600.220 Intermediate Programming

Pointer Arithmetic

Outline

- Pointer arithmetic basics
- Pointer arithmetic for arrays

Pointer Arithmetic!

- +, -, +=, -= for other pointers or integers
- Most often used on pointers that are arrays
- Doesn't add the actual number, it adds that number times how many bytes each element takes up
  - for variable `int * p`, code `p+1` will in fact add 4 bytes (`sizeof(int)`) to `p`'s address
- `ptr1 = ptr2` assignment works for pointers of same type
- `ptr1 == ptr2` etc makes sense to compare ptrs ("do they point to the same memory location?"), and `ptr == NULL`

Pointer Arithmetic and Arrays

- A declared array, say `int a[10]`, is "really" just an address that starts a block of memory.
- Writing `a` is generally the same as writing `&a[0]`
- `a[3]` is a synonym for `*(a + 3)` (offset three from pointer to start of array)
- `&a[3]` is a synonym for `a + 3`

Pointer Arithmetic and Arrays

```c
// pointerArith.c:
#include <stdio.h>
#include <stdlib.h>
int main() {

   int array[] = {2, 4, 6};

   printf("array[1] = %d, ", array[1]);
   printf(" *(array + 1) = %d, ", *(array+1));
   printf(" array = %p\n", (void *)array);
   printf(" &array[1] = %p, ",(void *)  &array[1]);
   printf(" array + 1 = %p\n", (void *)(array + 1));
   return 0;
}
```

```
$ gcc -std=c99 -Wall -Wextra -pedantic pointerArith.c
$ ./a.out
array[1] = 4,  *(array + 1) = 4,  array = 0x7ffee2b3c89c
 &array[1] = 0x7ffee2b3c8a0,  array + 1 = 0x7ffee2b3c8a0
```

Checkpoint Poll!

What is the correct output?

```c
#include <stdio.h>

int main() {
  int a[] = {1, 1, 2, 3, 4};
  printf("%d ", (*a) + 7);
  printf("%d ", *(a + 4));
  printf("%d ", *(&a[1] + 1));
  return 0;
}
```

A. 2 4 3

B. 8 4 2

C. 8 5 2

D. 7 4 2

E. The program does not compile and/or has an error.