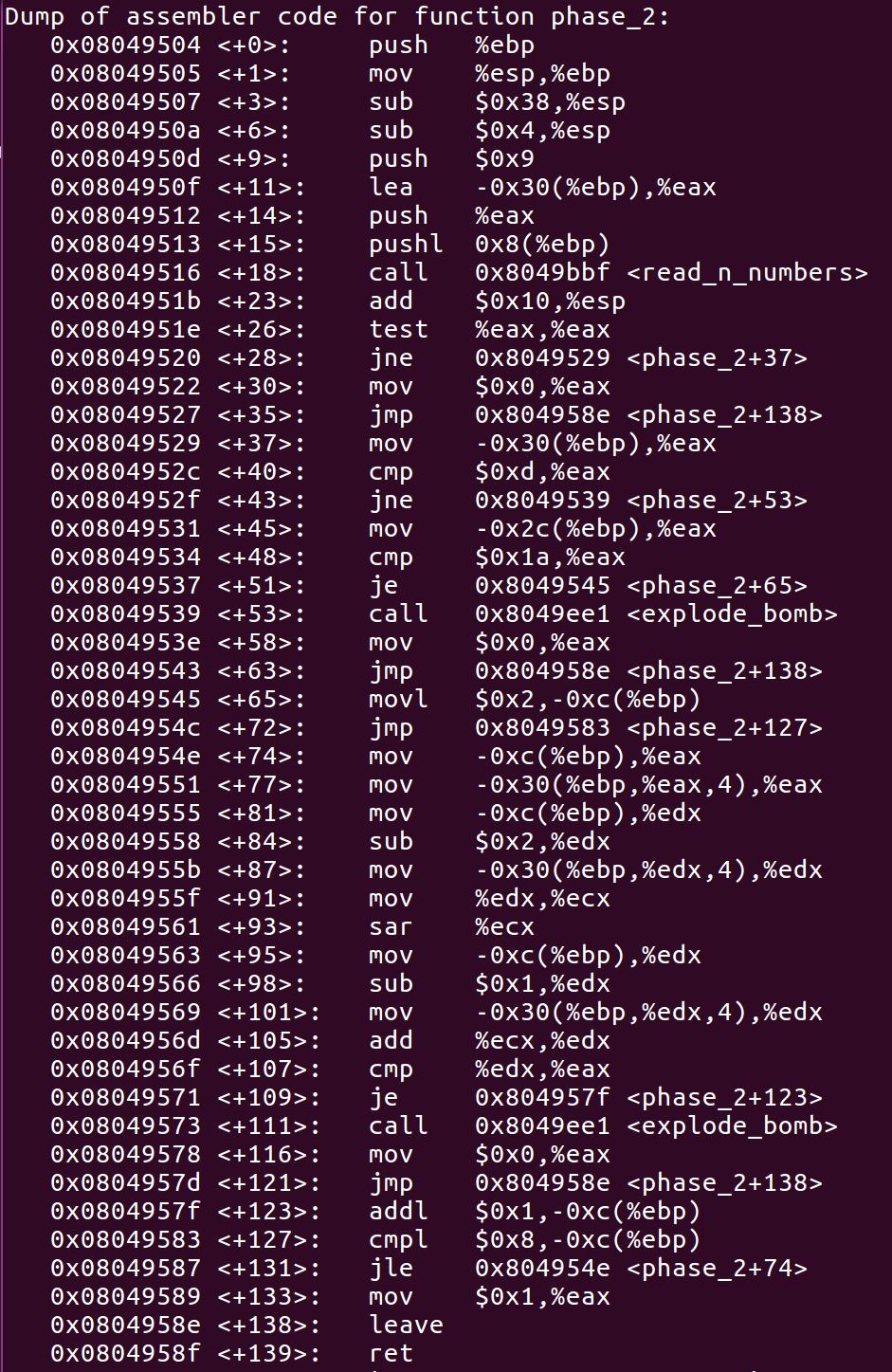


%eax == 2

-0x14(%ebp) == \*(-0x10(%ebp))低八位

\*(-0x10(%ebp) + 2)低八位 == -0x18(%ebp)



