8.12 (*Simulation: The Tortoise and the Hare*) In this exercise, you'll re-create 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 is a clock that ticks once per second. With each tick of the clock, your program should use function `moveTortoise` and `moveHare` to adjust the position of the animals according to the rules in Fig. 8.21. These functions should use pointer-based pass-by-reference to modify the position of the tortoise and the hare.

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

Fig. 8.21 | Rules for moving the tortoise and the hare.

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 $1 \leq i \leq 10$. For the tortoise, perform a "fast plod" when $1 \leq i \leq 5$, a "slip" when $6 \leq i \leq 7$ or a "slow plod" when $8 \leq i \leq 10$. 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 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 whether either animal has 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, you may want to favor the tortoise (the "underdog"), or you may want to print `It's a tie.` If neither animal wins, perform the loop again to simulate the next tick of the clock.

Please place race.h, step1.cpp, step2.cpp, and step3.cpp in the same folder. The following steps need to be taken.

1. Please complete function void printCurrentPositions(const int * const snapperPtr, const int * const bunnyPtr) in step1.cpp to print out the positions of the tortoise and the hare as shown below.

```
---------|---------|---------|---------|---------|---------|---------|
                   H
         T
```

2. Please complete function void moveTortoise(int * const turtlePtr) in step2.cpp to simulate movements of the tortoise (that is, to change the position of the tortoise) according to the rules in Fig 8.21. The function sleep(n) in main will pause the program n microseconds as well as Clr() will clear the screen. Please copy the function printCurrentPositions from step1.cpp to complete this task.

3. Please complete function void moveHare(int * const rabbitPtr) in step3.cpp to simulate movements of the hare (that is, to change the position of the hare) according to the rules in Fig 8.21. Please copy the function completed in step1.cpp and step2.cpp to complete this task.

4. After the while loop is terminated, print the winner and the total clock ticks (seconds) the race elapsed, as shown below.

```
---------|---------|---------|---------|---------|---------|---------|
                   H
                                                              T


TORTOISE WINS!!! YAY!!!
TIME ELAPSED = 64 seconds
```

```
---------|---------|---------|---------|---------|---------|---------|
                                                              H
      T
Hare wins. Yuch.
TIME ELAPSED = 63 seconds
```