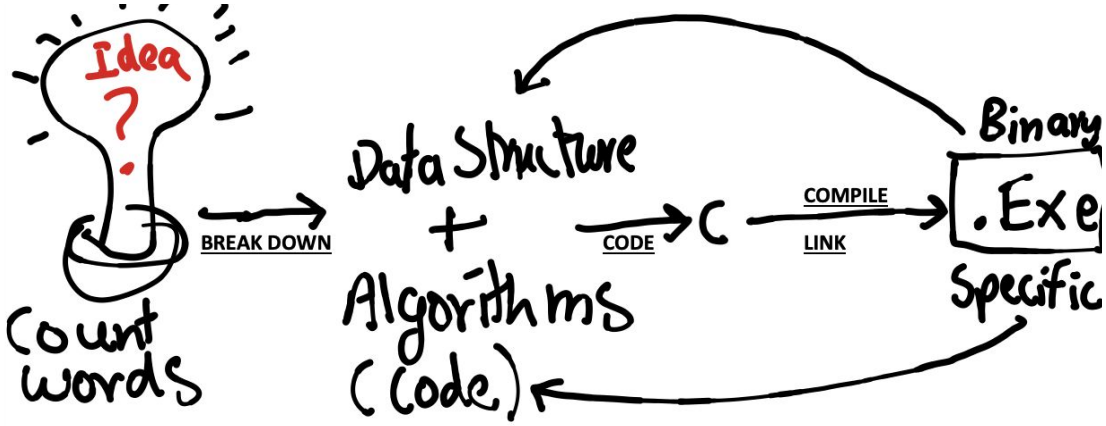
esreveR gnireenignE

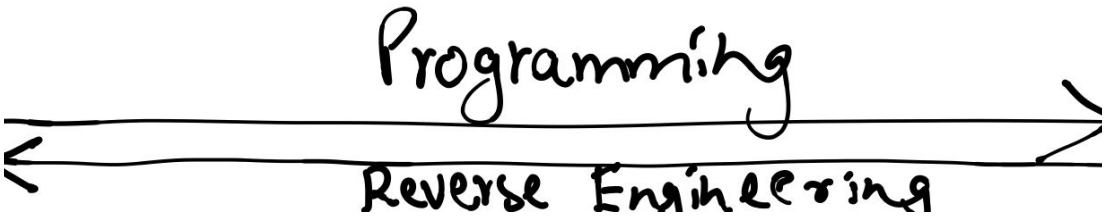
Bhavik Goplani

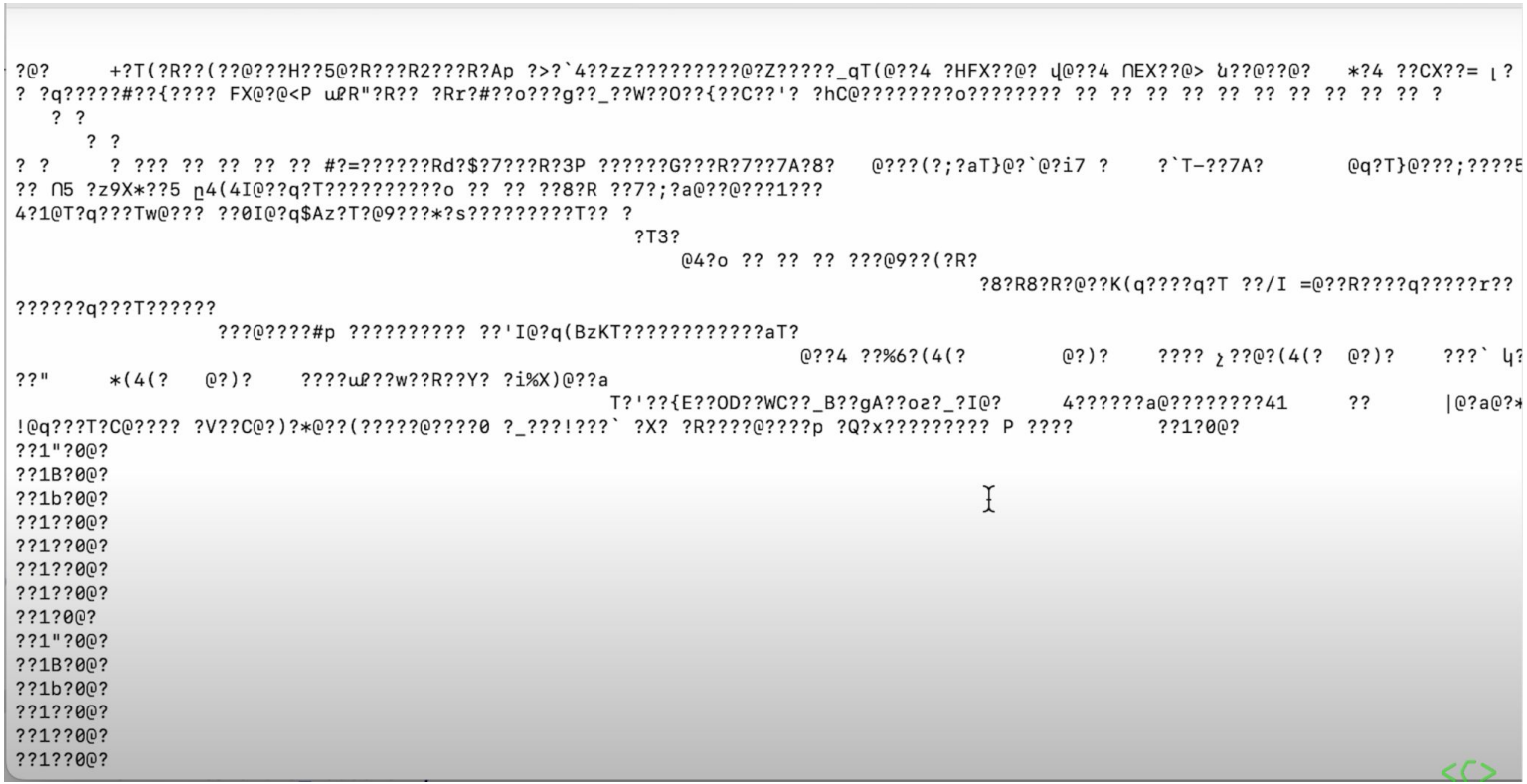
What is it?

