Palomar College
Computer Science &
Information Technology
CSCI 112 Programming Fundamentals I
Lab 6 – Pointers & Modular Programming

1. Team Assignment – Instructions in Canvas as to team assignments.

Simulation: The Tortoise and the Hare
In this problem, you'll recreate one of the truly great moments in history, 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. The finish line is at square or value 70. The first contender to reach or pass 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 and print the updated position of each contender.

Rules of Play

Animal	Move type	Percentage of the time	Actual Move
Tortoise	Fast Plod	50%	3 squares forward
	Slip	20%	6 squares backward
	Slow Plod	30%	1 square forward
Hare	Sleep	20%	No move at all
	Big hop	20%	9 squares forward
	Big slip	10%	12 squares backward
	Small hop	30%	1 square forward
	Small slip	20%	2 squares backward

Use variables to keep track of the positions of the animals (position numbers are 1 to 70). Start each animal at position 1. Generate the percentages in the table by producing a random integer I between the values of 1 to 10. For the tortoise, perform a "fast plod" when i is between 1 and 5, a "slip" for 6 and 7, "slow plod" for 8, 9 and 10. Use a similar technique to move the hare.

Begin the race by printing something fun like:

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

Then for each tick of the clock (iteration 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 and on the same square. In this case, the tortoise bites the hare and your program should print OUCH!!! beginning at the tied position. All print positions other than the T, H, or the OUCH!!! should be blank.

After each line is printed, test whether either animal has reached or passed square 70. If so, the game is over and determines and prints the winner. Modular programming, pass by value, pass by reference, function definitions and main listed below must be used.

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

//function prototypes
void printInstructions();
void moveTortoise(int *turtle);
void moveHare(int *hare);
void printCurrentPosition(int turtle, int hare);
void results(int turtle, int hare);

void main()
{
 srand(time(NULL));

 int turtle = 1, hare = 1, timer = 0;

 printInstructions();

 while (turtle < 70 && hare < 70)
  {moveTortoise(&turtle);
   moveHare(&hare);
   printCurrentPosition(turtle, hare);
   timer++;
   }

 results(turtle, hare);
}
```