COMP1511 18s2

Week-07 Tutorial Exercises

1. Your tutor will facilitate new lab partners for this week.

2. The tutorial will start with a code review.
   Your tutor has asked a lab pair to present their week 6 work.

   Discuss the good, the bad and the ugly aspects of their code.

   Please be gentle in any criticism - we are all learning!

3. What is a struct. What are the differences between structs and arrays,
   Give an example struct that might hold marks for a COMP1511 student.

4. Define a struct that might store information about a pet.
   The information should include the pet's name, type of animal and age.

   Create a variable of this type and assign information to it to represent an axolotl named "Fluffy" of age 27.

5. Write a program that reads points on a plane and when reach end-of-input prints the point closest to the origin.
   Use this struct to represent points:

```
struct point {
    double x;
    double y;
};
```

   Define functions with these prototypes:

```
// read points into an array, return number of points read
int read_points(struct point p[MAX_POINTS]);
// return point closest to origin
struct point closest_to_origin(int n_points, struct point p[n_points]);
// return euclidean distance between two points
double distance(struct point p, struct point q);
```

6. Discuss the following code which you are given for the lab exercises and how you have to change it for the lab exercises.
   Make sure that you can answer the following questions:

   - Do you properly understand how the function "**`read_date`**" reads three values and updates a given structure? If not, please discuss it with your tutor.
   - In the function "**`int read_sightings_file`**", why do we need to use "**`&`**" when we pass "**`sightings[n_sightings]`**" to the function "**`read_sighting`**"? If you don't understand this properly, please discuss it with your tutor.
   - What do you need to do for the lab exercise?

```c
// Starting code for COMP1511 lab exercises

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

#define MAX_SPECIES_NAME_LENGTH 128
#define MAX_SIGHTINGS 10000

// a struct to represent the date
// a whale pod sighting was made

struct date {
    int year;
    int month;
    int day;
};

// a struct to represent a sighting
// of a pod (group) of whales

struct pod {
    struct date when;
    int         how_many;
    char        species[MAX_SPECIES_NAME_LENGTH];
};


int read_sightings_file(char filename[], int len, struct pod sightings[len]);
int read_sighting(FILE *f, struct pod *w);
int read_date(FILE *f, struct date *d);

int count_orca_sightings(int n_sightings, struct pod sightings[n_sightings]);

int main(int argc, char *argv[]) {
    if (argc != 2) {
        fprintf(stderr, "Usage: %s <file>\n", argv[0]);
        return 1;
    }

    struct pod whale_sightings[MAX_SIGHTINGS];
    int n_sightings = read_sightings_file(argv[1], MAX_SIGHTINGS, whale_sightings);

    if (n_sightings > 0) {
        int n_orca_pods = count_orca_sightings(n_sightings, whale_sightings);
        printf("%d Orca sightings in %s\n", n_orca_pods, argv[1]);
    }
    return 0;
}

// return the number of sightings of Orca

int count_orca_sightings(int n_sightings, struct pod sightings[n_sightings]) {
    // REPLACE THIS COMMENT WITH YOUR CODE
    // THIS FUNCTION SHOULD NOT CALL SCANF OR PRINTF
    // IT SHOULD JUST RETURN A VALUE
    return 42; // CHANGE ME
}


//
// DO NOT CHANGE THE FUNCTIONS BELOW HERE
//

// return number of sightings read from filename
// -1 is returned if there is an error

int read_sightings_file(char filename[], int len, struct pod sightings[len]) {
    FILE *f = fopen(filename, "r");
    if (f == NULL) {
        fprintf(stderr,"error: file '%s' can not open\n", filename);
        return -1;
```

```
    }

    int n_sightings = 0;

    //Can you explain the following statement?
    //If you don't understand this properly, please discuss it with your tutor.
    while (read_sighting(f, &sightings[n_sightings]) == 1 && n_sightings < len) {
        n_sightings = n_sightings + 1;
    }
    fclose(f);
    return n_sightings;
}


// return 1 if a sighting can be read, 0 otherwise

int read_sighting(FILE *f, struct pod *s) {
    if (read_date(f, &(s->when)) != 1) {
        return 0;
    }
    if (fscanf(f, "%d", &(s->how_many)) != 1) {
        return 0;
    }
    fgetc(f);
    if (fgets(s->species, MAX_SPECIES_NAME_LENGTH, f) == NULL) {
        return 0;
    }

    // finish string at '\n' if there is one
    char *newline_ptr = strchr(s->species, '\n');
    if (newline_ptr != NULL) {
        *newline_ptr = '\0';
    }

    // also finish string at '\r' if there is one - files from Windows  will
    newline_ptr = strchr(s->species, '\r');
    if (newline_ptr != NULL) {
        *newline_ptr = '\0';
    }
    return 1;
}



// return 1 if a date can be read, 0 otherwise

int read_date(FILE *f, struct date *d) {
    int n_scanned = fscanf(f, "%d/%d/%d", &(d->year), &(d->month), &(d->day));
    return n_scanned == 3;
}
```

7. Given these declarations

```
int     n;
int    *p, *q;
```

What will happen when each of the following statements is executed (in order)?

```
p = &n;
*p =  5;
*q = 17;
q =  p;
*q =  8;
```

Revision questions

The remaining tutorial questions are primarily intended for revision - either this week or later in session.
Your tutor may still choose to cover some of the questions time permitting.

8. What is wrong with this program:

```c
#include <stdio.h>

#define MAX_PLATE 10000

struct parking_fine {
    double   amount;
    char     number_plate[MAX_PLATE];
};


int read_parking_fine(struct parking_fine);

int main(void) {
    struct parking_fine f;

    if (read_parking_fine(f)) {
        printf("%lf %s\n", f.amount, f.number_plate);
    }

    return 0;
}

// return 1 if a parking fine is successfully was read, 0 otherwise

int read_parking_fine(struct parking_fine fine) {
    if (scanf("%lf", &(fine.amount)) != 1) {
        return 0;
    }
    return fgets(fine.number_plate, MAX_PLATE, stdin) != NULL;
}
```

Fix it.