CS220 - Computer System II
Assignment 4 (100 points)

Due: 09/30/2016, 11:59pm

1 Instructions

- Answer the questions individually. Group effort is not allowed.
- Reading: `http://c-faq.com`, K&R C, chapter 1-6.

2 Questions

1. Within a file point.h, *define a type "Point" as a structure* that contains a point in space with x, y and z coordinates as floating point numbers. The structure also contains a pointer "next" to the next Point in space. The last point in space points to NULL as the next Point. Further, *declare prototypes* to functions: (1) `contains_loop` that accepts a pointer to Point and returns an integer. (2) `transform_points` that accepts 2 pointers to Points, and returns a pointer to a Point. That is: `Point* transform_points(Point *points, Point *origin)` Include appropriate comments. (20 points)

2. Within a file point.c, include point.h and *implement* `contains_loop`. This function accepts a pointer to a Point (although a single pointer is accepted, it is indeed a list because each point points to the next point in space until a NULL is reached) and detects if the list of points contain a loop. A series of Points contains a loop if the same point appears more than once during traversal from 1st to the last point in the list. Two points are equal if and only if their x, y and z components are equal. The function returns 1 if a loop exists, 0 otherwise. HINT: Eventually, a faster runner always laps a slower runner on a circular track. (40 points)

3. Within the file point.c, *implement* `transform_points`. This function accepts `points`, a pointer to the first Point in the list of points, and `origin`, a single point that becomes the new origin. Create a new list of points where each point has been transformed such that new values of x, y and z in the point are obtained by adding `origin`'s x, y and z components to the old values of x, y and z. NOTE: You are not allowed to modify the input `points`. You are to create a new instances of Point using malloc. (40 points)

3 How to test your code

You are provided with driver.o, a driver for your code. Once you implement point.h and point.c, compile your code using:

```
$ gcc point.c driver.o -std=c89 -o main
```

Run main to run different test cases against your code. The tests will generate a large amount of output. You can copy it to a text file to analyze.

4 Submitting your code

Once your program passes all the tests, create assn4.tar.gz comprising of a Makefile, point.h, point.c and driver.o. Submit assn4.tar.gz on Blackboard.