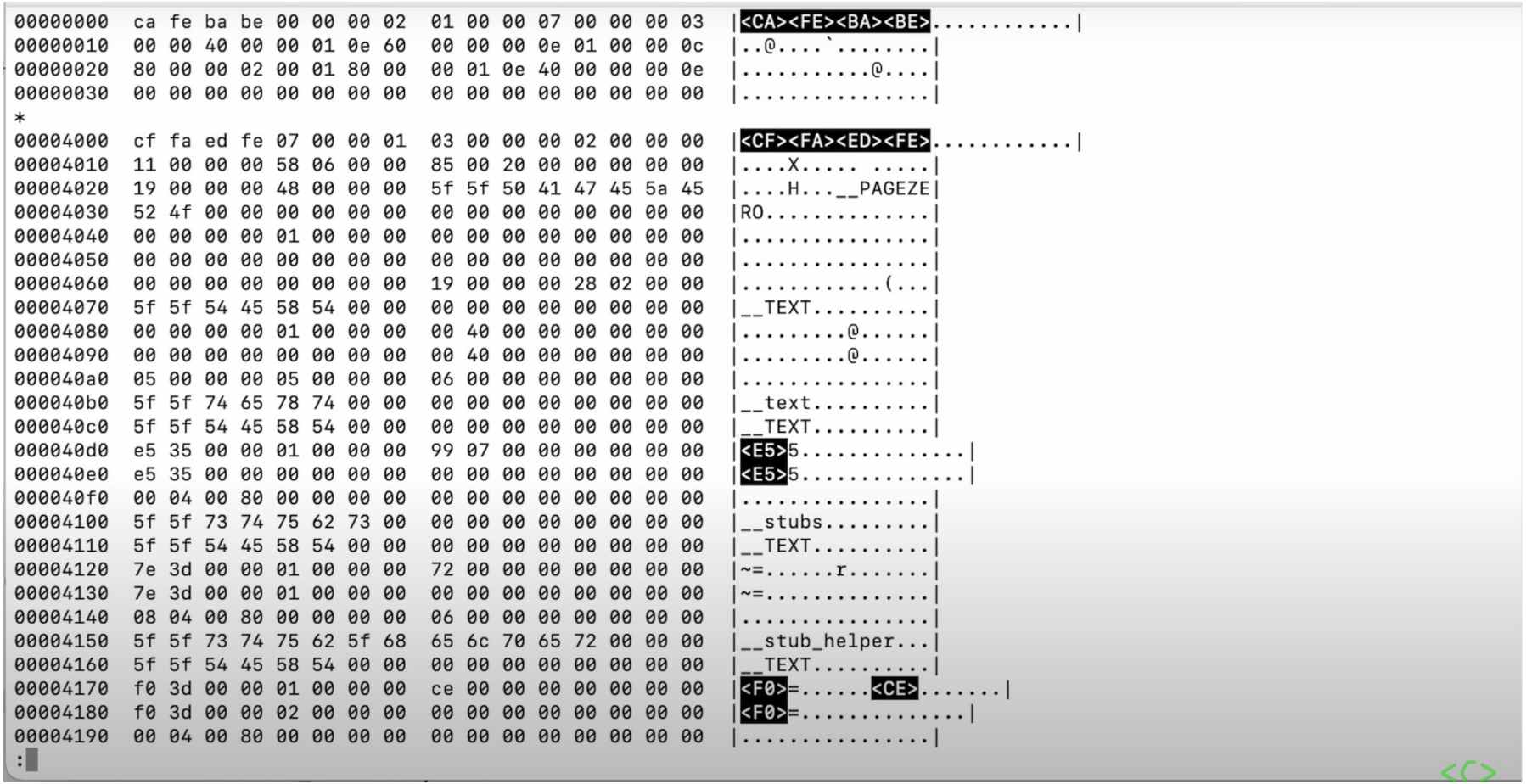


Hello 1 World! 2 This 3 is 4 my 5 first 6 Reverse 7 Engineering 8 code. 9

A word is a run of characters that aren't spaces.
space is ASCII 32,9,10,13...




?@? +?T(?R??(??@???H??5@?R???R2???R?Ap ?>?`4??zz??????????@?Z?????_qT(@??4 ?HFX??@? Ц@??4 ΠEX??@> ὐ??@??@? *?4 ??CX??= ⌊?
? ?q?????#??{???? FX@?@<P ɯℓR"?R?? ?Rr?#??o???g??_??W??O??{??C??'? ?hC@????????o???????? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?
? ?
? ?
? ? ? ??? ?? ?? ?? ?? #?=??????Rd?$?7???R?3P ??????G???R?7??7A?8? @???(?;?aT}@?`@?i7 ?  ?`T−??7A? @q?T}@???;????5
?? Π5 ?z9X*??5 ⊓4(4I@??q?T??????????o ?? ?? ??8?R ??7?;?a@??@???1???
4?1@T?q???Tw@??? ??0I@?q$Az?T?@9???*?s?????????T?? ?
?T3?
@4?o ?? ?? ?? ???@9??(?R?
?8?R8?R?@??K(q????q?T ??/I =@??R????q?????r??
??????q???T??????
???@?????#p ?????????? ??'I@?q(BzKT????????????aT?
@??4 ??%6?(4(? @?)? ???? ¿??@?(4(? @?)? ???` Ц?
??" *(4(? @?)? ????ɯℓ??w??R??Y? ?i%X)@??a
T?'??{E??OD??WC??_B??gA??oƨ?_?I@? 4??????a@????????41 ?? |@?a@?*
!@q???T?C@???? ?V??C@?)?*@??(?????@????0 ?_???!???` ?X? ?R????@????p ?Q?x????????? P ???? ??1?0@?
??1"?0@?
??1B?0@?
??1b?0@?
??1??0@?
??1??0@?
??1??0@?
??1??0@?
??1?0@?
??1"?0@?
??1B?0@?
??1b?0@?
??1??0@?
??1??0@?
??1??0@?


00000000 ca fe ba be 00 00 00 02 01 00 00 07 00 00 00 03 |<CA><FE><BA><BE>............|
00000010 00 00 40 00 00 01 0e 60 00 00 00 0e 01 00 00 0c |..@....`........|
00000020 80 00 00 02 00 01 80 00 00 01 0e 40 00 00 00 0e |...........@....|
00000030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |................|
*
00004000 cf fa ed fe 07 00 00 01 03 00 00 00 02 00 00 00 |<CF><FA><ED><FE>............|
00004010 11 00 00 00 58 06 00 00 85 00 20 00 00 00 00 00 |....X..... .....|
00004020 19 00 00 00 48 00 00 00 5f 5f 50 41 47 45 5a 45 |....H...__PAGEZE|