push $0x9

-0x30(%ebp) == 13

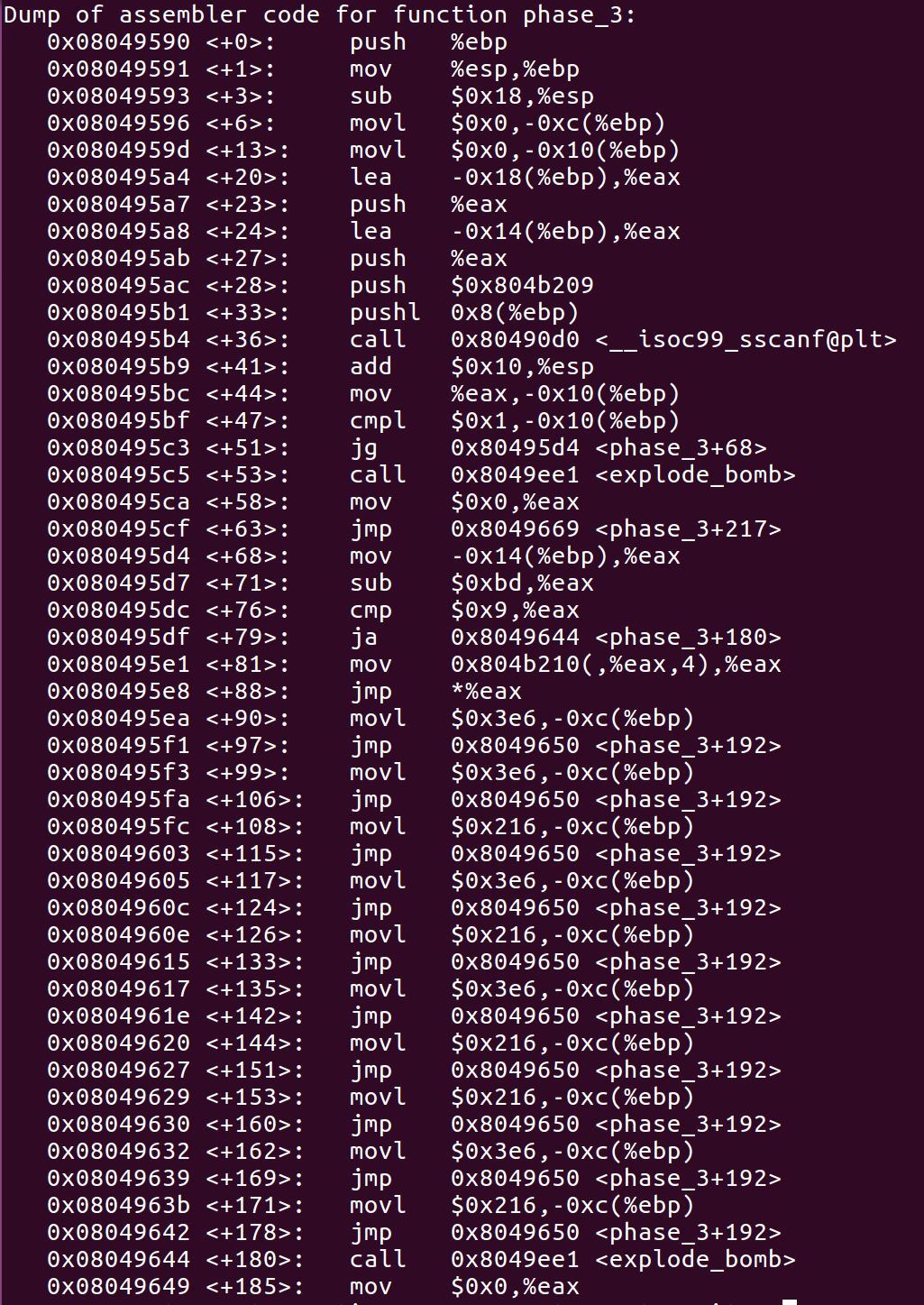
-0x2c(%ebp) == 26

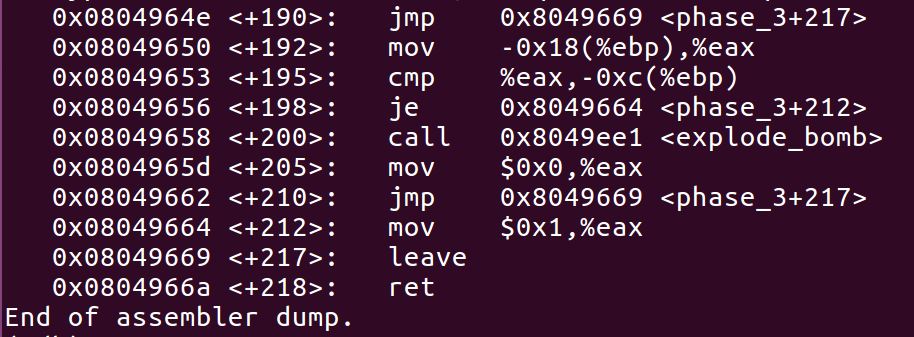
-0xc(%ebp) = 2

for (i = 2; i <= 8; i++)

A[i] == A[i-2] / 2 + A[i-1];

