

Exam 2, Fall 2015

Here are answers without solutions. Determine why these answers are correct.

1. Pointer and Integer Arithmetic

base address: 100
sold: 168
copy: 128
paint: 208

2. Memory Map

| label | address | value |
|-----------------------|---------|-------------------|
| i | 1000 | 1 |
| race.doors | 1004 | 5 12 (4) (9) |
| race.body.trim[0] | 1008 | 7 (4) |
| race.body.trim[1] | 1012 | 9 (4) |
| race.link | 1016 | 1052 1036 (4) (5) |
| decal[0] | 1020 | 2 13 (8) |
| decal[1] | 1024 | 3 |
| decal[2] | 1028 | 4 |
| toy | 1032 | 1036 (3) |
| sport[0].doors | 1036 | 5 |
| sport[0].body.trim[0] | 1040 | 7 |
| sport[0].body.trim[1] | 1044 | 9 11 (6) |
| sport[0].link | 1048 | 1052 (1) |
| sport[1].doors | 1052 | 6 12 (7) |
| sport[1].body.trim[0] | 1056 | 8 |
| sport[1].body.trim[1] | 1060 | 10 |
| sport[1].link | 1064 | 1068 (2) |

3. Dynamic Memory for Structures

Very similar to homework problem.

```
struct Book *ece222;  
ece222 = (struct Book *) malloc(sizeof(struct Book));  
ece222->author = (struct Faculty *) malloc(sizeof(struct Faculty));  
ece222->author->name = (struct Person *) malloc(sizeof(struct Person));  
ece222->author->name->first = (char *) malloc((strlen("Adam")+1)*sizeof(char));  
ece222->author->name->last = (char *) malloc((strlen("Hoover")+1)*sizeof(char));  
  
strcpy(ece222->title, "System Programming");  
strcpy(ece222->author->name->first, "Adam");  
strcpy(ece222->author->name->last, "Hoover");  
strcpy(ece222->author->rank, "Professor");  
strcpy(ece222->author->degree, "Ph.D.");
```

4. Magnitude

```
double magnitude( struct Complex c )
{
    return sqrt( pow(c.real, 2) + pow(c.imag, 2) );
}
```

5. Maximum Waveform Magnitude

```
double max_mag( struct Signal wav, int *index )
{
    double candidate = magnitude ( wav.samples[0] );
    int candidate_pos = 0, int i;
    for (i = 1; i < wav.num_samples; i++) {
        if ( magnitude (wav.samples[i]) > candidate) {
            candidate = magnitude( wav.samples[i] );
            candidate_pos = i;
        }
    }
    *index = candidate_pos;
    return candidate;
}
```

An alternative solution could pass in a pointer to the waveform instead of a copy of the Signal structure.

6. Copy and Smooth Filtering Function

Critical parts of this problem is to understand how the return value works and why two mallocs are required. Also, to demonstration when to use "." and "->".

```
struct Signal *filter( struct Signal wav)
{
    struct Signal *newwave = (struct Signal *) malloc(sizeof(struct Signal));
    int j, num = wav.num_samples;
    newwave->num_samples = num;
    newwave->samples = (struct Complex *) malloc(num * sizeof(struct Complex));
    for( j = 0; j < num; j++ ) {
        if (j == 0 || j == num - 1 )
            newwave->samples[j] = wav.samples[j];
        else {
            newwave->samples[j].real = (wav.samples[j-1].real
                                      + wav.samples[j].real
                                      + wav.samples[j+1].real) / 3.0;
            newwave->samples[j].imag = (wav.samples[j-1].imag
                                       + wav.samples[j].imag
                                       + wav.samples[j+1].imag) / 3.0;
        }
    }
    return newwave;
}
```