00004030 52 4f 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |RO..............|
00004040 00 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00 |................|
00004050 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |................|
00004060 00 00 00 00 00 00 00 00 19 00 00 00 28 02 00 00 |............(...|
00004070 5f 5f 54 45 58 54 00 00 00 00 00 00 00 00 00 00 |__TEXT..........|
00004080 00 00 00 00 01 00 00 00 00 40 00 00 00 00 00 00 |.........@......|
00004090 00 00 00 00 00 00 00 00 00 40 00 00 00 00 00 00 |.........@......|
000040a0 05 00 00 00 05 00 00 00 06 00 00 00 00 00 00 00 |................|
000040b0 5f 5f 74 65 78 74 00 00 00 00 00 00 00 00 00 00 |__text..........|
000040c0 5f 5f 54 45 58 54 00 00 00 00 00 00 00 00 00 00 |__TEXT..........|
000040d0 e5 35 00 00 01 00 00 00 99 07 00 00 00 00 00 00 |<E5>5............|
000040e0 e5 35 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |<E5>5............|
000040f0 00 04 00 80 00 00 00 00 00 00 00 00 00 00 00 00 |................|
00004100 5f 5f 73 74 75 62 73 00 00 00 00 00 00 00 00 00 |__stubs.........|
00004110 5f 5f 54 45 58 54 00 00 00 00 00 00 00 00 00 00 |__TEXT..........|
00004120 7e 3d 00 00 01 00 00 00 72 00 00 00 00 00 00 00 |~=......r.......|
00004130 7e 3d 00 00 01 00 00 00 00 00 00 00 00 00 00 00 |~=..............|
00004140 08 04 00 80 00 00 00 00 06 00 00 00 00 00 00 00 |................|
00004150 5f 5f 73 74 75 62 5f 68 65 6c 70 65 72 00 00 00 |__stub_helper...|
00004160 5f 5f 54 45 58 54 00 00 00 00 00 00 00 00 00 00 |__TEXT..........|
00004170 f0 3d 00 00 01 00 00 00 ce 00 00 00 00 00 00 00 |<F0>=......<CE>.......|
00004180 f0 3d 00 00 02 00 00 00 00 00 00 00 00 00 00 00 |<F0>=............|
00004190 00 04 00 80 00 00 00 00 00 00 00 00 00 00 00 00 |................|
:

