

Lab 3 Worksheet

1a. Pointer Switcheroo! What is the output of the following C code?

```
#include <stdio.h>

void change(int *p1, int *p2) {
    *p1 = *p2;
    p1 = p2;
    *p1 = 30;
}

int main() {
    int a = 3410, b = 3420;
    int *pa = &a;
    int *pb = &b;

    change(pa, pb);
    printf("%d %d\n", a, b);
}
```

1b. Pointer Switcheroo x 1000! What are the values for each of the variables at the end of this program?

```
#include <stdio.h>

void mystery(int *p1, int *p2) {
    int *temp;
    temp = p2;
    p2 = p1;
    p1 = temp;
}

void mystery2(int *p1, int *p2) {
    int temp = *p1;
    *p1 = *p2;
    *p2 = temp;
}

int main() {
    int a = 3410, b = 3420, c = 2800, d = 3110, e = 1100, f = 2110;
    int *pa = &a, *pb = &b, *pc = &c, *pd = &d;
```

```

    if (b == 3420)
        mystery(pa, pb);

    if (*pc > e)
        mystery2(&a, &e);

    if (d < c)
        mystery2(pa, pc);

    if (a > f)
        mystery(pc, pd);

    mystery2(pc, pd);
}

```

2. Stack vs Heap! For each variable in this function, indicate whether it's stored on the stack or heap:

```

#include <stdlib.h>

void process_data() {
    int x = 3410;
    int* ptr = malloc(sizeof(int));
    char message[] = "Hi";
    double y = 3.41;
    char* name = "World";
    int arr[3] = {1, 2, 3};
    float* buffer = malloc(5 * sizeof(float));

    *ptr = 10;
    buffer[0] = 1.5;

    free(ptr);
    free(buffer);
}

```

```
int main() {
    process_data();
    return 0;
}
```

Variable	Storage Location
x	
ptr	
*ptr	
message	
y	
name	
arr	
buffer	
buffer[0-4]	

3. Memory Layout: In which memory segment is each of these values located? The options are: the stack, the heap, the text segment (a.k.a. the code segment), and the data segment (a.k.a. the constant or global segment).

```
#include <stdio.h>
#include <stdlib.h>

const int ARRAY_SIZE = 6;

void findMax(int* data, int n, int* maxVal) {
    int current_max = data[0];
    for (int i = 1; i < n; ++i) {
        if (data[i] > current_max) {
            current_max = data[i];
        }
    }
    *maxVal = current_max;
}

int main() {
    int numbers[ARRAY_SIZE] = {15, 3, 42, 8, 27, 11};
    int* buffer = malloc(sizeof(int));
    findMax(numbers, ARRAY_SIZE, buffer);
    printf("Maximum: %d\n", *buffer);
    free(buffer);
    return 0;
}
```

ARRAY_SIZE	stack	heap	text	data
numbers[2]	stack	heap	text	data
current_max	stack	heap	text	data
*buffer	stack	heap	text	data
n	stack	heap	text	data
findMax	stack	heap	text	data