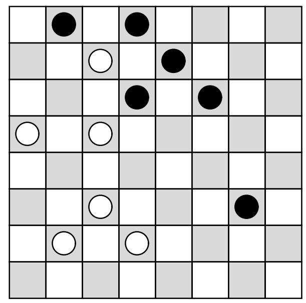
3. In the game of checkers (or *draughts* for some of you foreigners), one must always capture pieces if it is possible to do so, but when faced with more than one possible capture, one can choose any of the possible captures. A simple strategy, when faced with a choice of captures, is to capture so as to maximize the number of pieces you get over the number the opponent can capture on his next turn.

The game is played with black and white pieces on a chess board, using only the dark squares. For notational purposes, the squares are numbered as shown on the left:

	1		2		3		4
5		6		7		8	
	9		10		11		12
13		14		15		16	
	17		18		19		20
21		22		23		24	
	25		26		27		28
29		30		31		32	



We'll assume a simplified game in which there are no kings, so that the rules are as follows:

- The black pieces move down the board (toward larger numbers), the white pieces up.
- A piece A may capture (*jump*) a piece B of the opposite color, if B is on a (diagonally) adjacent square in the direction of A's motion *and* the square beyond B in the same direction is unoccupied. Piece A captures by jumping over B to the unoccupied square, and B is then removed from the board.
- If jumping over piece B causes A to land on a square from which another capture is possible, A must move again, capturing another piece. This continues until A moves to a square from which it cannot capture.
- When more than one piece of a given color can make a capture, exactly one of them captures; that same piece must continue capturing as long as it can.
- When a given piece has a choice of captures, it may make any one of them.

For example, suppose that it is Black's move in the configuration on the right above. Black has two possible jump sequences: either 2–9–18 (removing the two white pieces at 6 and 14 and moving the piece that was on 2 to 18) or 10–17 (removing the piece at 14 and moving the piece that was on 10 to 17). If Black chooses 2–9–18, White can respond with either 22–15–8 or 22–15–6. If Black chooses 10–17, White has no capture. So, if we look only at Black's move and White's next move, the 10–17 move gives the best gain (Black wins one piece, White none; when Black moves 2–9–18, on the other hand, White wins back two pieces, evening the score).

The input to your program will consist of a positive integer N_b followed by N_b integers in the range 1–32, giving the positions of the Black pieces, followed by a positive integer N_w and then N_w integers giving the positions of the white pieces. You may assume that the piece positions are all distinct and in the range 1–32. All input is in free form.

The output from your program should be the capture move sequence by Black that gives Black the greatest advantage (in relative numbers of pieces) after White's best next capture move. You may assume that Black will always have a capture move. Some moves by black may give White no capture to make, as in the example above, in which case Black's advantage for that move will simply be the number of pieces he captures. Otherwise, the advantage of a particular move by Black is the number of White pieces it captures, minus the largest number of Black pieces that White can capture from the resulting position. If there is more than one best move for Black, it doesn't matter which you report. The formats of input and output are illustrated below.

Example 1:

Input	Output

`6 1 2`
`7 10 11`
`24`
`6 6`
`13 14 22 25 26`
	`Best move sequence is 10-17`

Example 2:

Input	Output

`7 1 2 15`
`7 10 11`
`24`
`6 6`
`13 14 22 25 26`
	`Best move sequence is 2-9-18`