## CS 288 2018S Section 002
## Homework 01

**Due**: At the beginning of class on Monday February 5th, 2018.

The Tortoise and the Hare; adapted from the "Java How to Program" by Deitel & Deitel. In this problem you will recreate the classic race of the tortoise and the hare. You will use random number generation to develop a simulation of this memorable event.

Our contenders begin the race at "square 1" of 70 squares. Each square represents a possible position along the race course. The finish line is at square 70. The first contender to reach or pass square 70 is rewarded with a pail of fresh carrots and lettuce. The course weaves its way up the side of a slippery mountain, so occasionally the contenders lose ground.

There is a clock that ticks once per second. With each tick of the clock, your program should adjust the position of the animals according to the following rules:

| Animal | Move Type | Percentage of Time | Actual Move |
|---|---|---|---|
| Tortoise | Fast plod | 50% | 3 squares to the right |
| | Slip | 20% | 6 squares to the left |
| | Slow plod | 30% | 1 square to the right |
| Hare | Sleep | 20% | No move at all |
| | Big hop | 20% | 9 squares to the right |
| | Big slip | 10% | 12 squares to the left |
| | Small hop | 30% | 1 square to the right |
| | Small slip | 20% | 2 squares to the left |

Use variables to keep track of the positions of the animals. Start each animal at position 1 (i.e., the "starting gate"). If an animal slips left before square 1, move the animal back to square 1.

Generate the percentages in the preceding table by producing a random integer, i, in the range $1 \le i \le 10$. For the tortoise, perform a "fast plod" when $1 \le i \le 5$, a "slip" when $6 \le i \le 7$, or a "slow plod" when $8 \le i \le 10$. Use a similar technique to move the hare.

Begin the race by printing:
```
BANG !!!!!
AND THEY'RE OFF !!!!!
```

For each tick of the clock (i.e. each repetition of a loop), print a 70-position line showing the letter `T` in the tortoise's position and the letter `H` in the hare's position. Occasionally, the contenders will land on the same square. In this case, the tortoise bites the hare and your program should print `OUCH!!!` beginning at that position. All print positions other than the `T`, the `H`, or the `OUCH!!!` (in case of a tie) should be blank.

After printing each line, test if either animal reached or passed square 70. If so, print the winner and terminate the simulation. If the tortoise wins, print `TORTOISE WINS!!! YAY!!!` If the hare wins, print `HARE WINS. YUCH.` If both animals win on the same clock tick, print `IT'S A TIE.` If neither animals wins, perform the loop again to simulate the next tick of the clock.

### Hints and Tips:

Break the actions into functions. For example, functions like

```
int MoveTortoise();
int MoveHare();
void PrintCurrentPositions(int tortoisePosition, int harePosition);
```

The `rand()` C Standard Library Function generates random integers. Its prototype is defined in `<stdlib.h>` header file. Use the expression `((rand() % 10) + 1)` to generate random integers between 1 and 10.