3 basic steps of reverse-engineering

1. Information extraction
The original object or design is studied, and information about it is extracted.

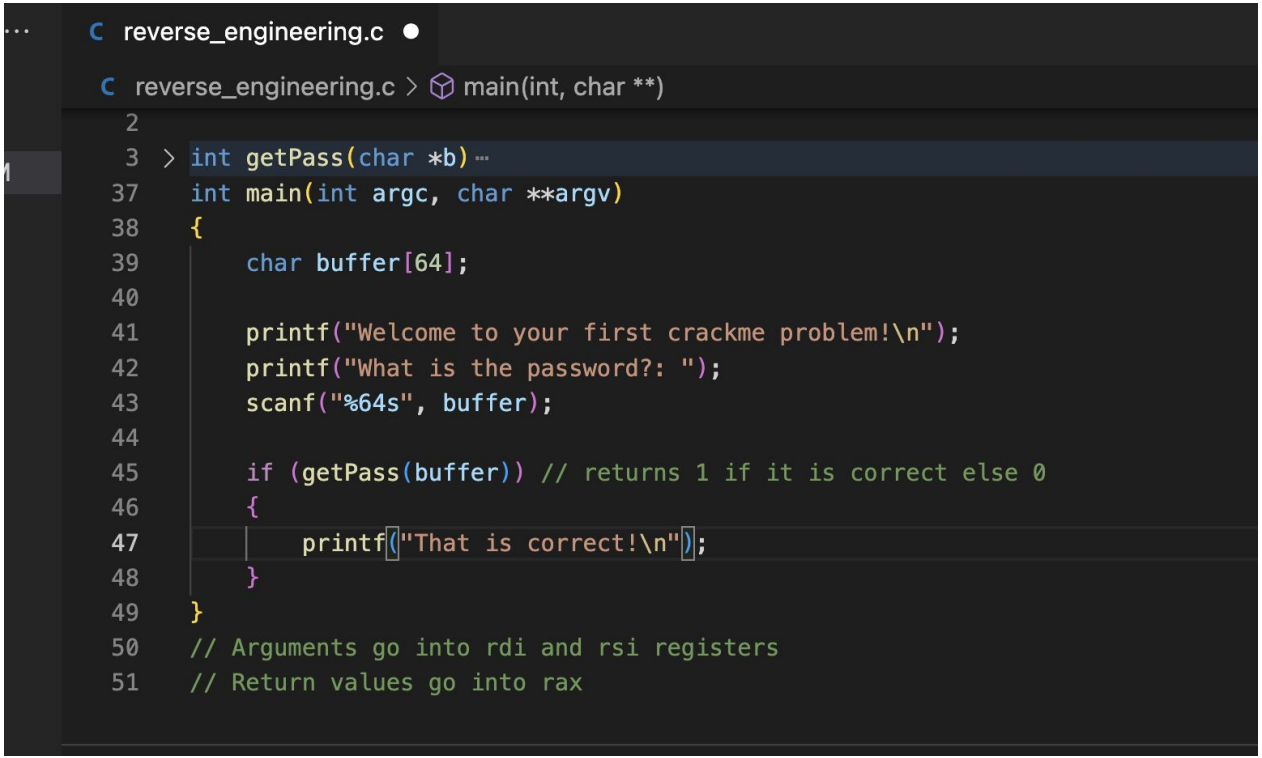
2. Modeling
The information collected is abstracted into a conceptual model.

