AP® COMPUTER SCIENCE A
2013 SCORING GUIDELINES

Question 2: TokenPass

Part (a) `TokenPass` constructor **4 points**

Intent: *Create* `TokenPass` *object and correctly initialize game state*

+1 Creates instance variable `board` as `int` array of size `playerCount`

+1 Computes a random number between 1 and 10, inclusive, and a random number between 0 and `playerCount-1`, inclusive

+1 Initializes all entries in `board` with computed random value *(no bounds errors)*

+1 Initializes instance variable `currentPlayer` to computed random value

Part (b) `distributeCurrentPlayerTokens` **5 points**

Intent: *Distribute all tokens from* `currentPlayer` *position to subsequent positions in array*

+1 Uses initial value of `board[currentPlayer]` to control distribution of tokens

+1 Increases at least one `board` entry in the context of a loop

+1 Starts distribution of tokens at correct board entry

+1 Distributes next token (if any remain) to position 0 after distributing to highest position in board

+1 On exit: token count at each position in `board` is correct

Question-Specific Penalties

-2 (v) Consistently uses incorrect array name instead of `board`

-1 (y) Destruction of persistent data (`currentPlayer`)

-1 (z) Attempts to return a value from `distributeCurrentPlayerTokens`

AP® COMPUTER SCIENCE A
2013 CANONICAL SOLUTIONS

Question 2: TokenPass

Part (a):

```
public TokenPass(int playerCount)
{
    board = new int[playerCount];
    for (int i = 0; i < playerCount; i++){
        board[i] = 1 + (int) (10 * Math.random());
    }
    currentPlayer = (int) (playerCount * Math.random());
}
```

Part (b):

```
public void distributeCurrentPlayerTokens()
{
    int nextPlayer = currentPlayer;
    int numToDistribute = board[currentPlayer];
    board[currentPlayer] = 0;

    while (numToDistribute > 0){
        nextPlayer = (nextPlayer + 1) % board.length;
        board[nextPlayer]++;
        numToDistribute--;
    }
}
```

These canonical solutions serve an expository role, depicting general approaches to solution. Each reflects only one instance from the infinite set of valid solutions. The solutions are presented in a coding style chosen to enhance readability and facilitate understanding.