13 26 32 45 61 83 113 154 210





0 <= -0x14(%ebp) – 189 <= 9

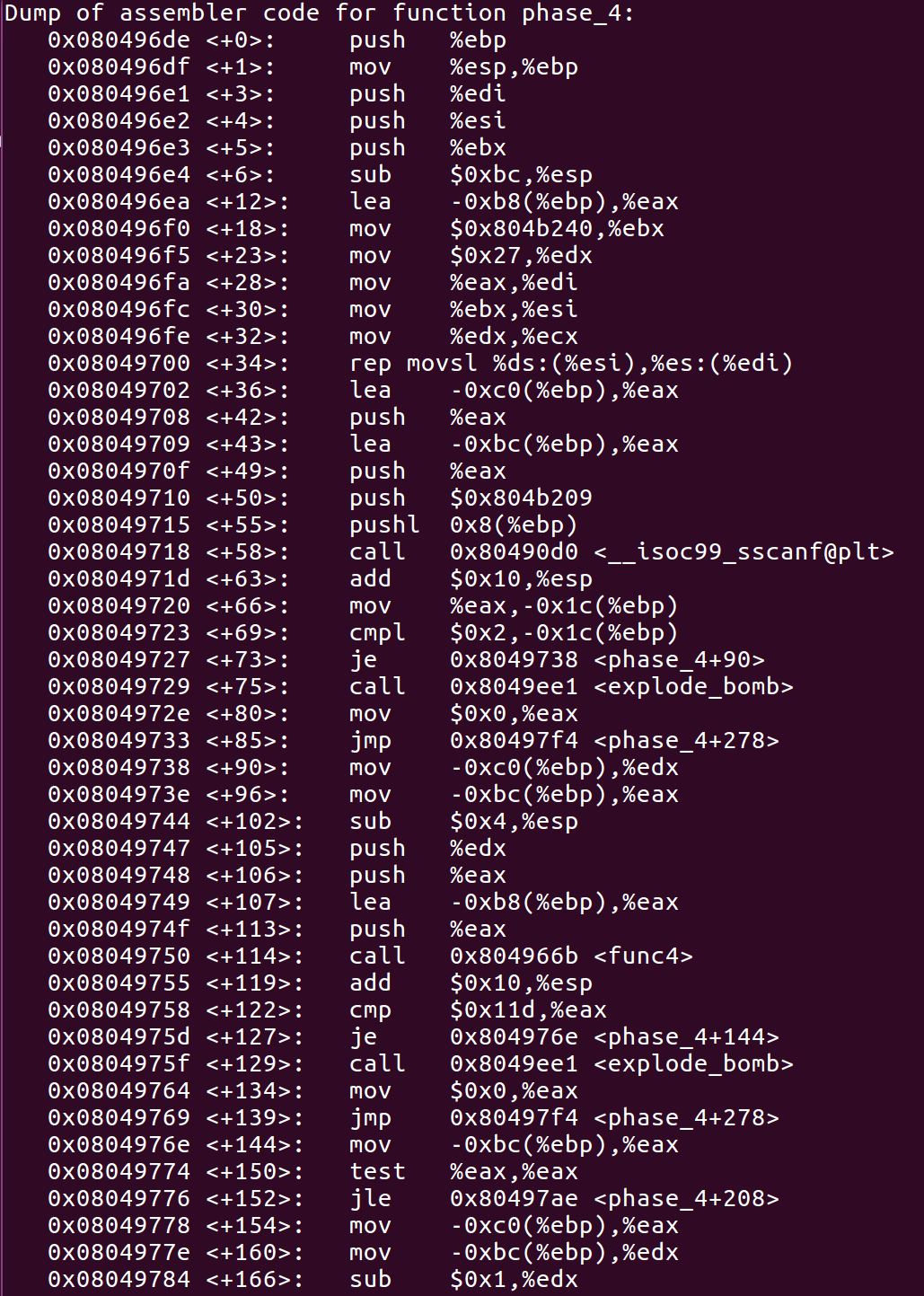
189 <= -0x14(%ebp) <= 198

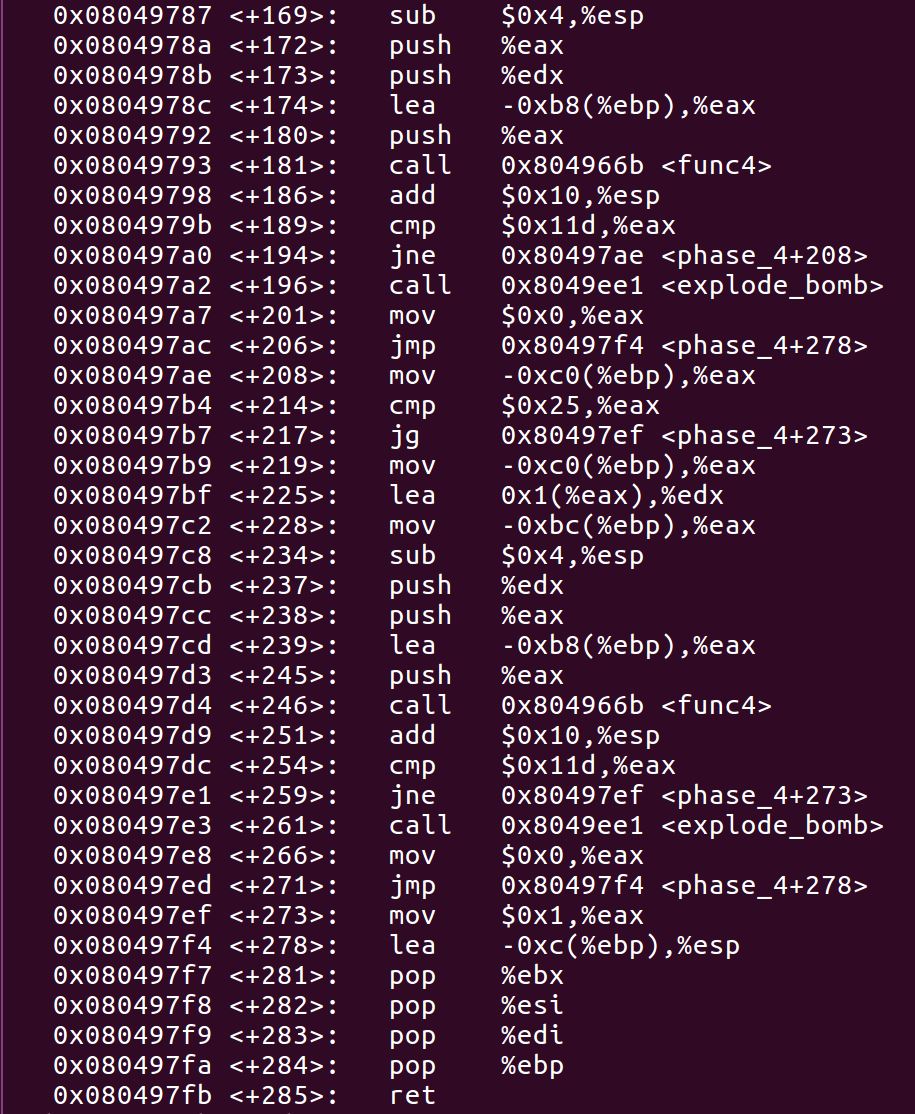
jmp \*(0x804b210 + 4 \* (-0x14(%ebp) – 189))

-0x18(%ebp) == -0xc(%ebp)

0x3e6 = 998

190 998





func4(a, b, 0xffffcf00) == 285

if (a <= 0) {

if (b > 0x25 == 37)

return 1;

else

c = func4((a + 1), b, 0xffffcf00)

}

else

c = func4(a – 1, b, 0xffffcf00)

