COMP10120 Practical Set 4: Data Types & Pointers

Please read the questions carefully. Name each program based on your student number, the practical set number and question number. For this set (set4), question 1 should be named 1234567s4q1.c where your student number replaces 1234567. All questions that you are submitting can be zipped into a single file called 1234567s4.zip, where 1234567 is your student number and s4 refers to set 4. This zipped file can be submitted via Moodle for grading.

1. Write a C Program which prints the size (in bytes) of the following variables to the screen. Depending on your system, you may get different answers to other students.
    - char
    - int
    - long
    - long long
    - double
    - long double
    - float
    - an array of integers
    - a pointer to an array

2. Write a C Program that mimics a two-horse race. The horses' names are *Firefoot* and *Shadowfox.* Each horse will start at the first grid/space on the race track. The finish line is at 100 hops/spaces away and the first horse to reach the end wins the race! A **loop** will control how far each horse moves on each iteration by using random number generation according to the rules below (ending when a horse reaches or passes space 100 on the race track):

    A. 50% of the time, the horse will progress at full speed (2 spaces).
    B. 10% of the time, the horse will only progress 1 space.
    C. 10% of the time, the horse will progress 3 spaces.
    D. 10% of the time, jockey will fall off and the horse will not move at all.
    E. 20% of the time, the horse will jump backwards 2 spaces.

    To achieve these rules, generate a random integer between 1 and 10. If the integer is between 1 and 5, then use option A (move 2 spaces forward). Do this conditional check for the other possibilities too (e.g. if the number generated is 6 do option C because 6 will occur approximately 10% of the time in the random number generation). Checks need to be included in case a horse happens to go backwards past the first element (they should be reset to the first element space, likewise if the horse goes past the end of the track, move it back to the last position.

    You need to use **variables** to keep track of the position of the horses (i.e. their progress and position on the track). On each iteration of the loop, each horse and jockey will progress 1, 2 or 3 spaces, not progress at all or move back 2 spaces, a **variable** for each horse will control this.

The program should print the location of each horse at every loop iteration. If the two horse are in the same space, print a T for 'Tied'. Finally, when one horse reaches the last space, print out which horse won. If the two horses reach the last element on the same iteration of the loop, print out 'Race Tied'.

There is skeleton some code to support this program on Moodle (horse_race_skeleton.c), you need to fill in the missing parts which are highlighted with ^^ in the comments. Place emphasis on the use of pointers to store the location of the horses and the importance of them in this program.

Pay particular attention to how the program uses pointers instead of global variables.