AP Computer Science A - Code Tracing Questions

ANSWER KEY

Question 1: `6 32` - The loop doubles y while 2*y <= 40 - y: 1 –> 2 –> 4 –> 8 –> 16 –> 32 - 2*32 = 64 > 40, so loop stops - x counts iterations: 6

Question 2: `[3, 7, 4, 1, 8, 9]` - The code swaps adjacent elements when left > right - Initial: [7, 3, 8, 4, 1, 9] - i=0: 7 > 3, swap –> [3, 7, 8, 4, 1, 9] - i=1: 7 < 8, no swap - i=2: 8 > 4, swap –> [3, 7, 4, 8, 1, 9] - i=3: 8 > 1, swap –> [3, 7, 4, 1, 8, 9] - i=4: 8 < 9, no swap

---

Question 3: `15` - Sum of diagonal elements (where r == c) - grid[0][0] = 2 - grid[1][1] = 4 - grid[2][2] = 9 - Sum = 2 + 4 + 9 = 15

Question 4: `1` - Trace of recursive calls: - mystery4(5) = mystery4(4) + mystery4(2) - mystery4(4) = mystery4(3) + mystery4(1) - mystery4(3) = mystery4(2) + mystery4(0) - mystery4(2) = mystery4(1) + mystery4(-1) - Base cases: mystery4(1) = 1, mystery4(0) = 0, mystery4(-1) = -1 - Working back up: - mystery4(2) = 1 + (-1) = 0 - mystery4(3) = 0 + 0 = 0 - mystery4(4) = 0 + 1 = 1 - mystery4(5) = 1 + 0 = 1

Question 5: `"EUMCOPTR"` - Even indices (0,2,4,6) go to front, odd indices (1,3,5,7) go to back - "COMPUTER" processing: - i=0 (C): even –> front: "C" - i=1 (O): odd –> back: "CO" - i=2 (M): even –> front: "MCO" - i=3 (P): odd –> back: "MCOP" - i=4 (U): even –> front: "UMCOP" - i=5 (T): odd –> back: "UMCOPT" - i=6 (E): even –> front: "EUMCOPT" - i=7 (R): odd –> back: "EUMCOPTR"

Question 6: `[5, 4, 8, 3, 6, 12, 7]` - Processing backwards from index 4 to 0: - i=4: list.get(4) = 7 (odd), no change - i=3: list.get(3) = 12 (even), insert 6 at index 3 - Result: [5, 8, 3, 6, 12, 7] - i=2: list.get(2) = 3 (odd), no change - i=1: list.get(1) = 8 (even), insert 4 at index 1 - Result: [5, 4, 8, 3, 6, 12, 7] - i=0: list.get(0) = 5 (odd), no change - Final: [5, 4, 8, 3, 6, 12, 7]