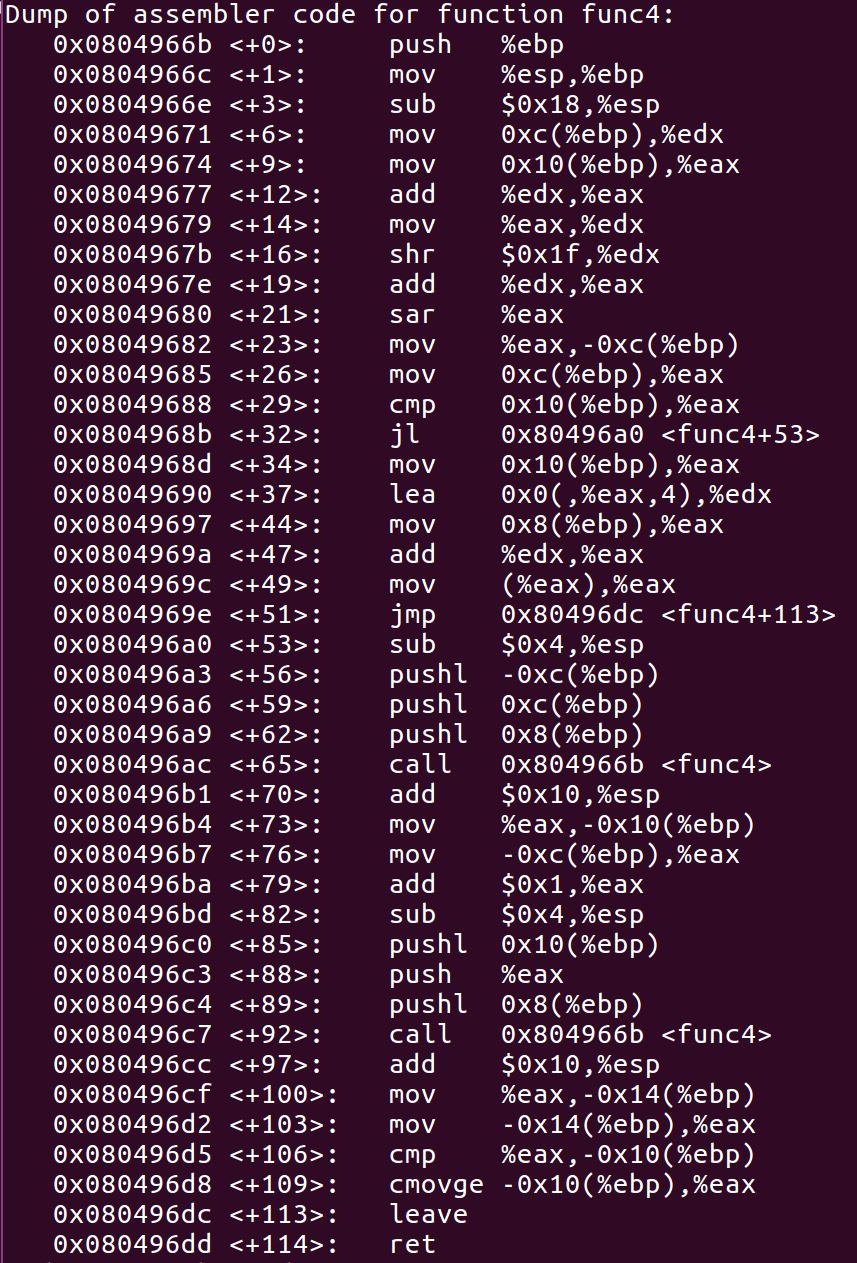
c != 285

a = 0, b = 40

-0xb8(%ebp) = 0xffffcf00

a -0xbc(%ebp) = 0xc edx

b -0xc0(%ebp) = 0x10 eax



b = a + b

a = b

a = a >> 0x1f (31) 逻辑右移（取符号位？）

b = a + b

b = b / 2

int func4(int a, int b, int address) {

c = (a + b + (a + b) >> 31) / 2

if (a < b) {

d = func4(a, c, address)

e = func4(c + 1, b, address)

if (d >= e)

return d

else

return e

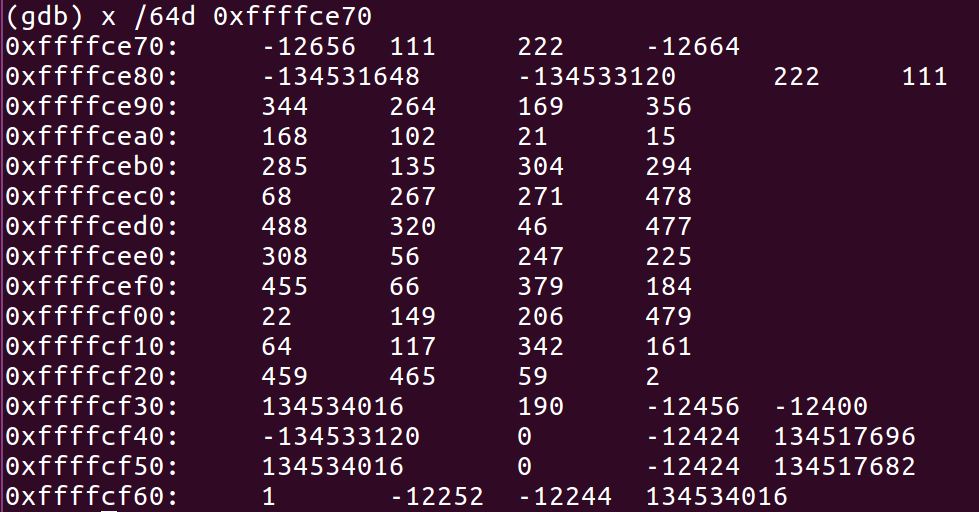
}

else

return \*(address + 4 \* b)

