Questions on April 18, 2024

1. I do not understand Road.cpp and how it is compared to Hare.cpp.

   Answer: Road.cpp is the definition of Road class. Hare.cpp is the definition of Hare class. A hare can run on a road.

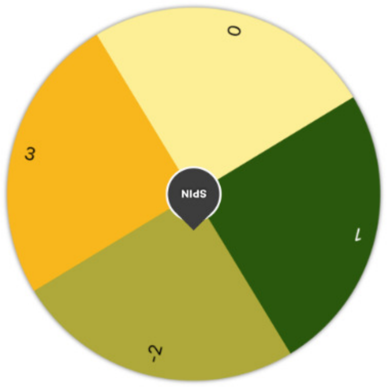
   (a) A hare has the following data members.

   i. move pattern represented by a vector of integers. For example, given {1, -2, 3, 0}, the hare moves 1 step forward in 25% of the time (1 out of 4 entries of the vector is 1), move 2 steps backward in 25% of the time (1 out of 4 entries of the vector is -2), move 3 steps forward in 25% of the time (1 out of 4 entries of the above vector has value 3), sleep (aka do not move) in 25% of the time (1 out of 4 entries has value 0).

   ii. current position represented by an integer.

   (b) A hare has the following methods

   i. void move(), update the value of data member position after the animal moves. The spin wheel is created using `https://pickerwheel.com/`.



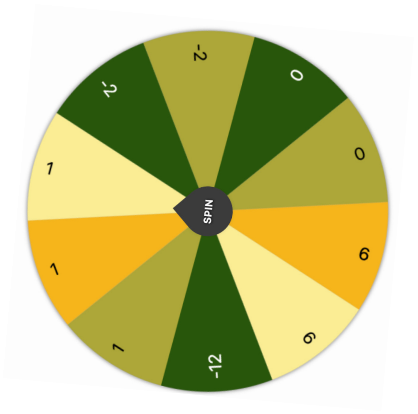
   In the above example,

   A. Spin the wheel. This step is equivalent to select a random integer in [0,3], which represents the index starting from 0 and ending at 3, since there are four elements 1, -2, 3, and 0 in data member `patters` of the current hare. Note that the last index of vector `patterns` is one fewer than the size of `patterns`.

   B. Choose the corresponding value pointed by the spin header. This step is equivalent to find out the value of `patterns` at the chosen index. That value is the number of steps the animal moves.

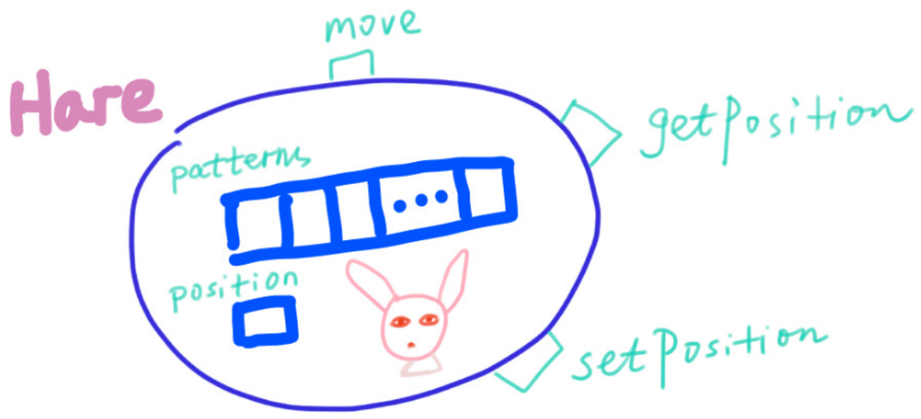
   C. Update data member `position` after the animal moves that many steps (can be negative, since an animal can move backwards).
   Note that the number of options in a spin wheel depending on data member `patterns`. For a default hare, which sleeps 20% of the time, move forward 9 steps 20% of the time, move backward 12 steps 10% of the time, move 1 step forward 30% of the time, and move 2 stesps

backward 20% of the time, its data member `patterns` is represented by the following spin wheel.



ii. void setPosition(int position), reset the value of the data member position by formal parameter position.

iii. int getPosition() const, return the value of data member position.



(c) A road object represents a sequence of blocks where animals run on. We may think when an animal steps on a block, it leaves its foot prints on that block. So, a road is represented by a vector of characters.

A road has data member `blocks`, defined as a vector of chars.

Operations (aka methods) on the data member of road class are as follows.

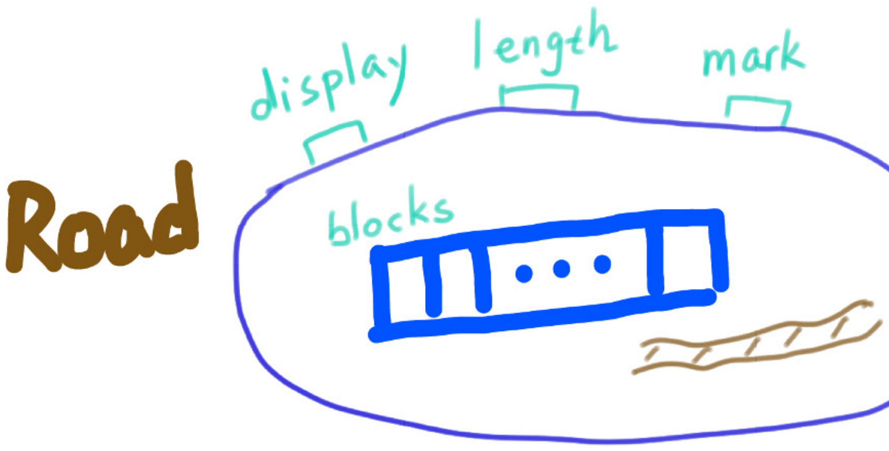
i. int length() const returns the number of `blocks` of the road.