3. Review
The model is tested in different contexts to determine if it was successfully reverse-engineered.

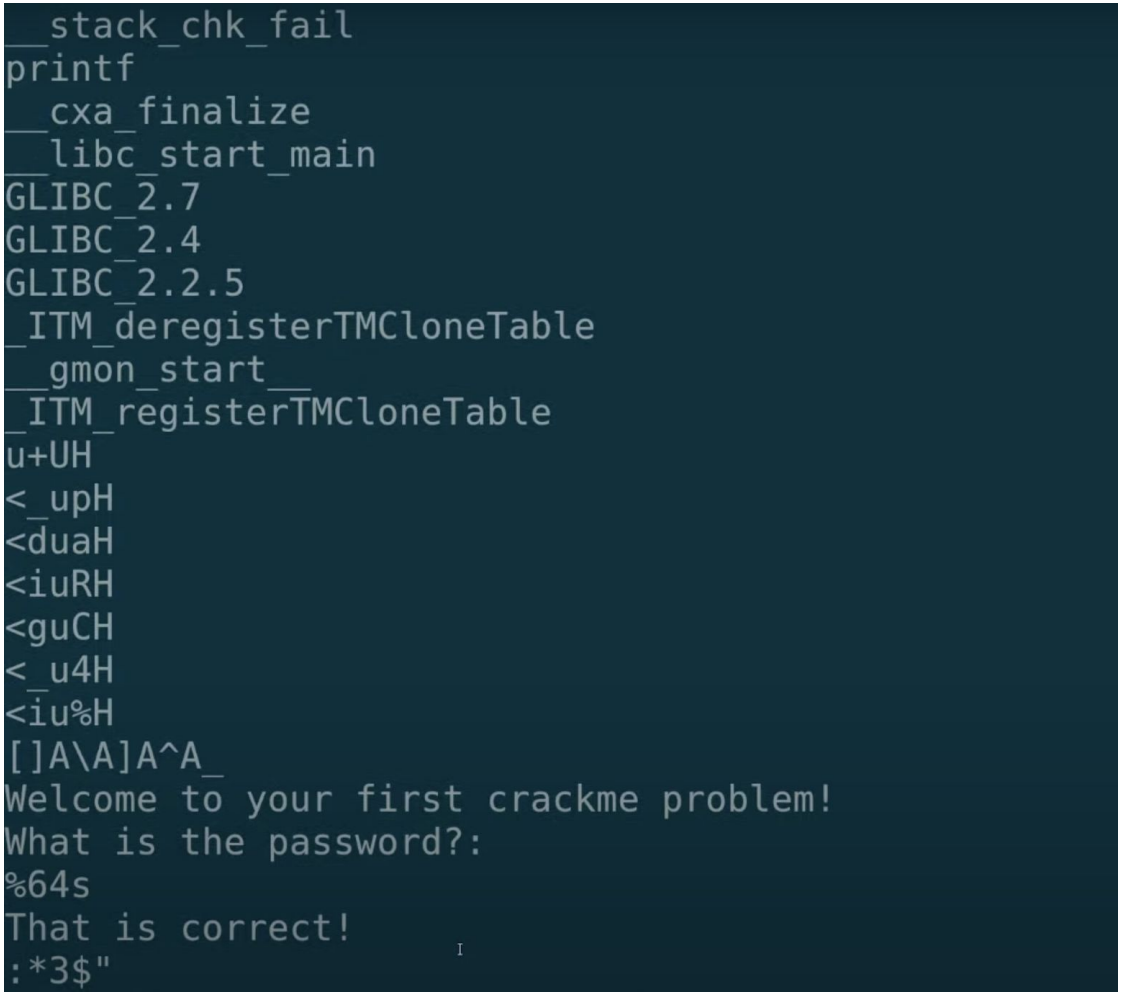
Popular Uses

- Phoenix Technologies Ltd., which in the mid-1980s wanted to produce a BIOS for PCs that would be compatible with the IBM PC's proprietary BIOS.
- Network Security Assessments at companies.
- White Box Reverse Engineering - often used during beta testing.
- Using CAD to reconstruct the object as a 3D model.
- NSA’s Ghidra - open source reverse engineering tool

