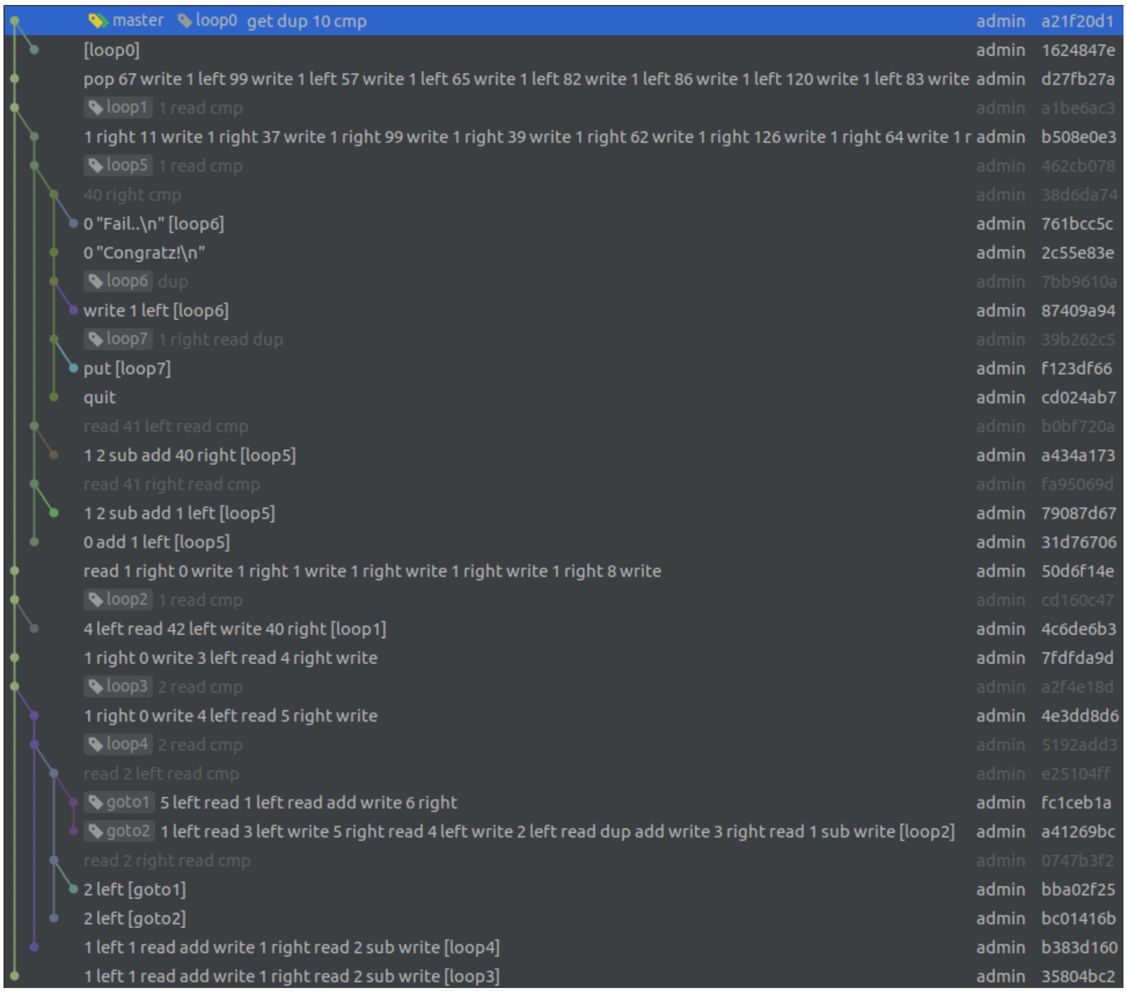
Legitimate Writeup

Exploitation:

