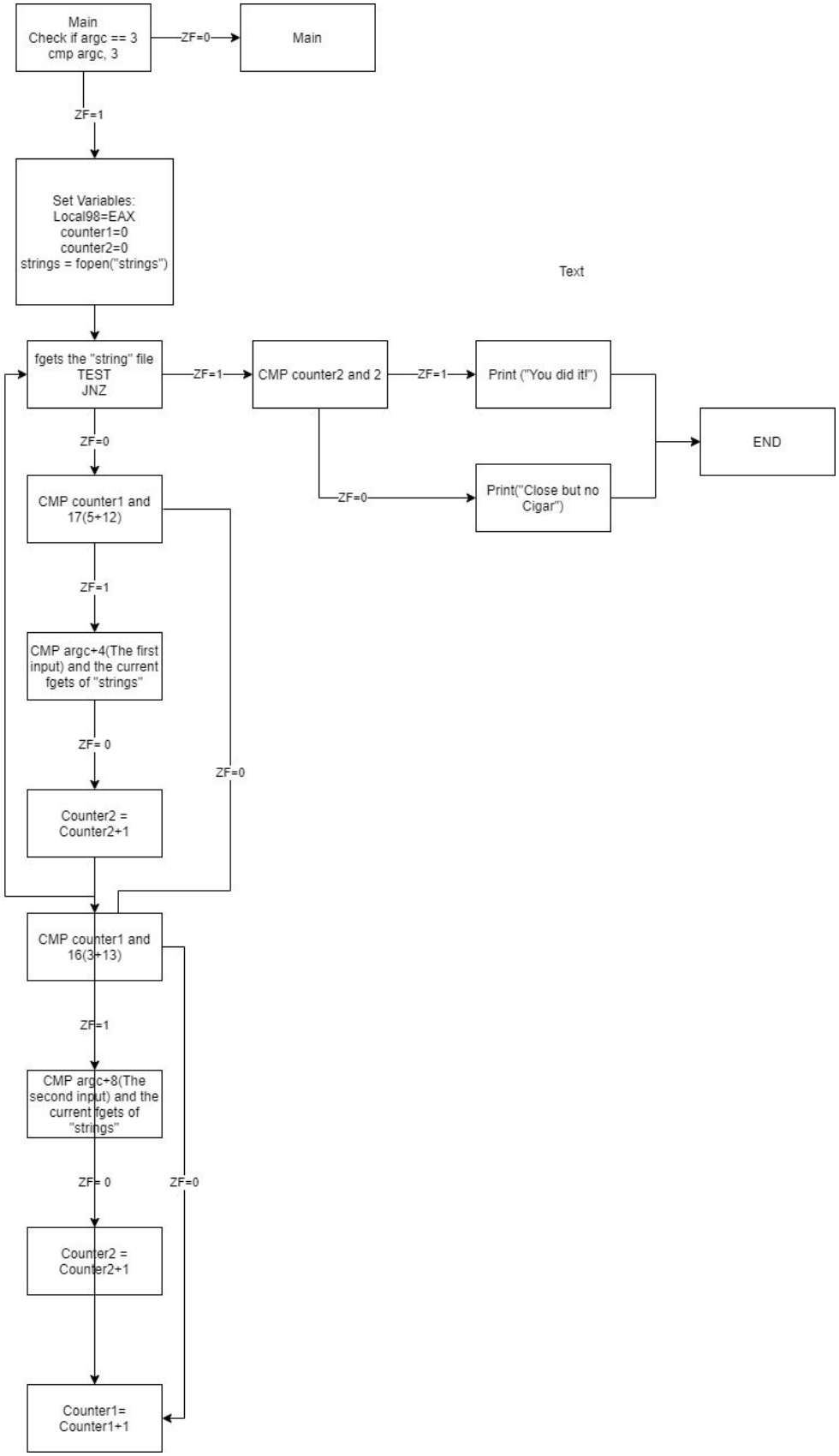
Christopher Gallardo

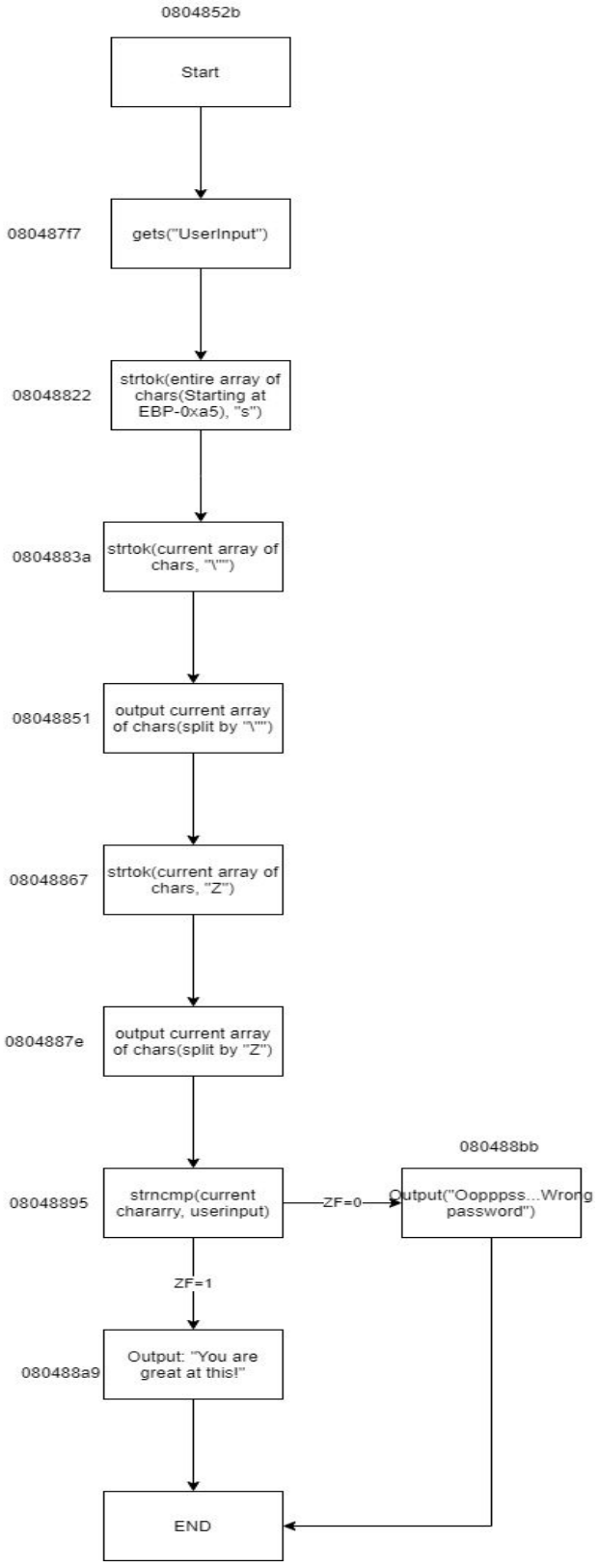
Problem 1:
Goal: print “You did it!”

Solution found: enter the two arguments, “deregister_tm_clones”
-First, I ran objdump -s -j .rodata to see the strings. I see that the string that we are trying to print is located at the address 8048788
-Then, I ran objdump -M intel -d, and found the main function. I then see that the string gets pushed at the address 80486ad.
-I See that the section starting at 080486a1 jumps to the address that we want if the cmp of the dword ptr (Which I labeled as counter #2) is equal to 2.
Looking before that, I can see that the counter is incremented each time
-Looking near the beginning of the code, I saw that there were two main branches: one, if two strings are entered, leads down the path that starts a loop
-Otherwise, the code just outputs “Try again”
-The while loop branch only breaks when the result of the fgets on the file “String” returns 0, meaning it has reached the end of the file
-One of the counter increments each time the loop executes.
-I saw that in the loop there are two if cases, one if the counter equals 16, and one if it equals 17. This would mean that the fgets variable would be currently equal to the strings at lines 17 and 18, since the counter starts at 0. Because of the strncmps, I assumed that the strings on those lines were the strings that would result in the correct output.
-Since the string at line 17 was compared to the user input+8, I knew that it was the second string input and that the first string was at line 18.
-Therefore, the correct input was “deregister_tm_clones __JCR_LIST__”