ii. void mark(int index, char ch) puts ch to (index)th block of `blocks`.

iii. void display() const, prints the value of each block in a row, then followed by a new line.
For example, suppose a road has 10 blocks. If a hare jumps to the 2nd block and leaves a ‘H’ mark, and a tortoise moves to the 9th block and leaves a ‘T’ mark, then the road looks like the following, we use the half-box character to represent a space character:

```
␣␣H␣␣␣␣␣␣T
```

2. What is mark method of Road class and how it is used?



Answer: Suppose a road consists of several blocks (or squares) and is represented by a vector (or an array) of characters. When an animal runs on the road, it sets its feet on a block of the road and leaves a character representing the animal. For example, a hare leaves mark 'H' while a tortoise leaves mark 'T'.

Given an index and a character, Method `mark` changes the specified block of a road to be that character.

3. How to define the play method?

   Answer: Think how a hare runs on a road.

   (a) Declare and initialize integer variable `round` to be 1.

   (b) Move the hare object.

   (c) The hare might move outside the road, there are two possibilities:

      i. If the hare is out of the leftmost block of road indexed at 0, that is, the position of the hare object is smaller than 0, move the hare to position ... (you fill in the dots).

      ii. Do similar things when the hare is out of the rightmost block of the road. Hints: how to find out the length of a road object? What is the index of the last block when the first block is indexed at 0?

   (d) After the previous steps to move and adjust the hare's position if necessary, the hare is on the road. The hare stomps on the road and leaves mark 'H'. How to update the status of the road object?

   (e) Print the value of round. Display the current status of the road using one of its method, what is that method?

   (f) Prepare for the next round by clearing the marks left by the animal.

   (g) Increase round by 1.

   (h) Go back to Step (b) until the hare reaches the last block of the road.

   (i) After mimicing the running of a hare on a road, we can add an tortoise object to the do the competition, which is what `play` method does.

   (j) Once at least one of the animals reach the last block of the road, declare the result, which is one of the following:

      i. a tie

      ii. hare wins

      iii. tortoise wins

4. Can you explain vectors more in terms of the project?

   Answer: You can think a vector is an intelligent array that can grow dynamically. A vector has collection of data and operations on those data.

   Just like other existing classes in C++ like `string` and `ifstream`, we do not need to know the implementation details of vector, all we need to do is to read the document in `https://cplusplus.com/reference/vector/vector/`. Read its examples on constructors and member functions. For this project, you need to know the following:

   (a) The size of a vector can be found by its size method. Read `https://cplusplus.com/reference/vector/vector/size/` and check the example.

   (b) Like an array, an element of a vector can be accessed using operator [], where index is put inside the square brackets. Read `https://cplusplus.com/reference/vector/vector/operator[]/`.

   (c) As in hare and tortoise competition project,

      i. In Hare or Tortoise class, data member `patterns` consists of a collection of integers, so we use a vector of integers to hold those numbers. We do not use an array, since an array in C++ is not a class and its size needs to be specified explicitly.

      ii. In Road class, a road consists of a collection of blocks. Each block has wet mud. In the beginning, each block is clean. When an animal stomps on a block, it puts a character on the block. As a result, we think a clean block holds a space character, a block stomped by a hare holds character 'H', and a block stomped by a tortoise holds character 'T'.
      Note that block is not a type in C++, so we abstract it as a character after the above discussion. So Road class has data member `blocks`, which is a vector of characters(abbreviated as char in C++).
      Sometimes a hare and tortoise might stop at the same block, then the block is first marked by 'H', then is changed to 'T'. In this case, we print "Ouch. Tortoise bites hare." after displaying the content of each block of the road.
      Hence, data member `blocks` is a collection of character and is represented by a vector of characters.

   (d) Here is an example to show one difference between array and vector.

      i. (Midterm S24 V1) Define function called tally, for an **array** of integers, calculate the number of positive integers, negative integers, and zeros, then return the maximum of these three numbers. For example, if the array has elements 1, -2, 3, 2, 0, the return is 3.
      The function header is as follows. Note that `int arr[]` and `int* arr` are the same in function header, since the name of an array is also the address of the first element of the array, while the address of the element of an array is saved in an int pointer.
      `int tally(int arr[], int size);` or
      `int tally(int* arr, int size);`

      ii. If we change the above problem as follows.
      Define function called tally, for an **vector** of integers, calculate the number of positive integers, negative integers, and zeros, then return the maximum of these three numbers. For example, if the array has elements 1, -2, 3, 2, 0, the return is 3.
      Then the function header changes to
      `int tally(vector<int> nums);`
      The number of elements in vector of integers `nums` can be found by `nums.size()` and does not need to be passed as a parameter.