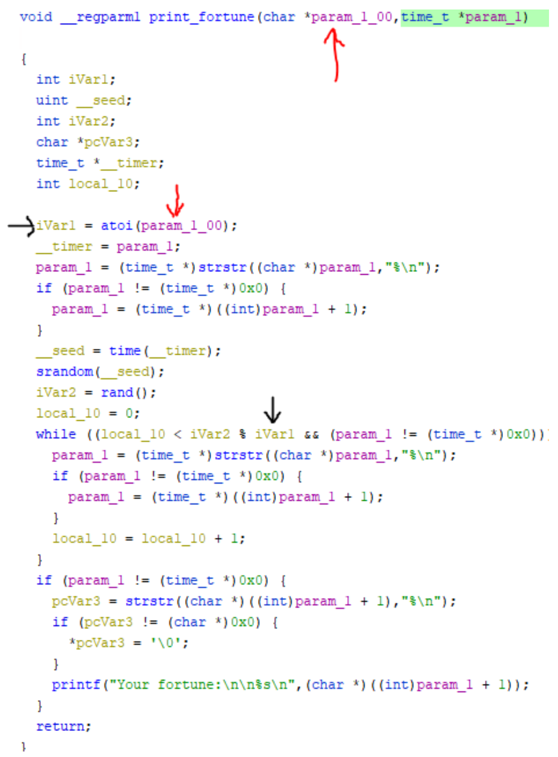
```
mrc621@illinois:~/ExtractingSecrets$ ./fortune_static_cracked
Enter the CD key and press <enter>: password123
Your fortune:

Win98 error 004: Virus activated from DOS Prompt - but the virus requires
Windows. Your system will be rebooted for the Virus to take effect. [ OK ]

mrc621@illinois:~/ExtractingSecrets$ ./fortune_static_cracked
Enter the CD key and press <enter>: test
Your fortune:

                 ___          ______           Frobtech, Inc.
                /__/\     ___/_____/\
                \  \ \   /         /\\
                 \  \ \_/__       /  \         "If you've got the job,
                 _\  \ \  /\_____/___ \         we've got the frob."
                // \__\/ /  \       /\ \
        _______//_______/    \     / _\/______
       /      / \       \    /    / /        /\
    __/      /   \       \  /    / /        / _\__
   / /      /     \_______\/    / /        / /   /\
  /_/______/___________________/ /________/ /___/  \
  \ \      \    ___________    \ \        \ \   \  /
   \_\      \  /          /\    \ \        \ \___\/
      \      \/          /  \    \ \        \  /
       \_____/          /    \    \ \________\/
            /__________/      \    \  /
            \   _____  \      /_____\/
             \ /    /\  \    / \  \ \
              /____/  \  \  /   \  \ \
              \    \  /___\/     \  \ \
               \____\/            \__\/

mrc621@illinois:~/ExtractingSecrets$ ./fortune_static_cracked
Enter the CD key and press <enter>: hello_world
Your fortune:

Win98 error 003: Illegal ASM instruction. If your modem worked properly, the
FBI would have been called.
```