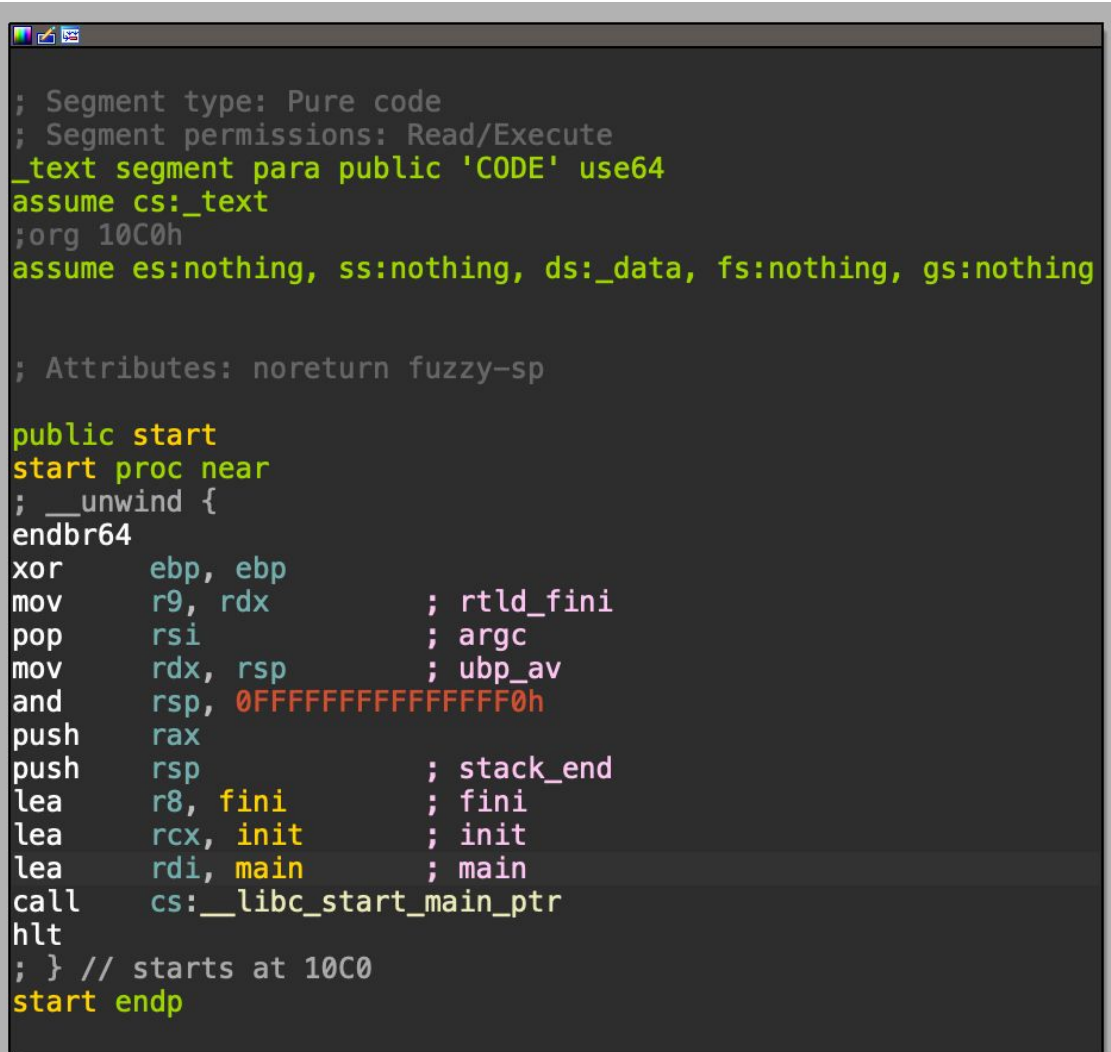
Implementation



```
00002bf0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00002c00: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00002c10: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00002c20: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00002c30: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00002c40: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00002c50: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00002c60: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00002c70: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00002c80: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00002c90: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00002ca0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00002cb0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00002cc0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00002cd0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00002ce0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00002cf0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00002d00: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00002d10: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00002d20: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00002d30: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00002d40: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00002d50: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00002d60: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00002d70: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00002d80: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00002d90: 0000 0000 0000 0000 0000 0000 0000 0000  ................
:
```

