# ***MORE PRACTICE QUESTIONS FOR***
# CS16 Midterm Exam 2

# (Practice Exam 02a—printer friendly version)
# E02, 09F, Phill Conrad, UC Santa Barbara
# Actual exam scheduled for: 11/18/2009

---

1. (15 pts) Suppose that you are performing the selection sort algorithm—as explained in lecture, in our "field day" exercise, and in the code for lab06, i.e.

   - After the entire sort, elements will be in order from smallest to largest
   - Each pass through the array "selects" the largest element from the smaller array, and swaps it into place (sometimes "swapping with itself" if it is already in place)

   With that in mind, indicate the values of each element of the array after the 1st, 2nd and 3rd swap.

   Keeping in mind that some of these swaps may be cases of an element being swapped with itself—*those still "count" as swaps!*

| | a[0] | a[1] | a[2] | a[3] | a[4] |
|---|---|---|---|---|---|
| initial values | 27 | 6 | 42 | 9 | 17 |
| after 1st swap | | | | | |
| after 2nd swap | | | | | |
| after 3rd swap | | | | | |
| after 4th swap (final array) | 6 | 9 | 17 | 27 | 42 |

2. Suppose there is an array of integers a, and we want to swap element a[i] with element a[j], as in the following contrived example:

```
int main()
{
   int a[5]={15,35,25,30,20};
   int i, j;
   int temp = 0; // iniialized to zero for no good reason.


   // initialize i and j
   i = 1;
   j = 4;

   printf("Before: a[i] = %d  a[j]=%d \n",a[i],a[j]);

   // Now swap a[i] with a[j]


      some code goes here...



   // a[i] and a[j] are now swapped!



   printf("After:  a[i] = %d  a[j]=%d \n",a[i],a[j]);


}
```

If the swap is correct, the output will be:

```
Before: a[i]=35 a[j]=20
After:  a[i]=20 a[j]=35
```

Below and on the next page are several attempts at writing this swap. In some cases, there is a function definition that goes along with the code.

In each case, is the swap is correct, simply circle "CORRECT". If it is incorrect, fill in what will be printed "after" the incorrect swap is run.

    a. (4 pts) Circle CORRECT, or fill in the values: After: a[i] = _____ a[j] = _____

| **function definition goes earlier in file** | **put this at** *some code goes here* |
|---|---|
| <pre>void swap(int a, int b)<br>{<br>   int temp;<br>   temp = a;<br>   a = b;<br>   b = temp;<br>}</pre> | `swap(a[i],a[j]);` |

b. (4 pts) Circle CORRECT, or fill in the values: `After: a[i] = _____ a[j] = _____`

| function definition goes earlier in file | put this at *some code goes here* |
|---|---|
| ```
void swap(int *a, int *b)
{
    int temp = 0;
    temp = (*a);
    (*a) = (*b);
    (*b) = temp;
}
``` | ```
swap(&a[i],&a[j]);
``` |

c. (4 pts) Circle CORRECT, or fill in the values: `After: a[i] = _____ a[j] = _____`

| function definition goes earlier in file | put this at *some code goes here* |
|---|---|
| ```
(None)
``` | ```
a = temp;
temp = b;
b = a;
``` |

d. (4 pts) Circle CORRECT, or fill in the values: `After: a[i] = _____ a[j] = _____`

| function definition goes earlier in file | put this at *some code goes here* |
|---|---|
| ```
None
``` | ```
temp = a;
a = b;
b = temp;
``` |

e. (4 pts) Circle CORRECT, or fill in the values: `After: a[i] = _____ a[j] = _____`

| function definition goes earlier in file | put this at *some code goes here* |
|---|---|
| ```
void swap(int *a, int i, int j)
{
    int temp = 0;
    temp = a[i];
    a[i] = a[j];
    a[j] = temp;
}
``` | ```
swap(a,i,j);
``` |

# End of Extra Practice Questions

**Total Points: 35**