Finally, we needed to extract all the possible secrets. I chose to run the cracked version in GDB and then break right after the call to *get_quotes_file*. I examined the registers and found the address 0x80ae410 in EAX. When examining that address, I got the number thirteen followed by what I assumed to be the first string:



```
   0x80485e0 <main>        push   %ebp
   0x8048615 <main+53>             call   0x80481e0 <check_cdkey>
   0x804861a <main+58>             add    $0x10,%esp
   0x804861d <main+61>             mov    %eax,%eax
   0x804861f <main+63>             test   %eax,%eax
   0x8048621 <main+65>             jne    0x8048640 <main+96>
   0x8048623 <main+67>             sub    $0xc,%esp
   0x8048626 <main+70>             push   $0x8095565
   0x804862b <main+75>             call   0x804bb30 <printf>
   0x8048630 <main+80>             add    $0x10,%esp
   0x8048633 <main+83>             sub    $0xc,%esp
   0x8048636 <main+86>             push   $0x1
   0x8048638 <main+88>             call   0x804ad70 <exit>
   0x804863d <main+93>             lea    0x0(%esi),%esi
   0x8048640 <main+96>             call   0x8048290 <get_quotes_file>
B+ 0x8048645 <main+101>            mov    %eax,%eax
   0x8048647 <main+103>            mov    %eax,-0x10c(%ebp)
   0x804864d <main+109>            cmpl   $0x0,-0x10c(%ebp)
   0x8048654 <main+116>            jne    0x8048660 <main+128>
           6          8>           sub    $0xc,%esp
   0x8048659 <main+121>            push   $0x2
b+ 0x804865b <main+123>            call   0x804ad70 <exit>
B+>0x8048660 <main+128>            sub    $0xc,%esp
   0x8048663 <main+131>            pushl  -0x10c(%ebp)
   0x8048669 <main+137>            call   0x80484e0 <print_fortune>
   0x804866e <main+142>            add    $0x10,%esp
   0x8048671 <main+145>            mov    %ebp,%esp
   0x8048673 <main+147>            pop    %ebp
   0x8048674 <main+148>            ret
   0x8048675                           lea    0x0(%esi,%eiz,1),%esi
   0x8048679                           lea    0x0(%edi,%eiz,1),%edi
   0x8048680 <MD5Transform>        push   %ebp
   0x8048681 <MD5Transform+1>      mov    %esp,%ebp
   0x8048683 <MD5Transform+3>      sub    $0x10,%esp
native process 2462348 In: main                                                                    L??   PC: 0x80485e9
(gdb) c                                                                                                          660
0x80aea50:      "  /    / /        /\\\n    __/      /   \\       \\  /    / /        / _\\__\n   / /      /     \\\_______\\/    / /        / /   /\\\n  /_/______/__________________/ /________/ /___/  \\\n  \\ \\      \\    _______"...
0x80aeb18:      "____    \\ \\        \\ \\   \\  /\n   \\_\\      \\  /          /\\    \\ \\        \\ \\___\\/\n      \\      \\/         /  \\    \\ \\       \\  /\n      \\_____/          /    \\    \\ \\________\\/\n
/________"...
0x80aebe0:      "__/      \\    \\  /\n          \\   _____  \\      /_____\\/\n           \\ /    /\\  \\    / \\  \\ \\\n            /____/  \\  \\  /   \\  \\ \\\n            \\    \\  /___\\/    \\  \\ \\\n            \\_
___\\/   "...
--Type <RET> for more, q to quit, c to continue without paging--q
Quit
(gdb) info registers
eax            0x80ae410           134931472
ecx            0x80af227           134935079
edx            0x7                 7
ebx            0x80954c0           134829248
esp            0xffffd950          0xffffd950
ebp            0xffffda68          0xffffda68
esi            0xffffdad4          -9516
edi            0x1                 1
eip            0x8048660           0x8048660 <main+128>
eflags         0x202               [ IF ]
cs             0x23                35
ss             0x2b                43
ds             0x2b                43
es             0x2b                43
fs             0x0                 0
gs             0x0                 0
(gdb) x 0x80ae410
0x80ae410:      "13\n%\nA Thaum is the basic unit of magical strength.  It has been universally\nestablished as the amount of magic needed to create one small white pigeon\nor three normal sized billiard balls.\n        "...
```

