

## CS240 Lab 6 Assignment C Pointers

For each row, evaluate the expression in the first column, and make a prediction for the **type** and the **numeric value** of the expression. Assume that you are using a machine with 32-bit addresses and integers and little endian storage, and that:

- `char* p = (char*) 0x1100`
- `char* q = (char*) 0x1110`

for pointer types, write the *numeric address*

for integer types, write the *numeric value*

	type	numeric address/value
0. <code>p</code>	<code>char *</code>	<code>0x1100</code>
1. <code>&amp;p[1]</code>		
2. <code>&amp;p[-1]</code>		
3. <code>&amp;p[0]</code>		
4. <code>&amp;p[1] - &amp;p[0]</code>		
5. <code>&amp;p[8]</code>		
6. <code>(p + 1) - p</code>		
7. <code>&amp;p[16] - p</code>		
8. <code>q - p</code>		
9. <code>sizeof(p)</code>		
10. <code>sizeof(*p)</code>		
<code>//assume this statement is executed before evaluating the following statements</code> <code>int* ip = (int*) p;</code>		
11. <code>&amp;ip[0]</code>		
12. <code>&amp;ip[1]</code>		
13. <code>&amp;ip[1] - &amp;ip[0]</code>		
14. <code>(char*) &amp;ip[1] - p</code>		
15. <code>sizeof(ip)</code>		
16. <code>sizeof(*ip)</code>		
17. <code>&amp;ip[sizeof(int)]</code>		
18. <code>ip + sizeof(int)</code>		
19. <code>ip + 1</code>		
20. <code>p + sizeof(int)</code>		
<code>int* iq = (int*) q; //assume this statement is executed before</code> <code>evaluating the next statements</code>		
21. <code>iq - ip</code>		
22. <code>&amp;iq[-1] - ip</code>		
<code>p[0] = p[1] = p[2] = p[3] = 0; //assume this statement is</code> <code>executed before evaluating the next statement</code>		
23. <code>*ip</code>		
<code>*(char*) ip = 1; //assume this statement is executed before</code> <code>evaluating the next statement</code>		
24. <code>*ip</code>		
<code>*((char*) ip + 1) = 1; //assume this statement is executed before</code> <code>evaluating the next statements</code>		
25. <code>p[1]</code>		
26. <code>*ip</code>		
<code>*((char*) ip) = 2; //assume this statement is executed before</code> <code>evaluating the next statements</code>		
27. <code>*((char*) ip)</code>		
28. <code>*ip</code>		