1. What is the value of y after both of the following operations?

x = x ^ (~y);

y = y ^ x;

x = 0, y = 0, x = 1, y = 1

x = 1, y = 1, x = 1, y = 0

x = 1, y = 0, x = 0, y = 0

x = 0, y = 1, x = 0, y = 1

y = y ^ (x ^ (~y)) = y ^ x ^ (~y) = (y ^ (~y)) ^ x = 1 ^ x = 1 & x

1. Given the following declarations, do the statements below always evaluate to true?

int x = foo();

int y = bar();

unsigned ux = cookie();

a.

x > ux ====> (~x+1) < 0

false

Left hand side is unsigned comparison, so any negative x with absolute value graeter than ux will be treated true, but the declaration actually wants to prove x itself is positive and greater than ux

Counter-example: x = 111…1, ~x + 1 = 000…1 > 0

b.

ux - 2 >= -2 ====> ux <= 1

false

Left hand side is unsigned comparison

Counter-example: ux = umax = 111…1, -2 (unsigned) = 111…0, ux – 2 = 111…01 < 111…0. And ux = umax > 1 (000…1)

c.

(x^y)^x == (x+y)^((x+y)^y)

true

(x+y) ^ ((x+y)^y) = (x+y)^(x+y)^y = 0^y = y

(x^y)^x = (x^x)^y = 0^y = y

d.

(x < 0) && (y < 0) == (x + y) < 0

false

negative overflow

1. char\*\* apple[5][9];

char\* banana[1][9];

char strawberry[4][2];

How many bytes of space would these declarations require?

1. 8\*5\*9 = 360

2. 8\*1\*9 = 72

3. 1\*4\*2 = 8

1. Consider the following struct:

typedef struct {

char first;

int second;

short third;

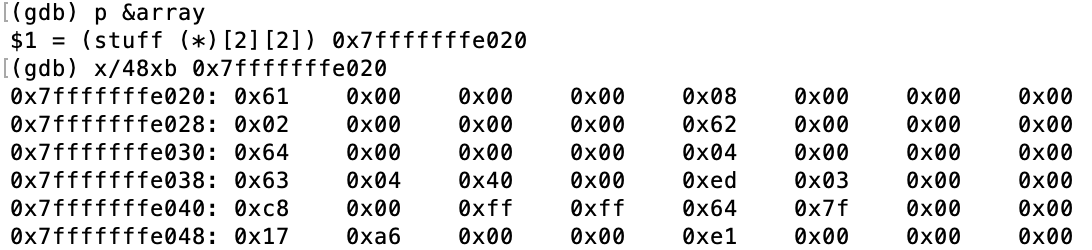
} stuff;

Size of stuff = 1+(3)+4+2+(2) = 12 bytes

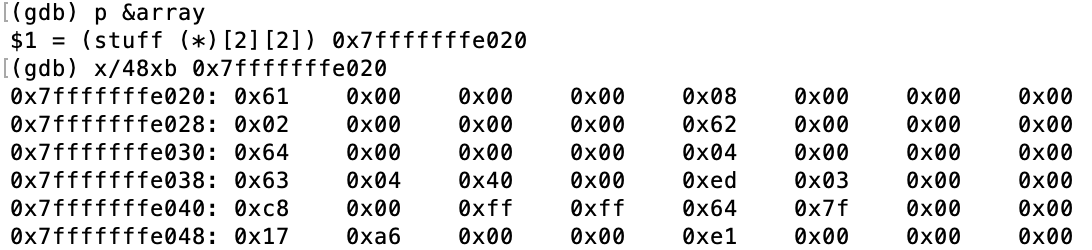
Say we are debugging an application in execution using gdb on a 64-bit, little-endian architecture. The application has a variable called array - defined as:

stuff array[2][2];

Using gdb we find the following information at a particular stage in the application:



And:



