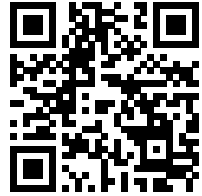


Name:

UID:

For Discussion Credit, it's required to fill out this survey:
tinyurl.com/cs33-25-laeval



1. Floating Point Fun!

Assume a floating-point representation with 1 sign bit, 3 exponent bits, and 4 mantissa bits.

- (a) Decode the 8-bit floating point $0xe7$ to decimal.
- (b) Encode the following numbers with the floating-point representation.
- (i) -15.5
 - (ii) -0
 - (iii) -1
 - (iv) $+0$
 - (v) $+\infty$

2. Always True?

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

```
int x = foo();  
int y = bar();  
unsigned ux = cookie();
```

(a) $x > ux \Rightarrow (\sim x + 1) < 0$

(b) $ux - 2 \geq -2 \Rightarrow ux \leq 1$

(c) $(x \wedge y) \wedge x == (x + y) \wedge ((x + y) \wedge y)$

(d) $(x < 0) \&\& (y < 0) == (x + y) < 0$

(e) $y \wedge x \wedge \sim y == \sim x$

3. What's the Size?

```
char** apple[5][9];
char* banana[1][9];
char strawberry[4][2];
struct ucla {
    char blue[6];
    union {
        int gold;
        char joe[7];
    } bruin;
} arr[4];
```

How many bytes of space would these declarations require?

(a) apple

(b) banana

(c) strawberry

(d) arr

4. What's in that Struct

```
typedef struct {  
    char first;  
    int second;  
    short third;  
} stuff;
```

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:

```
(gdb) p &array  
$1 = (stuff (*)[2][2]) 0x7fffffff020
```

```
(gdb) x/48xb 0x7fffffff020  
0x7fffffff020: 0x61 0x00 0x00 0x00 0x08 0x00 0x00 0x00  
0x7fffffff028: 0x02 0x00 0x00 0x00 0x62 0x00 0x00 0x00  
0x7fffffff030: 0x64 0x00 0x00 0x00 0x04 0x00 0x00 0x00  
0x7fffffff038: 0x63 0x00 0x40 0x00 0xed 0x03 0x00 0x00  
0x7fffffff040: 0xc8 0xff 0xff 0x64 0x7f 0x00 0x00 0x00  
0x7fffffff048: 0x17 0xa6 0x00 0x00 0xe1 0x00 0x00 0x00
```

What is the value of `array[1][0].second` at this particular stage of the application? i.e., what would be returned from the statement:

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

5. Avengers Assemble!

The following is part of the result of the command `objdump -d` on an executable.

```
1 0000000000401180 <Hulk>:
2   401180:      89 f8          mov     %edi,%eax
3   401182:      c1 e0 04       shl     $0x4,%eax
4   401185:      c3             retq
5
6 0000000000401190 <Thor>:
7   401190:      89 f8          mov     %edi,%eax
8   401192:      c1 f8 02       sar     $0x2,%eax
9   401195:      09 f8          or      %edi,%eax
10  401197:      c3             retq
11
12 00000000004011a0 <StrongestAvenger>:
13  4011a0:      55             push    %rbp
14  4011a1:      89 f5          mov     %esi,%ebp
15  4011a3:      53             push    %rbx
16  4011a4:      e8 d7 ff ff ff  callq   401180 <Hulk>
17  4011a9:      89 ef          mov     %ebp,%edi
18  4011ab:      89 c3          mov     %eax,%ebx
19  4011ad:      e8 de ff ff ff  callq   401190 <Thor>
20  4011b2:      39 c3          cmp     %eax,%ebx
21  4011b4:      0f 4e c3       cmovle  %ebx,%eax
22  4011b7:      5b             pop     %rbx
23  4011b8:      5d             pop     %rbp
24  4011b9:      c3             retq
```

(a) What is the return value of `Hulk` (23)?

(b) What is the return value of `Thor` (16)?

(c) What is the return value of `StrongestAvenger` (23, 16)?