Save The Island

Description: Two robots battle to the death over control of the island.

Writing a Bot:

- Bots may be written in either c++ or java.
- In order to play the game a bot must listen on standard in for boards as described in the Board Syntax section below.
- Once a board is received the bot must compute a valid move to make.
  - A valid move must follow all specifications outlines in the Move Syntax and Rules section below.
  - Submission of an invalid move will result in immediate disqualification and the end of the game.
  - Once the move has been computed you must output it to standard out.
  - You must submit a move via standard out in a maximum of three seconds from the time Bot!Battle! sends the board to your bot.
  - If your bot fails to submit a move in 3 seconds, your bot will be disqualified and the game will end.
- Your bot must loop while input is still available. There is no explicit end game command sent, your bot process will simply be killed when the game is over.

Testing your Bot:

- To make debugging your bots easier, you may output arbitrary debugging information via your standard error stream. These messages will appear in the Standard Error scroll box on the Bot!Battle! Test Arena.

- In addition all of your bot's moves made will appear in the Standard Output scroll box on the Bot!Battle! Test Arena.

Board Syntax:

- The board is a semi-colon separated string with three distinct components described below.

  [player number];[tile array];[island]

- [player number]: Either '1' or '2', indicating that you're player 1 or player 2 respectively.
  - Note: This number will be sent along with every board, however your bot's player number will be fixed throughout the entire game so it's safe to parse and store this only once if desired.

- [tile array]: A string of 5 characters, each is a digit between '0' and '4' inclusive.
  - These are the tiles you may use for moving, retreating, defending or attacking your opponent as specified below. After you use a tile, that tile will be replaced with a new random value between 0 and 5 on the next turn.

- [island]: A string of 15 characters representing the island.
  - Each character in the island string is either '0', '1', or '2'.
  - There will be exactly one '1' and one '2' in this string, these indicate the positions of player 1 and player 2 respectively.
    '0's indicate empty spaces on the board.

Starting Board:

- At the start of the game:
    - player 1 is placed at position 0 on the island.
    - Player 2 is placed at position 14 on the island.
    - Both players receive a set of 5 randomly generated between 0 and 4 inclusive.

Board Example:

- Here is an example of a board that may be sent to player 1 during a game.

```
1;44140;000100020000000
```

- The following can be deduced from this board:
    - Player 1 has the tiles, [4, 4, 1, 4, 0]
    - Player 1 is at board position 3
    - Player 2 is at board position 7
    - The distance between player 1 and 2 is 4.

Move Syntax and Rules:

- **Moving forward:**

```
move;x
```

  - x is a tile value between 0 and 4 inclusive.
  - You must have this tile in your tile array in order to use it.
  - You may only include a single tile when moving, selecting multiple tiles as done when attacking your opponent will be flagged as invalid.
  - You may not move to or past the position of your opponent.

- **Retreating from your opponent:**

```
retreat;x
```

  - x is a tile value between 0 and 4 inclusive.
  - You must have this tile in your tile array in order to use it.
  - You may only include a single tile when retreating, selecting multiple tiles as done when attacking your opponent will be flagged as invalid.

- **Attacking your opponent:**

  attack;x...x

  - x is a tile value between 0 and 4 inclusive, and x is equal to the distance between the players on the island.
  - If you have more than one tile with value x in your tile array, you may include x as many times as you have it. This amplifies your attack power.
  - Attack Power = 3 * (Number of Attack Tiles – Number of Defend Tiles)

    - The <u>Attack Power</u> is the distance that the victim will fall back after the attack.

    - <u>Number of Attack Tiles:</u> The number of tiles included after the attack keyword in the move sent by the attacker.
      - Note this is the cardinality of the set of tiles sent, do not confuse this with the value of the tiles used. Attacking with 3 '4' tiles is equivalent in power to attacking with 3 '1' tiles.

    - <u>Number of Defend Tiles:</u> The number of matching tiles in the defender's tile array. The semantics of defense is described in detail below.

- **Defending an attack:**

  - If the victim of an attack has tiles with the same value that were used in the attack. The attack will be automatically fully or partially defended.

  - For example consider the case where bot 1 makes the move "attack;22", and bot1 and bot2 are 5 positions away from each other.

    - If bot 2 has two '2''s in its tile array, the attack will be fully defended.
      - Bot 2 will stay in its current position.
      - Bot 1 will fall back to its position before the attack.
      - All tiles used by both bot 1 and bot 2 during this turn will be replaced by new random tiles on the next turn.

    - If bot 2 only has one '2', then the attack will be partially defended.
      - Using the formula above, Bot 2 will fall back 3 * (2 – 1) = 3 spaces.

    - If bot 2 has no '2' tiles, then the attack will not be defended.
      - Using the formula above Bot 2 will fall back 3 * (2 – 0) = 6 spaces.