

## CS61C Fall 2013 – 2 – Practice With C

---

### Pointers

1. What is terrible about the following function?

```
int *blasphemy(void) {  
    int x;  
    return &x;  
}
```

2. What is wrong about the following function that swaps the values of two int variables? What changes need to be made for swappy to function correctly?

```
void swappy( int a, int b) {  
    int temp = a;  
    a = b;  
    b = temp;  
}
```

3. What does the following function do?

```
int mystery(int *arr, int n) {  
    return n ? arr[0] + mystery(arr + 1, n - 1) : 0;  
}
```

### Bitwise Operators

C provides bitwise commands for AND (&), OR (|), XOR (^), and NOT (~). Ignoring NOT for now, let's see what happens when we reduce the 2-input gates to 1-input gates by fixing the second input.

1. Let  $x$  be the input. Fill in the following blanks with either 0, 1,  $x$ , or  $\bar{x}$  (NOT  $x$ ):

$x \& 0 = \underline{\hspace{2cm}}$      $x | 0 = \underline{\hspace{2cm}}$      $x \wedge 0 = \underline{\hspace{2cm}}$

$x \& 1 = \underline{\hspace{2cm}}$      $x | 1 = \underline{\hspace{2cm}}$      $x \wedge 1 = \underline{\hspace{2cm}}$

2. Based on your responses, look at the columns (grouped by operation) above. Which operation would be useful for turning bits OFF? For turning bits ON? For flipping bits?

# CS61C Fall 2013 – 2 – Practice With C

---

## C Programming Practice

Complete the implementation of the following functions based on the comments.

1. Increments the value of an int outside this function by one.

```
void increment( int *x ) {  
  
}
```

2. Returns the number of bytes within a string. Do not use strlen().

```
int mystrlen( char* str ) {  
  
  
}
```

3. Returns the number of elements in an array \*arr of ints. The array must be able to store any integer that fits in the array.

```
int arraylen( int array[] ) {  
  
  
}
```

## Structs

Structs are user-defined collections of variables. A structure definition goes as follows:

```
struct structure_tag {  
    type1 member1;  
    ...  
    typen membern;  
};
```

What does each of the four following statements do?

```
struct {int x; int y;} var;  
struct point {int x; int y;};  
struct point {int x; int y;} pt1;  
struct point {int x; int y;} pt1 = {1,2};
```