

Exercise 6-3: swap.c

Maximilian Fernaldy - C2TB1702

Note: some links and other HTML-related objects may not work in pdf form. Consider reading the webpage format of the report [here](#).

swap.c demonstrates a common useful function that swaps the value of two variables using their pointers. Instead of copying both their values and having to return two variables (which is impossible in C), the function will work directly with the variables and return nothing (which means it will use the `void` return type specifier).

```
// Use method explained in report 5-2 to swap values
void swap(int *x, int *y) {
    *x += *y;
    *y = *x - *y;
    *x -= *y;
}
```

As previously explained in [report 5-2](#), the method above swaps the values of two variables without having to allocate more memory. The difference is that this time, instead of using `a` and `b` explicitly, we use their memory addresses to modify them. Once the function is done, the values in `a` and `b` will have been swapped, without having to return any value from the function itself.

```
int main() {
    int a, b;
    // Input
    printf("Enter 2 integers separated by a space: ");
    scanf("%d %d", &a, &b);
    // Exchange the contents
    swap(&a, &b);
    // Display results
    printf("%d %d\n", a, b);
    return 0;
}
```

As noted in report 6-2, functions that use pointers to modify variables should have memory addresses (or pointers) passed into them, not the variables themselves. This is to avoid a type error when compiling.

Running the program gives us the following output:

```
max@victus16:~/repos/pip/lec06/ex6-3
max ex6-3 $ cat swap.c
#include <stdio.h>
/* prototype declaration */
void swap(int *x, int *y);

int main() {
    int a, b;
    // Input
    printf("Enter 2 integers separated by a space: ");
    scanf("%d %d", &a, &b);
    // Exchange the contents
    swap(&a, &b);
    // Display results
    printf("%d %d\n", a, b);
    return 0;
}

// Use method explained in report 5-2 to swap values
void swap(int *x, int *y) {
    *x += *y;
    *y = *x - *y;
    *x -= *y;
}

max ex6-3 $ ./swap
Enter 2 integers separated by a space: 8 10
10 8
max ex6-3 $
```