}

a >= 8, b == 8



func4(a, b) == 285

func4(a – 1, b) != 285

a = 7, b = 9, c = 8,

(7,8) c = 7, (7,7) 15 (8,8) 285

(9,9) 135

a = 6, b = 9, c = 8

(6,8) c = 7 (6,7) c = 6, (6,6) 21 (7,7) 15 (8,8) 285

(9,9) 135

a = 5, b = 9, c = 7

(5,7) c = 6, (5,6) c = 5, (5,5) 102 (6,6) 21

(8,9) c = 8, (8,8) 285 (9,9) 135

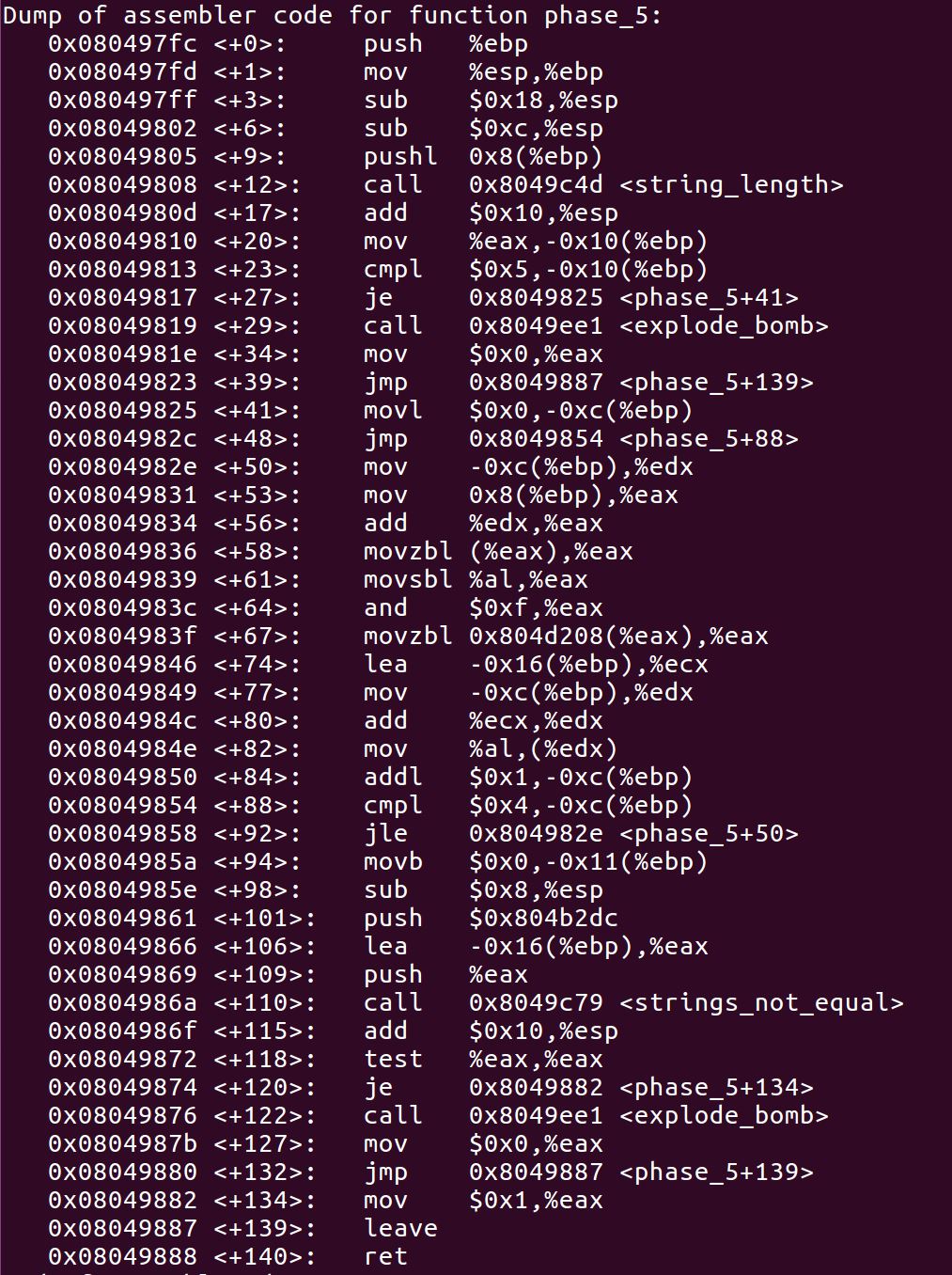
a = 4, b = 9, c = 6

(4,6) c = 5, (4,5) c = 4, (4,4) 168 (5,5) 102

(7,9) c = 8, (7,8) c = 7, (7,7) 15 (8,8) 285 (9,9) 135

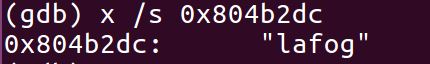
a = 3, b = 9, c = 6

(3,6) c = 4, (3,4) c = 3, (3,3)356



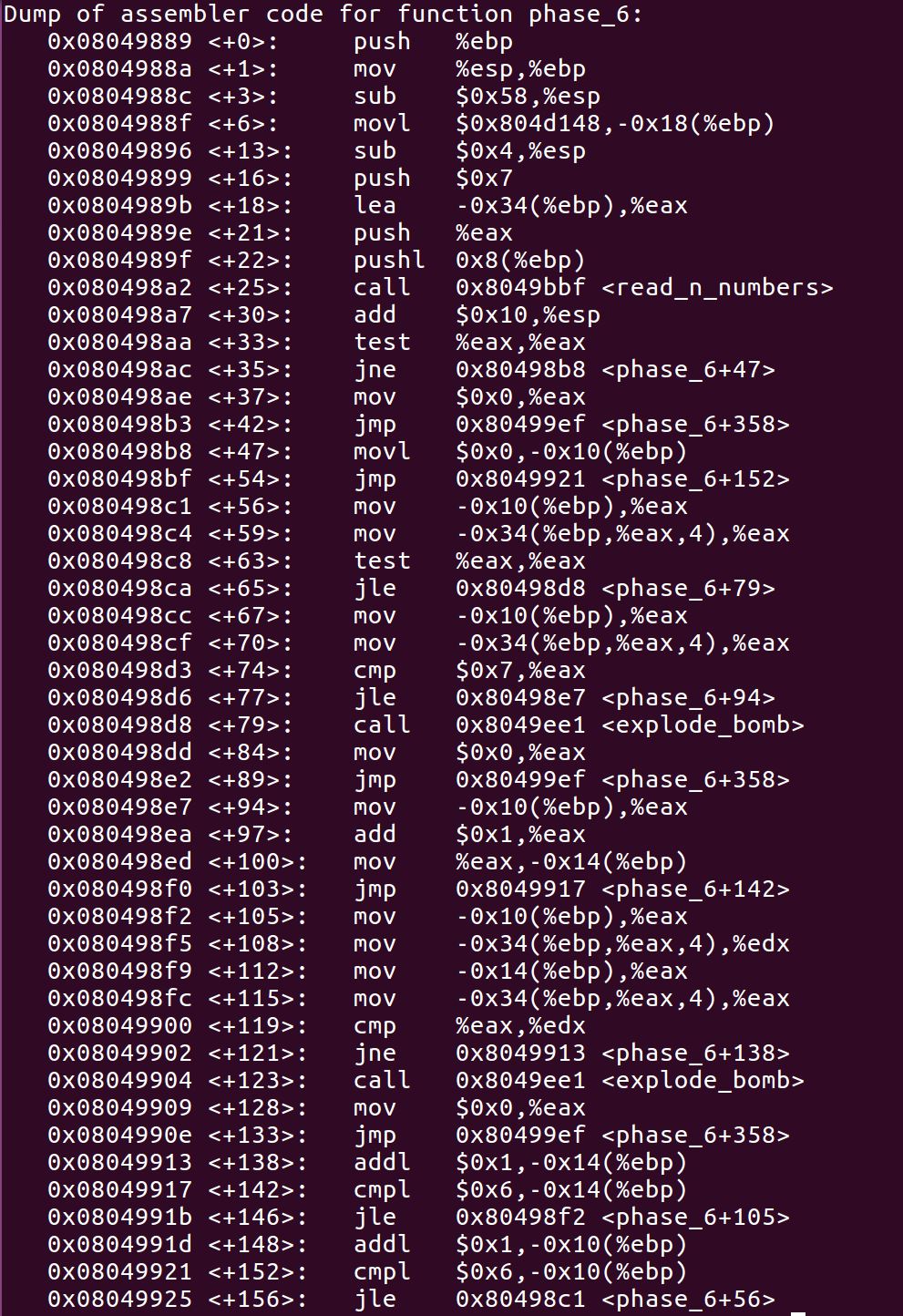
a b c d e f g h i j k l m n o p q r s t u v w x y z