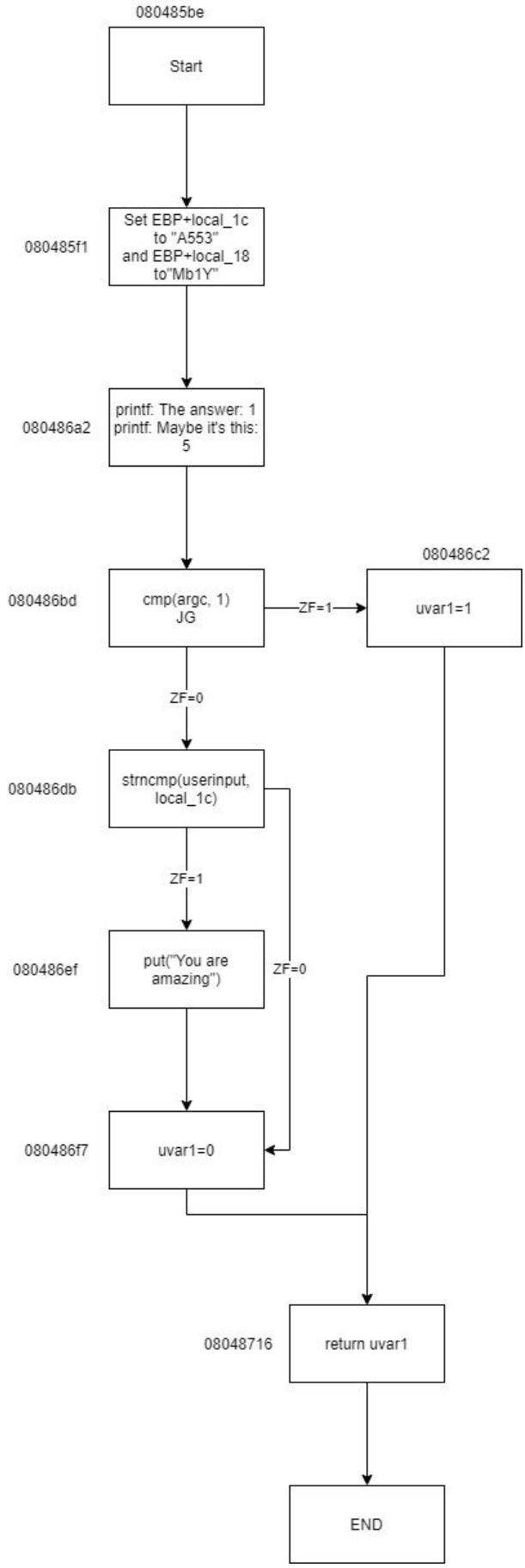

Main
Check if argc == 3
cmp argc, 3
ZF=0
Main
ZF=1
Set Variables:
Local98=EAX
counter1=0
counter2=0
strings = fopen("strings")
Text
fgets the "string" file
TEST
JNZ
ZF=1
CMP counter2 and 2
ZF=1
Print ("You did it!")
END
ZF=0
Print("Close but no Cigar")
ZF=0
ZF=0
CMP counter1 and 17(5+12)
ZF=1
CMP argc+4(The first input) and the current fgets of "strings"
ZF= 0
Counter2 = Counter2+1
ZF=0
CMP counter1 and 16(3+13)
ZF=1
CMP argc+8(The second input) and the current fgets of "strings"
ZF= 0
ZF=0
Counter2 = Counter2+1
Counter1= Counter1+1

Problem 2:
Goal: print “You are great at this :)”
Answer:”yagababa”

-The first thing I did was run the program and entering “a” as the password
-Once I saw the output, I decided to test each string as the password, just in case it was similar to the previous problem. I turned out that the second string displayed was the correct password.
-Upon looking at the binary code, I saw that the outputs were the result of parsing a long string multiple times until it results in the password, which is then compared to the user’s input.


0804852b
Start
080487f7
gets("UserInput")
08048822
strtok(entire array of chars(Starting at EBP-0xa5), "s")
0804883a
strtok(current array of chars, "\"")
08048851
output current array of chars(split by "\"")
08048867
strtok(current array of chars, "Z")
0804887e
output current array of chars(split by "Z")
080488bb
08048895
strncmp(current chararry, userinput)
ZF=0
Output("Oopppss...Wrong password")
ZF=1
080488a9
Output: "You are great at this!"
END

Problem 3:
Goal: Guess Flag (or hidden value...)
-As usual, the first thing I did was simply run the code. When I did this a line stated “The answer: 1”, while another line stated “Maybe it’s this:5”
-Keeping these in mind, I uploaded the code to ghidra, to see the binary code.
-I noted the array named 1c(located at stack-1c), which is made up of the characters: “A553”, and then a similar array, 18(located at stack-18), which is made of the characters:”Mb1Y”
-I tried to input the full string “A553Mb1Y”, and the program output the desired message: “You are amazing!!”


080485be
Start
080485f1
Set EBP+local_1c to "A553" and EBP+local_18 to"Mb1Y"
080486a2
printf: The answer: 1
printf: Maybe it's this: 5
080486bd
cmp(argc, 1)
JG
ZF=1
080486c2
uvar1=1
ZF=0
080486db
strncmp(userinput, local_1c)
ZF=1
080486ef
put("You are amazing")
ZF=0
080486f7
uvar1=0
08048716
return uvar1
END