```javascript

const game = {
  U: { L: [2, 6], C: [0, 4], R: [4, 4] },
  M: { L: [3, 3], C: [0, 0], R: [1, 5] },
  D: { L: [1, 1], C: [3, 5], R: [2, 3] },
}

const questionParts = {
  a: { player: 1, strategy: [ 1/6, 1/3, 1/2 ] },
  b: { player: 2, strategy: [ 1/6, 1/3, 1/2 ] },
  c: { player: 1, strategy: [ 1/4, 1/8, 5/2 ] },
  d: { player: 2, strategy: [ 1/3, 1/3, 1/3 ] },
  e: { player: 2, strategy: [ 1/2, 1/2, 0/1 ] },
}

function bestResponse({ game, strategy, player }) {
  let maxPayoff = -99999, brSet = []
  function updateBR({ payoff, s }) {
    if (payoff === maxPayoff) brSet.push(s)
    if (payoff > maxPayoff) { maxPayoff = payoff; brSet = [s] }
  }

  // eslint-disable-next-line default-case
  switch (player) {
    case 1:
      for (const s of ['U', 'M', 'D']) {
        const payoff =
          game[s].L[0] * strategy[0] +
          game[s].C[0] * strategy[1] +
          game[s].R[0] * strategy[2]
        updateBR({ payoff, s })

      }
      break
    case 2:
      for (const s of ['L', 'C', 'R']) {
        const payoff =
          game.U[s][1] * strategy[0] +
          game.M[s][1] * strategy[1] +
          game.D[s][1] * strategy[2]
        updateBR({ payoff, s })
      }
      break
  }
  return brSet
}

for (const letter of Object.keys(questionParts)) {
  const br = bestResponse({ game, ...questionParts[letter] })
  console.log(`${letter}) {${br.join(',')}}`)
}

/*
 * output:
 * a) {U}
 * b) {R}
 * c) {U}
 * d) {R}
 * e) {L,R}
 */
```