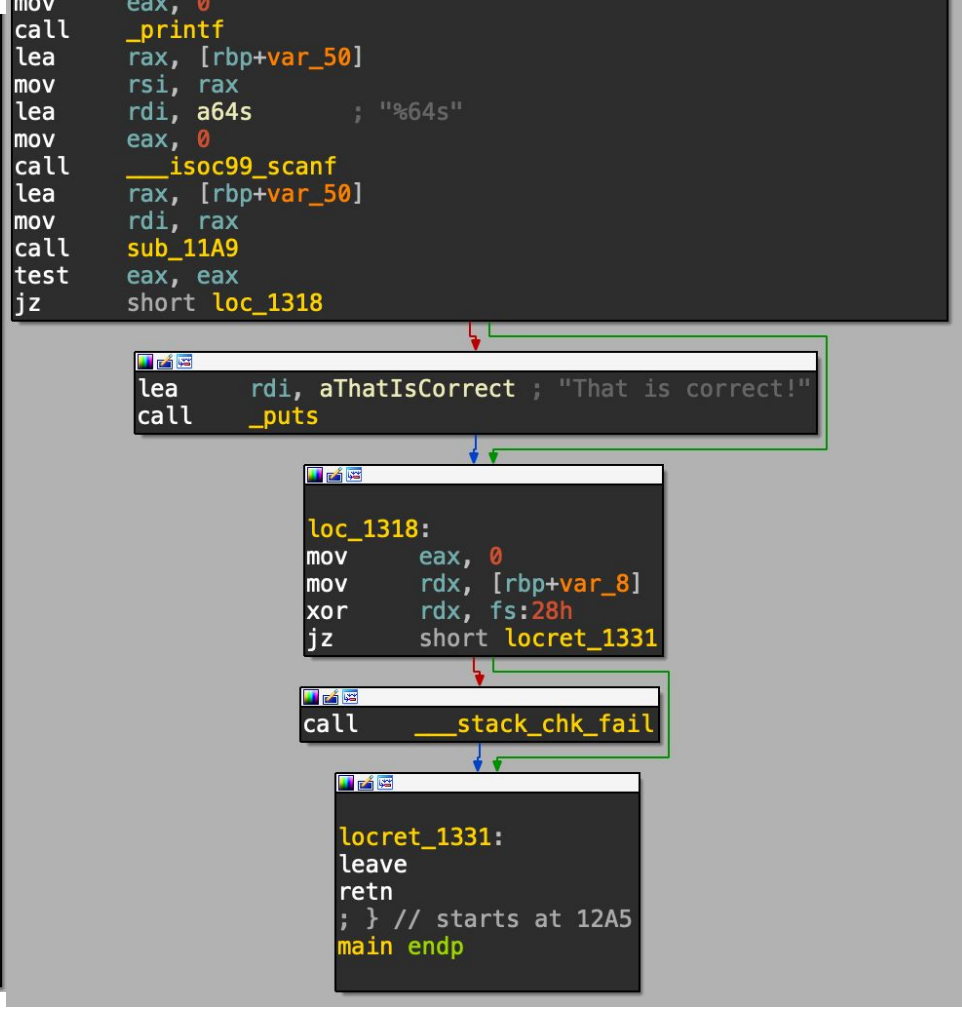

__stack_chk_fail
printf
__cxa_finalize
__libc_start_main
GLIBC_2.7
GLIBC_2.4
GLIBC_2.2.5
_ITM_deregisterTMCloneTable
__gmon_start__
_ITM_registerTMCloneTable
u+UH
<_upH
<duaH
<iuRH
<guCH
<_u4H
<iu%H
[]A\A]A^A_
Welcome to your first crackme problem!
What is the password?:
%64s
That is correct!
:*3$"

```
100003efc: e8 27 40 b9  ldr     w8, [sp, #36]
100003f00: e8 0f 00 b9  str     w8, [sp, #12]
100003f04: a9 83 5f f8  ldur    x9, [x29, #-8]
100003f08: 08 00 00 b0  adrp    x8, 0x100004000 <_main+0x8c>
100003f0c: 08 05 40 f9  ldr     x8, [x8, #8]
100003f10: 08 01 40 f9  ldr     x8, [x8]
100003f14: 08 01 09 eb  subs    x8, x8, x9
100003f18: 60 00 00 54  b.eq    0x100003f24 <_main+0xa4>
100003f1c: 01 00 00 14  b       0x100003f20 <_main+0xa0>
100003f20: 05 00 00 94  bl      0x100003f34 <_scanf+0x100003f34>
100003f24: e0 0f 40 b9  ldr     w0, [sp, #12]
100003f28: fd 7b 47 a9  ldp     x29, x30, [sp, #112]
100003f2c: ff 03 02 91  add     sp, sp, #128
100003f30: c0 03 5f d6  ret

Disassembly of section __TEXT,__stubs:

0000000100003f34 <__stubs>:
100003f34: 10 00 00 b0  adrp    x16, 0x100004000 <__stubs+0x4>
100003f38: 10 02 40 f9  ldr     x16, [x16]
100003f3c: 00 02 1f d6  br      x16
100003f40: 10 00 00 b0  adrp    x16, 0x100004000 <__stubs+0x10>
100003f44: 10 0a 40 f9  ldr     x16, [x16, #16]
100003f48: 00 02 1f d6  br      x16
100003f4c: 10 00 00 b0  adrp    x16, 0x100004000 <__stubs+0x1c>
100003f50: 10 0e 40 f9  ldr     x16, [x16, #24]
```



```
; Segment type: Pure code
; Segment permissions: Read/Execute
_text segment para public 'CODE' use64
assume cs:_text
;org 10C0h
assume es:nothing, ss:nothing, ds:_data, fs:nothing, gs:nothing


; Attributes: noreturn fuzzy-sp

public start
start proc near
; __unwind {
endbr64
xor     ebp, ebp
mov     r9, rdx         ; rtld_fini
pop     rsi             ; argc
mov     rdx, rsp        ; ubp_av
and     rsp, 0FFFFFFFFFFFFFFF0h
push    rax
push    rsp             ; stack_end
lea     r8, fini        ; fini
lea     rcx, init       ; init
lea     rdi, main       ; main
call    cs:__libc_start_main_ptr
hlt
; } // starts at 10C0
start endp
```

```
; Attributes: bp-based frame

; int __fastcall main(int, char **, char **)
main proc near

var_60= qword ptr -60h
var_54= dword ptr -54h
var_50= byte ptr -50h
var_8= qword ptr -8

; __unwind {
endbr64
push    rbp
mov     rbp, rsp
sub     rsp, 60h
mov     [rbp+var_54], edi
mov     [rbp+var_60], rsi
mov     rax, fs:28h
mov     [rbp+var_8], rax
xor     eax, eax
lea     rdi, s          ; "Welcome to your first crackme problem!"
call    _puts
lea     rdi, format     ; "What is the password?: "
mov     eax, 0
call    _printf
lea     rax, [rbp+var_50]
mov     rsi, rax
lea     rdi, a64s       ; "%64s"
mov     eax, 0
call    ___isoc99_scanf
lea     rax, [rbp+var_50]
mov     rdi, rax
call    sub_11A9
test    eax, eax
jz      short loc_1318
```