This matches with what we see in Ghirdra, as it appears to number of strings is first used as a way of indexing and moding the random seed:



```c
void __regparm1 print_fortune(char *param_1_00,time_t *param_1)

{
  int iVar1;
  uint __seed;
  int iVar2;
  char *pcVar3;
  time_t *__timer;
  int local_10;

  iVar1 = atoi(param_1_00);
  __timer = param_1;
  param_1 = (time_t *)strstr((char *)param_1,"%\n");
  if (param_1 != (time_t *)0x0) {
    param_1 = (time_t *)((int)param_1 + 1);
  }
  __seed = time(__timer);
  srandom(__seed);
  iVar2 = rand();
  local_10 = 0;
  while ((local_10 < iVar2 % iVar1 && (param_1 != (time_t *)0x0))
    param_1 = (time_t *)strstr((char *)param_1,"%\n");
    if (param_1 != (time_t *)0x0) {
      param_1 = (time_t *)((int)param_1 + 1);
    }
    local_10 = local_10 + 1;
  }
  if (param_1 != (time_t *)0x0) {
    pcVar3 = strstr((char *)((int)param_1 + 1),"%\n");
    if (pcVar3 != (char *)0x0) {
      *pcVar3 = '\0';
    }
    printf("Your fortune:\n\n%s\n",(char *)((int)param_1 + 1));
  }
  return;
}
```

I proceeded to dump the memory info a file (using logging and the "*x /3600 0x80ae410*" command) to find all the secret fortunes: (**requirement 3**)

A Thaum is the basic unit of magical strength. It has been universally
established as the amount of magic needed to create one small white pigeon
or three normal sized billiard balls.
-- Terry Pratchett, "The Light Fantastic"

"A wizard cannot do everything; a fact most magicians are reticent to admit,
let alone discuss with prospective clients. Still, the fact remains that
there are certain objects, and people, that are, for one reason or another,
completely immune to any direct magical spell. It is for this group of
beings that the magician learns the subtleties of using indirect spells.
It also does no harm, in dealing with these matters, to carry a large club
near your person at all times."
-- The Teachings of Ebenezum, Volume VIII

"Do not meddle in the affairs of wizards, for you are crunchy and good
with ketchup."

Rincewind had generally been considered by his tutors to be a natural wizard
in the same way that fish are natural mountaineers. He probably would have
been thrown out of Unseen University anyway--he couldn't remember spells and
smoking made him feel ill.
-- Terry Pratchett, "The Light Fantastic"

```
                 ___          ______           Frobtech, Inc.
                /__/\     ___/_____/\
                \  \ \   /         /\\
                 \  \ \_/__       /  \         "If you've got the job,
                 _\  \ \  /\_____/___ \         we've got the frob."
                // \__\/ /  \       /\ \
        _______//_______/    \     / _\/______
       /      / \       \    /    / /        /\
    __/      /   \       \  /    / /        / _\__
   / /      /     \_______\/    / /        / /   /\
  /_/______/___________________/ /________/ /___/  \
  \ \      \    ___________    \ \        \ \   \  /
   \_\      \  /          /\    \ \        \ \___\/
      \      \/          /  \    \ \        \  /
       \_____/          /    \    \ \________\/
            /__________/      \    \  /
            \   _____  \      /_____\/
             \ /    /\  \    / \  \ \
              /____/  \  \  /   \  \ \
              \    \  /___\/     \  \ \
               \____\/            \__\/
```

Win98 error 001: Unexpected condition: booted without crashing.

Win98 error 002: Insufficient diskspace. You need at least 300 GB free memory.

Win98 error 003: Illegal ASM instruction. If your modem worked properly, the
FBI would have been called.

Win NT error 001: Error recording error codes. All further errors not
displayed.

Win98 error 004: Virus activated from DOS Prompt - but the virus requires
Windows. Your system will be rebooted for the Virus to take effect. [OK]

Win98 error 005: Mouse not found. Click left mouse button on ok to continue.

Win98 error 006: Keyboard not found. Press F1 to continue.

(1) Office employees will daily sweep the floors, dust the furniture, shelves, and showcases.
(2) Each day fill lamps, clean chimneys, and trim wicks. Wash the windows once a week.
(3) Each clerk will bring a bucket of water and a scuttle of coal for the day's business.
(4) Make your pens carefully. You may whittle nibs to your individual taste.
(5) This office will open at 7 a.m. and close at 8 p.m. except on the Sabbath, on which day we will remain closed. Each employee is expected to spend the Sabbath by attending church and contributing liberally to the cause of the Lord.

-- "Office Worker's Guide", New England Carriage Works, 1872