

Arrays in C - Handout

Arrays are adjacent locations in memory storing the same type of data object

`a` (array name) returns the array's address

`&a[i]` is the address of `a[0]` plus `i` times the element size in bytes

Declaration: `int a[6];`

Indexing: `a[0] = 0x015f;`
`a[5] = a[0];`

No bounds checking: `a[6] = 0xBAD;`
`a[-1] = 0xBAD;`

Pointers: `int* p;`

equivalent $\left\{ \begin{array}{l} p = a; \\ p = \&a[0]; \\ *p = 0xA; \end{array} \right.$

array indexing = address arithmetic
 (both scaled by the size of the type)

equivalent $\left\{ \begin{array}{l} p[1] = 0xB; \\ *(p+1) = 0xB; \\ p = p + 2; \end{array} \right.$

`a[0]`
`a[2]`
`a[4]`

`p`

`*p = a[1] + 1;`

	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7
	0x8	0x9	0xA	0xB	0xC	0xD	0xE	0xF
0x00								
0x08								
0x10								
0x18								
0x20								
0x28								
0x30								
0x38								
0x40								
0x48								