; Attributes: bp-based frame
sub_11A9 proc near
var_8= qword ptr -8
; __unwind {
endbr64
push rbp
mov rbp, rsp
mov [rbp+var_8], rdi
mov rax, [rbp+var_8]
movzx eax, byte ptr [rax]
cmp al, 63h ; 'c'
jnz loc_129E
mov rax, [rbp+var_8]
add rax, 1
movzx eax, byte ptr [rax]
cmp al, 61h ; 'a'
jnz loc_129E
mov rax, [rbp+var_8]
add rax, 2
movzx eax, byte ptr [rax]
cmp al, 6Eh ; 'n'
jnz loc_129E
mov rax, [rbp+var_8]
add rax, 3
mov rax, [rbp+var_8]
add rax, 3
movzx eax, byte ptr [rax]
cmp al, 5Fh ; '_'
jnz loc_129E
mov rax, [rbp+var_8]
add rax, 4
movzx eax, byte ptr [rax]
cmp al, 79h ; 'y'
jnz loc_129E
mov rax, [rbp+var_8]
add rax, 5
movzx eax, byte ptr [rax]
cmp al, 61h ; 'a'
jnz short loc_129E
mov rax, [rbp+var_8]
add rax, 6
movzx eax, byte ptr [rax]
cmp al, 5Fh ; '_'
jnz short loc_129E
mov rax, [rbp+var_8]
add rax, 7
movzx eax, byte ptr [rax]
cmp al, 64h ; 'd'
jnz short loc_129E
mov rax, [rbp+var_8]
mov rax, [rbp+var_8]
add rax, 8
movzx eax, byte ptr [rax]
cmp al, 69h ; 'i'
jnz short loc_129E
mov rax, [rbp+var_8]
add rax, 9
movzx eax, byte ptr [rax]
cmp al, 67h ; 'g'
jnz short loc_129E
mov rax, [rbp+var_8]
add rax, 0Ah
movzx eax, byte ptr [rax]
cmp al, 5Fh ; '_'
jnz short loc_129E
mov rax, [rbp+var_8]
add rax, 0Bh
movzx eax, byte ptr [rax]
cmp al, 69h ; 'i'
jnz short loc_129E
mov rax, [rbp+var_8]
add rax, 0Ch
movzx eax, byte ptr [rax]
cmp al, 74h ; 't'
jnz short loc_129E
mov rax, [rbp+var_8]
add rax, 0Bh
movzx eax, byte ptr [rax]
cmp al, 69h ; 'i'
jnz short loc_129E
mov rax, [rbp+var_8]
add rax, 0Ch
movzx eax, byte ptr [rax]
cmp al, 74h ; 't'
jnz short loc_129E
mov rax, [rbp+var_8]
add rax, 0Dh
movzx eax, byte ptr [rax]
cmp al, 3Fh ; '?'
jnz short loc_129E
loc_129E:
mov eax, 0
mov eax, 1
jmp short loc_12A3
loc_12A3:
pop rbp
retn
; } // starts at 11A9
sub_11A9 endp

```
int getPass(char *b)
{

    if (b[0] == 'c') {
    if (b[1] == 'a') {
    if (b[2] == 'n') {
    if (b[3] == '_') {
    if (b[4] == 'y') {
    if (b[5] == 'a') {
    if (b[6] == '_') {
    if (b[7] == 'd') {
    if (b[8] == 'i') {
    if (b[9] == 'g') {
    if (b[10] == '_') {
    if (b[11] == 'i') {
    if (b[12] == 't') {
    if (b[13] == '?') {
        return 1;
    }
```

```
➜  reverse-engineering ./reverse_engineering
Welcome to your first crackme problem!
What is the password?: can_ya_dig_it?
That is correct!
➜  reverse-engineering
```