1. Find the open `.git` folder
2. Download folder `wget -r`
3. Examine git graph `git log --graph --oneline`



```
master loop0 get dup 10 cmp                                                                                       admin  a21f20d1
[loop0]                                                                                                           admin  1624847e
pop 67 write 1 left 99 write 1 left 57 write 1 left 65 write 1 left 82 write 1 left 86 write 1 left 120 write 1 left 83 write  admin  d27fb27a
loop1 1 read cmp                                                                                                  admin  a1be6ac3
1 right 11 write 1 right 37 write 1 right 99 write 1 right 39 write 1 right 62 write 1 right 126 write 1 right 64 write 1 r  admin  b508e0e3
loop5 1 read cmp                                                                                                  admin  462cb078
40 right cmp                                                                                                      admin  38d6da74
0 "Fail..\n" [loop6]                                                                                              admin  761bcc5c
0 "Congratz!\n"                                                                                                   admin  2c55e83e
loop6 dup                                                                                                         admin  7bb9610a
write 1 left [loop6]                                                                                              admin  87409a94
loop7 1 right read dup                                                                                            admin  39b262c5
put [loop7]                                                                                                       admin  f123df66
quit                                                                                                              admin  cd024ab7
read 41 left read cmp                                                                                             admin  b0bf720a
1 2 sub add 40 right [loop5]                                                                                      admin  a434a173
read 41 right read cmp                                                                                            admin  fa95069d
1 2 sub add 1 left [loop5]                                                                                        admin  79087d67
0 add 1 left [loop5]                                                                                              admin  31d76706
read 1 right 0 write 1 right 1 write 1 right write 1 right write 1 right 8 write                                  admin  50d6f14e
loop2 1 read cmp                                                                                                  admin  cd160c47
4 left read 42 left write 40 right [loop1]                                                                        admin  4c6de6b3
1 right 0 write 3 left read 4 right write                                                                         admin  7fdfda9d
loop3 2 read cmp                                                                                                  admin  a2f4e18d
1 right 0 write 4 left read 5 right write                                                                         admin  4e3dd8d6
loop4 2 read cmp                                                                                                  admin  5192add3
read 2 left read cmp                                                                                              admin  e25104ff
goto1 5 left read 1 left read add write 6 right                                                                   admin  fc1ceb1a
goto2 1 left read 3 left write 5 right read 4 left write 2 left read dup add write 3 right read 1 sub write [loop2]  admin  a41269bc
read 2 right read cmp                                                                                             admin  0747b3f2
2 left [goto1]                                                                                                    admin  bba02f25
2 left [goto2]                                                                                                    admin  bc01416b
1 left 1 read add write 1 right read 2 sub write [loop4]                                                          admin  b383d160
1 left 1 read add write 1 right read 2 sub write [loop3]                                                          admin  35804bc2
```

This is an esoteric language Legit.
Use interpretator to play around.

4. Examine Legit
   - `loop0` is a loop to push input string to stack
   - Then *magic* constants `67`, `99`, `57`, `65`, ... written to the tape
   - `loop1` is a loop over input string
   - Commit `50d6f14e` setup temporary variables on the stack and copy single character of input string and magic
   - `loop2` is a loop from `[8, 0)`
   - Loops `loop3` and `loop4` are quite similar and involves computations, comparisons and jumps to `goto1` and `goto2` (analysis later).
   - After heavy computations commit `4c6de6b3` write result on the tape and repeats the same for the next character in input string.
   - When end of the input sting reached (commit `b508e0e3`), another array of *hardcoded* constants written to the tape `11`, `37`, `99`, `39`, ...

  - `loop5` iterates over array of constants and compares each value with computed earlier values from the input string.
  - If values are not equal then topmost value on the stack decreases by 1 (commits `a434a173` and `79087d67`),
    or remains 0 otherwise (commit `31d76706`).
  - Then in commit `38d6da74`, if topmost stack value is less than 0 (indicates some errors) then `"Fail.."` prints,
    or `"Congratz!"` otherwise.

  So we have to invert computations on the input string.

5. Math without the limits

   Legit support operations `add`, `sub`, greater-than (`>`) operation `cmp`.
   So any other operations have to be represented via them.

   - Less-or-equal-than (`<=`) operation is an inversion of `>`.
   - Equal (`x == y`) comparision is `x > y || y > x`.
   - Division and Modulo `x / y` and `x % y` can be represented as:

     ```
         int mod = x;
         int div = 0;

         while (!(y > mod)) {
             div = div + 1;
             mod = mod - y;
         }
     ```

   - Given division by 2 and modulo by 2, `xor` operation can be represented:

     ```
     int result = 0;
     int mask = 1;

     for (int i = 0; i < 8; i++) {
         if (((a % 2) + (b % 2)) == 1) result = result + mask;
         a = a / 2;
         b = b / 2;
         mask = mask + mask;
     }

     return result;
     ```

   Thus:

   - Loops `loop3` and `loop4` are division and modulo by 2
   - Label `goto1` is `result = result + mask`
   - Label `goto2` is `mask = mask + mask`
   - Loop `loop1` is a loop over input string that XORs each character with *magic* constants

6. PROFIT!

   XOR *magic* constants with *hardcoded* and get the flag
   `SCTF{35073r1C_13617_CrYP70_15_M461C_X0r}`.