CS-1412-501: Programming Principles II

Summer I Semester 2017
Department of Computer Science
Texas Tech University

Project

Due Date: At the end of the day of 07/05/2017
Total Points: 100 points

Please write the codes in separate files and zip all the .c or .cpp extension files in a single folder and upload it in your blackboard before the next lab begins. The submission after the due date and time will be considered as late.

The project submission will not be accepted after 07/05/2017, until there is a valid reason for doing so.

In this problem, you'll recreate one of the truly great moments of our very own childhood tale, namely the classic race of the tortoise and the hare. You'll 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's 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 rules as follows:

Animal	Move type	Percentage of the 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 (i.e., position numbers are 1–70). 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 !!!!!

Then, for each tick of the clock (i.e., each repetition of a loop), print a 70-position line showing the letter T in the position of the tortoise and the letter H in the position of the hare. 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 as shown in Figure 1 and 2 below. You can print S as the start position and F as the finish position.

```
BANG !!
AND THEY'RE OFF!!

SH.T.....|.........|.........|.........|.........|.........|.........F
Tortoise: 4
Hare: 2



S.H.T....|.........|.........|.........|.........|.........|.........F
Tortoise: 5
Hare: 3



H....T...|.........|.........|.........|.........|.........|.........F
Tortoise: 6
Hare: 1



H.....T..|.........|.........|.........|.........|.........|.........F
Tortoise: 7
Hare: 1
```

Figure 1

```
S........|.....T...|.H.......|.........|.........|.........|.........F
Tortoise: 16
Hare: 22



S........OUCH.........|.........|.........|.........|.........|.........F
Tortoise: 10
Hare: 10
```

Figure 2

After each line is printed, test whether either animal has reached or passed square 70. If so, then print the winner and terminate the simulation. If the tortoise wins, print TORTOISE WINS!!! If the hare wins, print HARE WINS!!! As shown in Figure 3.

If both animals win on the same tick of the clock, print IT'S A TIE!!!

If neither animal wins, perform the loop again to simulate the next tick of the clock.

Figure 3