Peer Instruction #12: More C Programming

Subtitled: The nightmare never ends!

```
short i = 32767;
unsigned short j = 32767;
printf("%d,%d\n", ++i, ++j);
```

What does the code shown above print?

A.32767, 32767
B.32767, 32768
C.32768, 32768
D.-32768, 32768
E.0, 32768

Primitive Types

```
float d = 1.0;
int *p0 = (int *) &d;
float *p1 = &d;
printf("0x%08x,%.1f\n", *p0, *p1);
```

What does the code shown above print?

A.0x3f800000, 1.0
B.0x3f900000, 1.0
C.0x3fa00000, 1.0
D.0x3fb00000, 1.0

C Pointers

```
int array[] = {1, 2, 3, 4, 5, 6};
printf("%d,", array[2]);
printf("%d,", *(array+5));
printf("%d\n", *(&array[4]));
```

What does the code shown above print?

A.2,5,2
B.2,5,4
C.3,6,4
D.3,6,5
E.Will not compile

C Arrays

```
printf("%d, ", strcmp("Alpha", "Beta"));
printf("%d, ", strcmp("Beta", "Beta"));
printf("%d\n", strcmp("Tau", "Beta"));
```

What does the code shown above print?

A. 0, 1, 0
B. -1, 0, -1
C. -1, 0, 1
D. 1, 0, -1
E. It's implementation-defined.

C Strings

```
char s[6] = "abcde";
s[1] = 'f';
s[2] = 0;
printf("%s\n", s);
```

What does the code shown above print?

A.abcde
B.afcde
C.af0de
D.af0
E.af

C Strings

```
char s[6] = "abcde";
printf("%d, %d\n", strlen(s), sizeof(s));
```

What does the code shown above print?

A.5, 5
B.5, 6
C.6, 6
D.6, 5
E.None of the above

C Strings

```
int i = 1234567;
float f = 13579.24680;
char *s = "abcdefgh";
printf("%4d, %6.3f, %5s\n", i, f, s);
```

What does the code shown above print?

A.1234567, 13579.2468, abcdefgh
B.4567, 13579.247, defgh
C.1234, 579.247, abcde
D.1234567, 13579.247, abcdefgh

C Formatters

```
int i = 1234, j = 5678;
foo(&i, &j);
printf("%d, %d\n", i, j);
```

What does the code print, if the function it calls is defined as follows:

```
void foo(int *p, int *q)
{
    int *r = p; p = q; q = r;
    *p = 1357; *q = 2468;
}
```

A.1234, 5678
B.5678, 1234
C.1357, 2468
D.2468, 1357

Pointer Params