

**UNCA CSCI 255**  
**Final Exam Fall 2016**  
 15 November, 2016 - 3:00 PM to 5:30 PM

This is a closed book and closed notes exam. Communication with anyone other than the instructor is not allowed during the exam. Furthermore, calculators, cell phones, and any other electronic or communication devices may not be used during this exam. Anyone needing a break during the exam must leave their exam with the instructor. Cell phones or computers may not be used during breaks.

*If you want partial credit for imperfect answers, explain the reason for your answer!*

Name: \_\_\_\_\_

**Problem 1 (3 points) Range**

What are the smallest **and** largest integers that can be represented as 7-bit unsigned numbers?

What are the smallest **and** largest integers that can be represented as 7-bit signed two's complement numbers?

**Problem 2 (5 points) C expressions**

In the left column, there are some tricky, and some not-so tricky, C expressions. (Except for the first two, these are also Java expressions.) Write their values in the right column. Express your answers in base 10. Assume two's complement representation.

!(15 > 2)	0
6    (143 * 66 / 33)	1
25 << 3	200
25 >> 3	3
~25	-26
25 + 15	40
25   15	31
25 ^ 15	22
3 * 2 / 4	1
3 + 2 / 4	3

(in Java just  
it's 6)  
\* 0

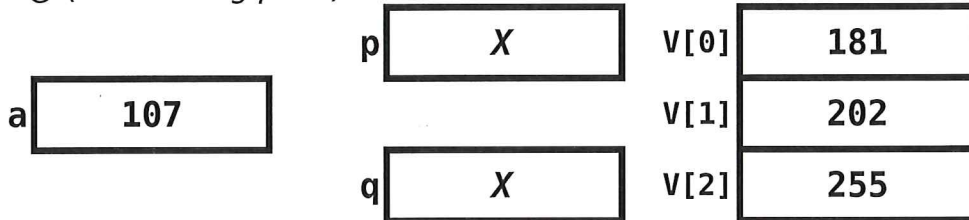
**Problem 16 (8 points)**

In this question, you are to fill in boxes representing the following C integer or pointer variables to show their values after each of seven sections of C code are executed. **You should consider all the sections as being independently executed after the following declaration and initialization statements:**

```
int    a = 107 ;
int    V[3] = {181, 202, 255} ;
int    *p = NULL ;
int    *q = NULL ;
```

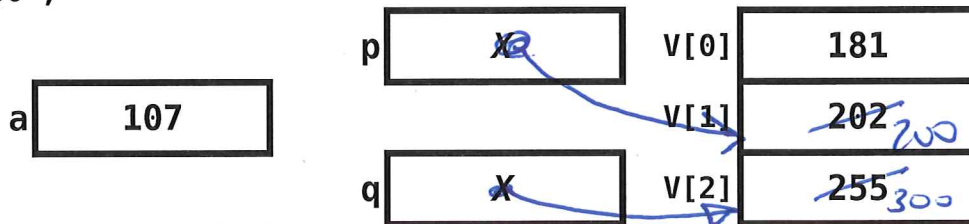
As you might guess, `null` in Java is similar to `NULL` in C. Draw the value `NULL` with a little `X`. Don't ever just leave the pointer variable boxes empty.

Code section @ (the starting point)



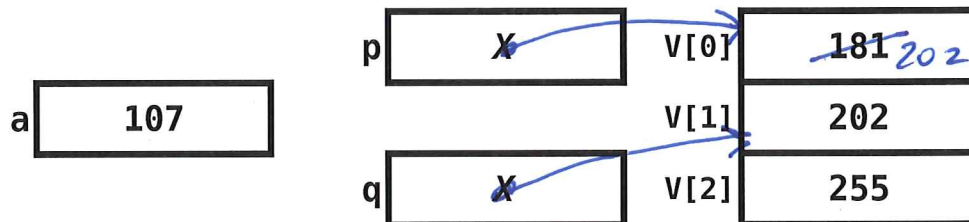
Code section A

```
p = &V[1] ;
q = &V[2] ;
*p = 200 ;
*q = 300 ;
```



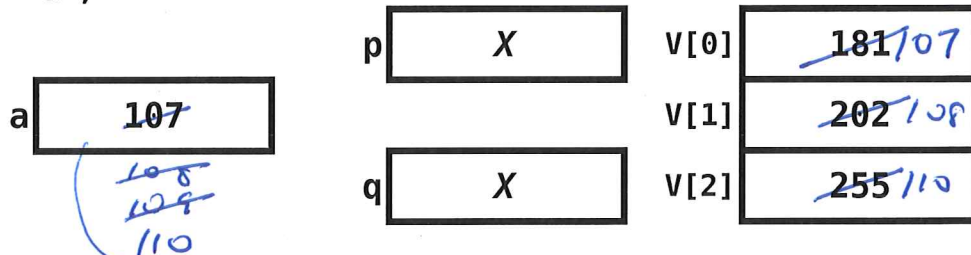
Code section B

```
p = V ;
q = p + 1 ;
*p = *q ;
```



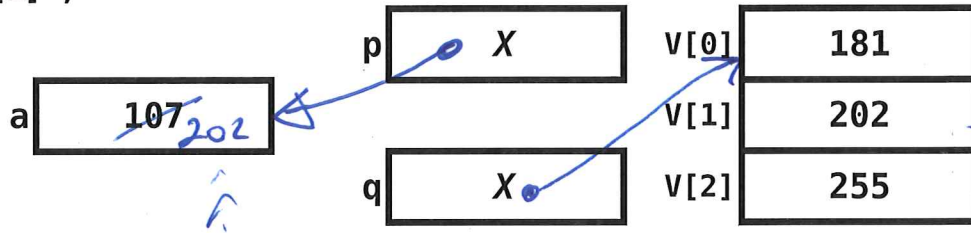
Code section C

```
V[0] = a++ ;
V[1] = a++ ;
V[2] = ++a ;
```



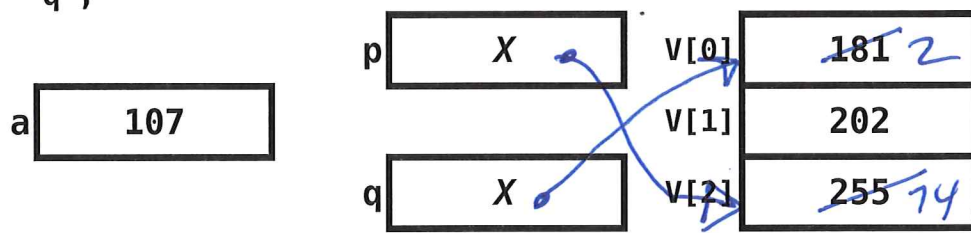
Code section D

```
p = &a ;
q = &V[0] ;
*p = q[1] ;
```



Code section E

```
p = &V[2] ;
q = &V[0] ;
*p = *p - *q ;
*q = p - q ;
```



Code section F

```
p = &V[0] ;
*p++ = 320 ; // same as *(p++) = 320 ;
a = (*p)++ ;
```

