```cpp
class Nim {
private:
  bool result;
public:
  Nim(vll a) {
    ll x = 0;
    Loop(i, a.size()) x ^= a[i];
    if (x != 0) result = true;
    else result = false;
  }
  bool get_result() {
    return result;
  }
  string get_winner(string player1, string player2) {
    if (result) return player1;
    else return player2;
  }
};

// Grundy number pseudo code
class Grundy {
private:
  bool result;
  vi grundies;
  vi diff;
public:
  void make_grundies(int k) {
    // memoization
    if (grundies[k] != -1) return;
    else {
      set<int> s;
      Loop(j, diff.size()) {
        // transition rule
        int index = k - diff[j];
        if (index >= 0) {
          if (grundies[index] == -1) make_grundies(index);
          s.insert(grundies[index]);
        }
      }
      int c = 0;
      while (s.find(c) != s.end()) c++;
      grundies[k] = c;
      return;
    }
  }
  Grundy(vi states, vi diff) {
    Grundy::diff = diff;
    // calculate all possible grundy numbers
    int grundy_size = 1000;
    grundies = vi(grundy_size, -1);
    Loop(i, grundy_size) make_grundies(i);
    // decide the grundy number in each states
    vll x(states.size());
    Loop(i, states.size()) x[i] = grundies[states[i]];
    // return to Nim
    Nim *nim = new Nim(x);
    result = nim->get_result();
  }
  bool get_result() {
    return result;
  }
  string get_winner(string player1, string player2) {
    if (result) return player1;
    else return player2;
  }
};

```