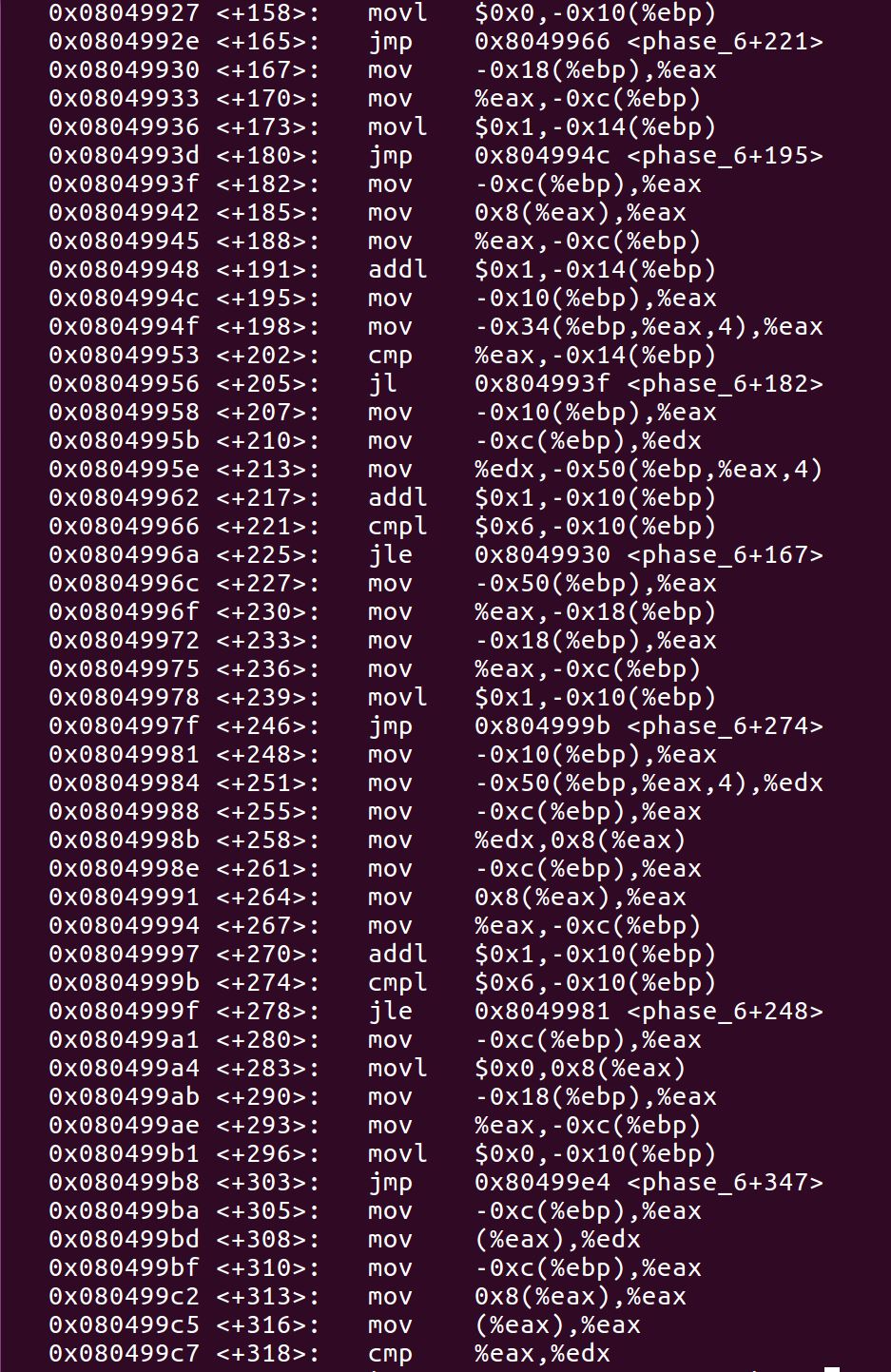
g m i k h f l b o a n j d p c e g m i k h f l b o a



