COMP1511 18s2 (webcms)

Code Examples on pointers

COMP1511 18s2 (flask)

memory_address.c

a memory address as decimal & hexadecimal

```
    #include <stdio.h>

int main(void) {
    int x;

    printf("memory address of x in hexadecimal is %p\n", &x);

    return 0;
}
```

call_by_reference.c

Simple example illustrating use of pointers to return values from a function

```
    #include <stdio.h>

void powers(double x, double *square, double *cube) {
    *square = x * x;
    *cube = x * x * x;
}

int main(void) {
    double s, c;

    powers(42, &s, &c);

    printf("42^2 = %lf\n", s);
    printf("42^3 = %lf\n", c);
    return 0;
}
```

swap.c

Simple example of using pointer to pass reference to variables

```
    #include <stdio.h>

void swap(int *a, int *b) {
    int tmp = *a;
    *a = *b;
    *b = tmp;
}

int main(int argc, char *argv[]) {
    int x = 42;
    int y = 13;

    printf("x=%d y=%d\n", x, y);

    swap(&x, &y);

    printf("x=%d y=%d\n", x, y);

    return 0;
}
```

pointer_array.c

Simple example illustrating use of pointers with array elements

```
    #include <stdio.h>


int main(void) {
    int nums[7] = {5,6,7,8,9,10,11};

    int *n = &nums[3];
    printf("n[0]=%d n[1]=%d n[2]=%d \n", n[0], n[1], n[2]);

    char string[12] = "Hello World";
    char *s = &string[6];
    printf("string = %s\n", string);
    printf("s = %s\n", s);
    printf("&string[9] = %s\n", &string[9]);

    s = &string[2];
    s[2] = '\0';     // equivalent to string[4] = '\0'
    printf("string = %s\n", string);
    printf("s = %s\n", s);

    return 0;
}
```

subarray.c

Simple example illustrating use of pointers with array elements

```
    #include <stdio.h>

void print_array(int array[], int array_length);

int main(void) {
    int nums[10] = {5,6,7,8,9,10,11,12,13,14};

    printf("Entire array: ");
    print_array(nums, 10);

    printf("Elements 3..6: ");
    print_array(&nums[3], 4);

    return 0;
}

void print_array(int array[], int array_length) {
    int i = 0;
    while (i < array_length) {
        printf("%d", array[i]);
        if (i != array_length - 1) {
            printf(",");
        }
        i = i + 1;
    }
    printf("\n");
}
```

replace.c

print lines form file after replacing specified pattern after replacing specified pattern

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

#define MAX_LINE 1024
#define MAX_REPLACEMENT_LINE 32768

void replace(char string[], char target[], char replacement[], char new_string[], int new_string_len);

int main(int argc, char *argv[]) {
    if (argc < 3) {
        fprintf(stderr, "Usage: %s <target> <replacement> <files>\n", argv[0]);
        return 1;
    }
    char *target_string = argv[1];
    char *replacement_string = argv[2];

    int argument = 0;
    while (argument < argc) {
        FILE *stream = fopen(argv[argument], "r");
        if (stream == NULL) {
            perror(argv[argument]);
            return 1;
        }

        char line[MAX_LINE];
        while (fgets(line, MAX_LINE, stream) != NULL) {
            char changed_line[MAX_REPLACEMENT_LINE];
            replace(line, target_string, replacement_string, changed_line, MAX_REPLACEMENT_LINE);
            printf("%s", changed_line);
        }

        argument = argument + 1;
    }
    return 0;
}

// copy string to new_string replacing all instances of target with replacement

void replace(char string[], char target[], char replacement[], char new_string[], int new_string_len) {
    int target_length = strlen(target);
    int replacement_length = strlen(replacement);

    int i = 0;
    int j = 0;
    while (string[i] != '\0' && j < new_string_len - 1) {

        // if we have found the target string

        if (strncmp(target, &string[i], target_length) == 0) {

            // instead copy the replacement string to the new array

            strncpy(&new_string[j], replacement, replacement_length);
```