What is the value of

array[1][0].second

array[1][0] starts at 0x63 (after 24 bytes in first row)

second is 5th to 8th bytes, 0xed, 0x03, 0x00, 0x00

By little endian, second is 0x000003ed = 1005

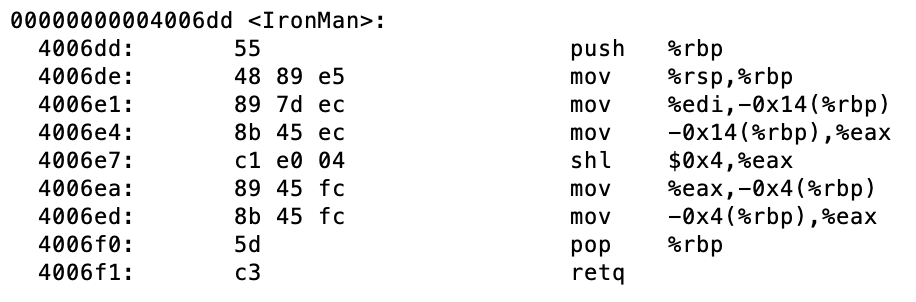
At this particular stage of the application?

i.e. what would be returned from the statement:

printf("%d\n", array[1][0].second);

1005

1. The following is part of the result of the command ‘objdump -d’ on an executable



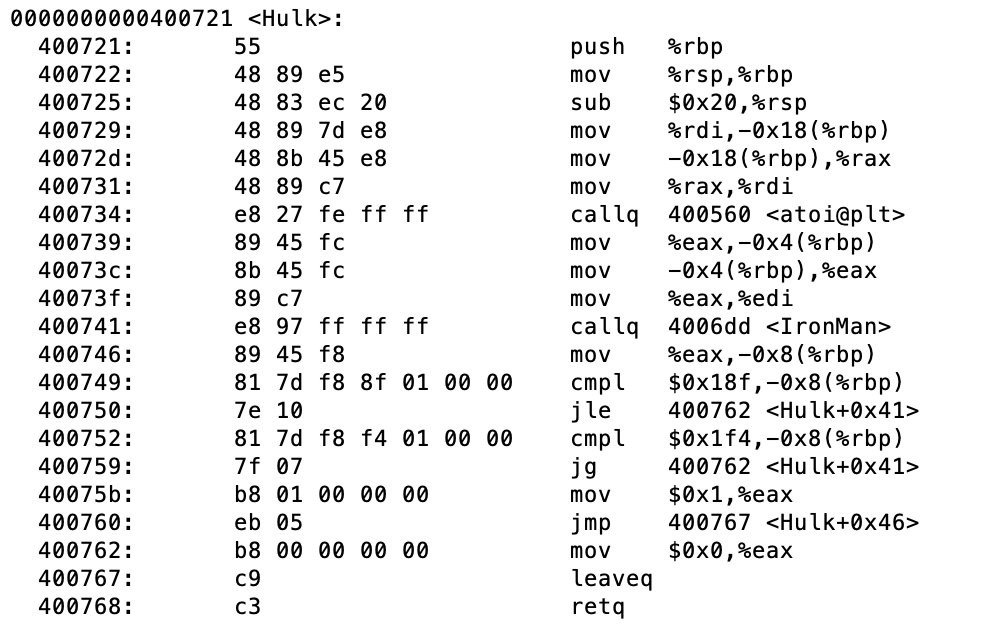
Say the declaration for the function IronMan was:

int IronMan(int scraps);

Given that the integer 23 was passed into the function, what is the return value?

368

1. The following is a continuation from the previous problem.



Given that the function returns 1, what do we know about the value of %edi right before instruction 0x400741 is executed?

Between 25 and 31 (inclusive)

1. What is the value of the following 8-bit, tiny floating point number? Note that the exponent field is 4 bits, and the fractional field is 3.

**01100000**

S: 0

E: 1100

F: 0000

Bias = 2^(4-1) – 1 = 6

(-1)^S \* F \* 2^E = 1 \* 1(leading one) \* 2^(12-7) = 32