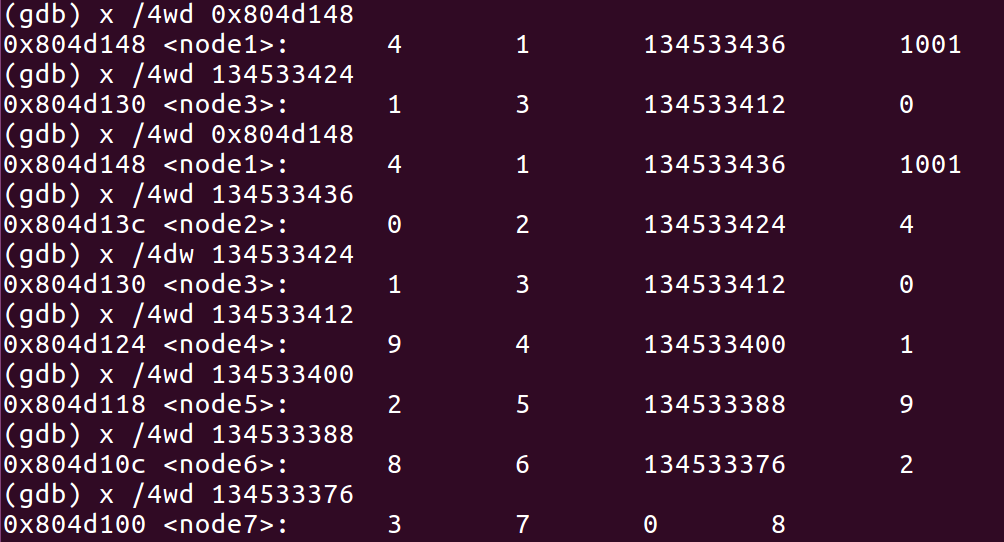


lafog -> gjfia



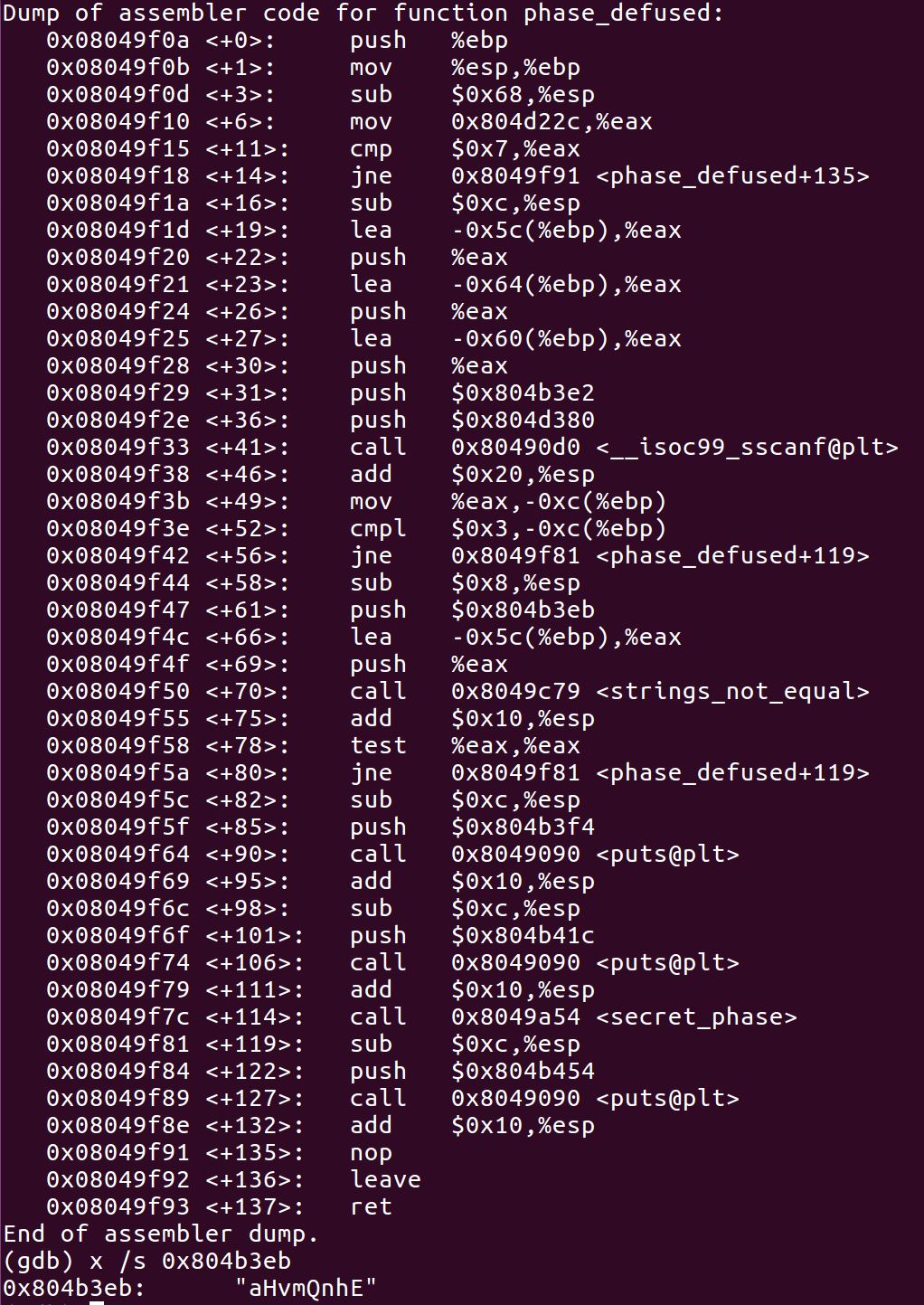


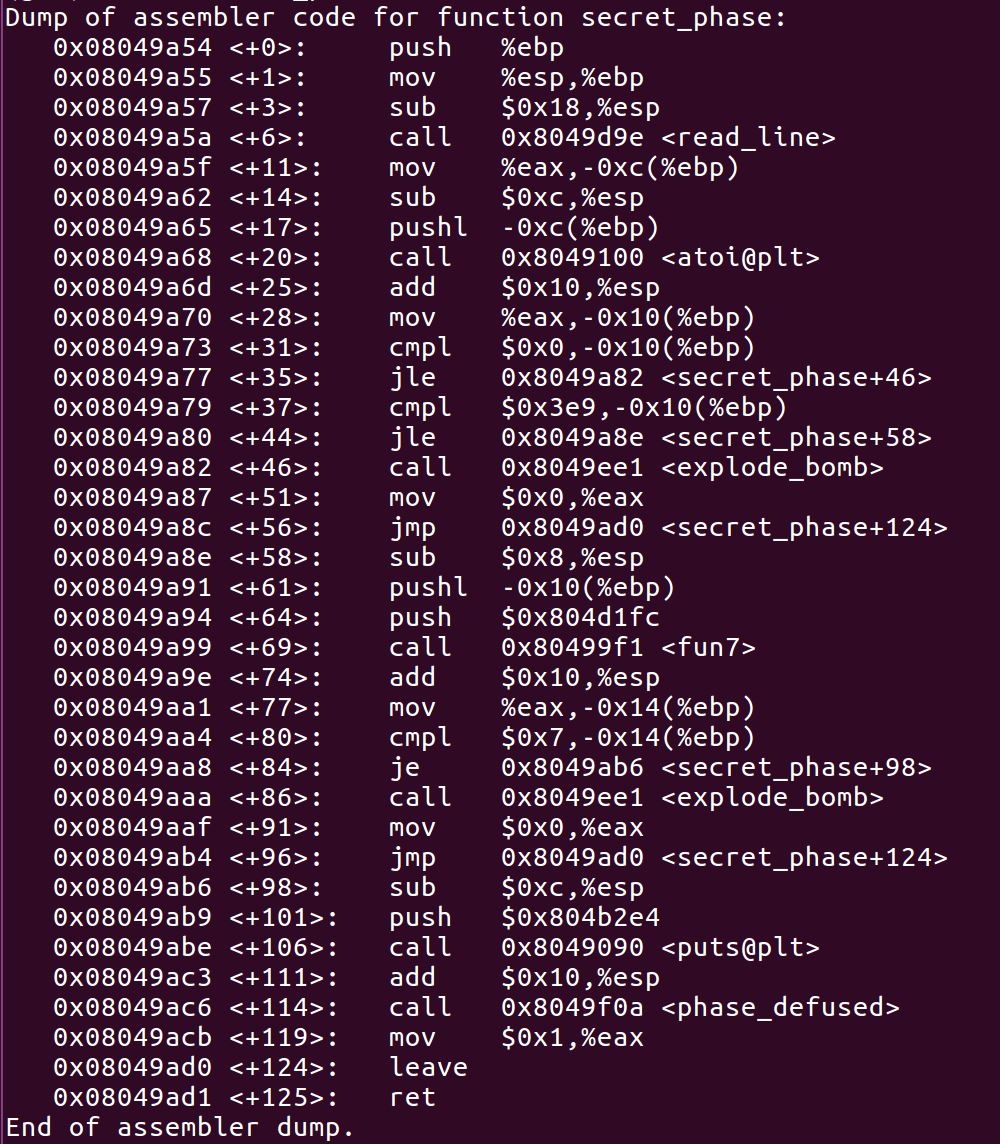




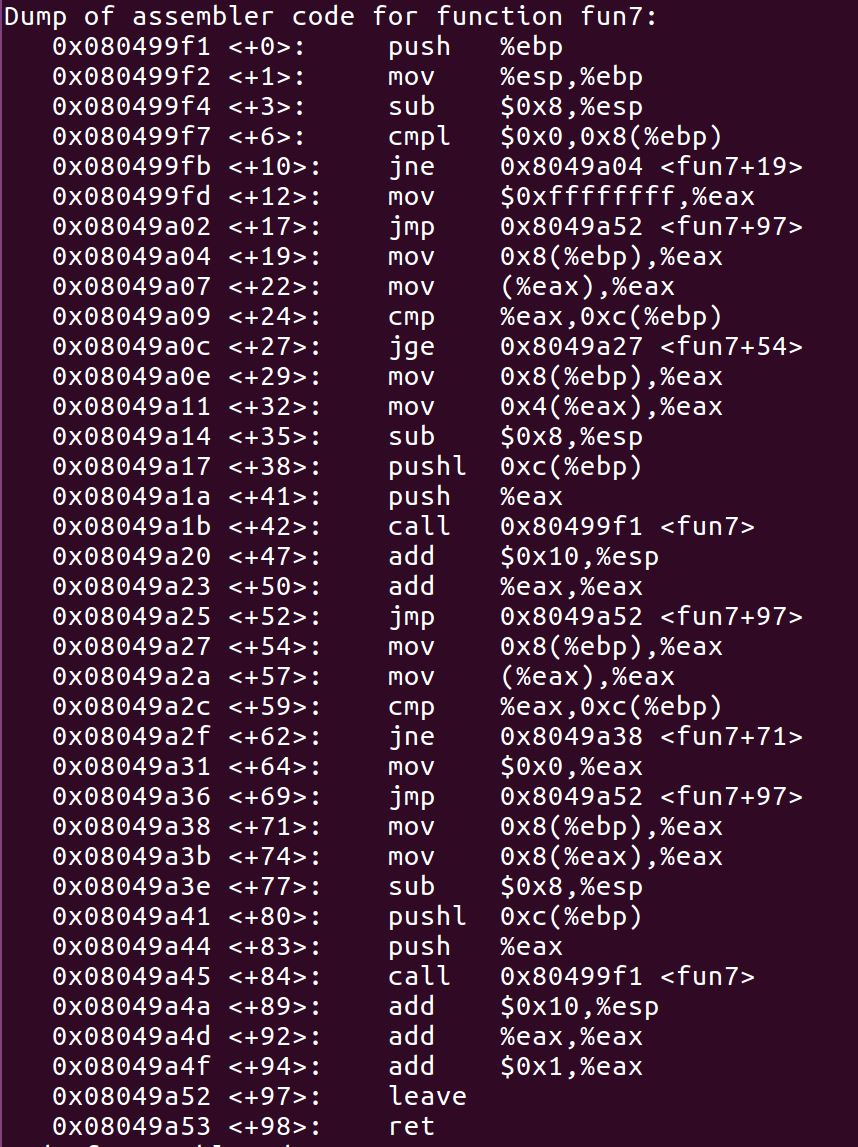
0 1 2 3 4 8 9

2 3 5 7 1 6 4





0 < x <= 0x3e9 == 1001，fun7返回值为7。



int fun7(int num, Node \*root) {

if (root == NULL)

return -1;

else if (root->val < num)

return fun7(num, root->left) \* 2;

else if (root->val == num)

return 0;

else

return fun7(num, root->right) \* 2 + 1;

}

24

/ \

8 32

/ \ / \

6 16 2d 6b

/ \ / \ / \ / \

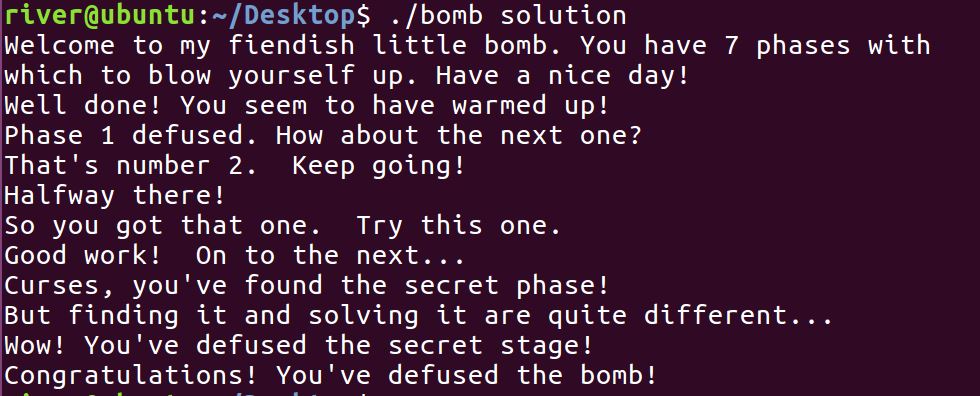
1 7 14 23 28 2f 63 3e9

7 num < 24

3 num < 32

1 num < 6b

0 num == 3e9



func4(a, b) == 285

if (a <= 0) {

if (b > 37)

return 1;

else

c = func4(a, b + 1)

}

else

c = func4(a – 1, b)

c != 285

int func4(int a, int b, int address)

c = (a + b + (a + b) >> 31) / 2

if (a < b)

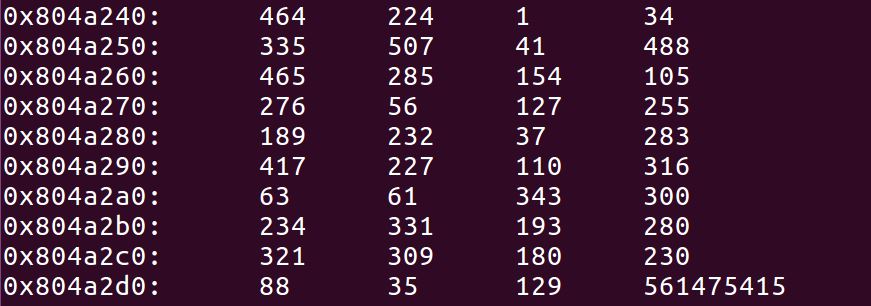
d = func4(a, c, address)

e = func4(c + 1, b, address)

return max(d, e)

else

return \*(address + 4 \* b)



func4(a, b) == 464

func4(a – 1, b) != 464

(0,0) 464

(0,1) c = 0, (0,0) 464, (1,1) 224 -> 464

(0,2) c = 1, (0,1) 464, (2,2) 1 –> 464

(0,3) c = 1, (0,1) 464, (2,3) c = 2, (2,2) 1, (3,2) 1 -> 464

(0,4) c = 2, (0,2) 464, (3,4) c = 3, (3,3) 34, (4,4) 335 -> 464

(0,5) c = 2, (0,2) 464, (3,5) c = 4, (3,4) 335, (5,5) 507 -> 507