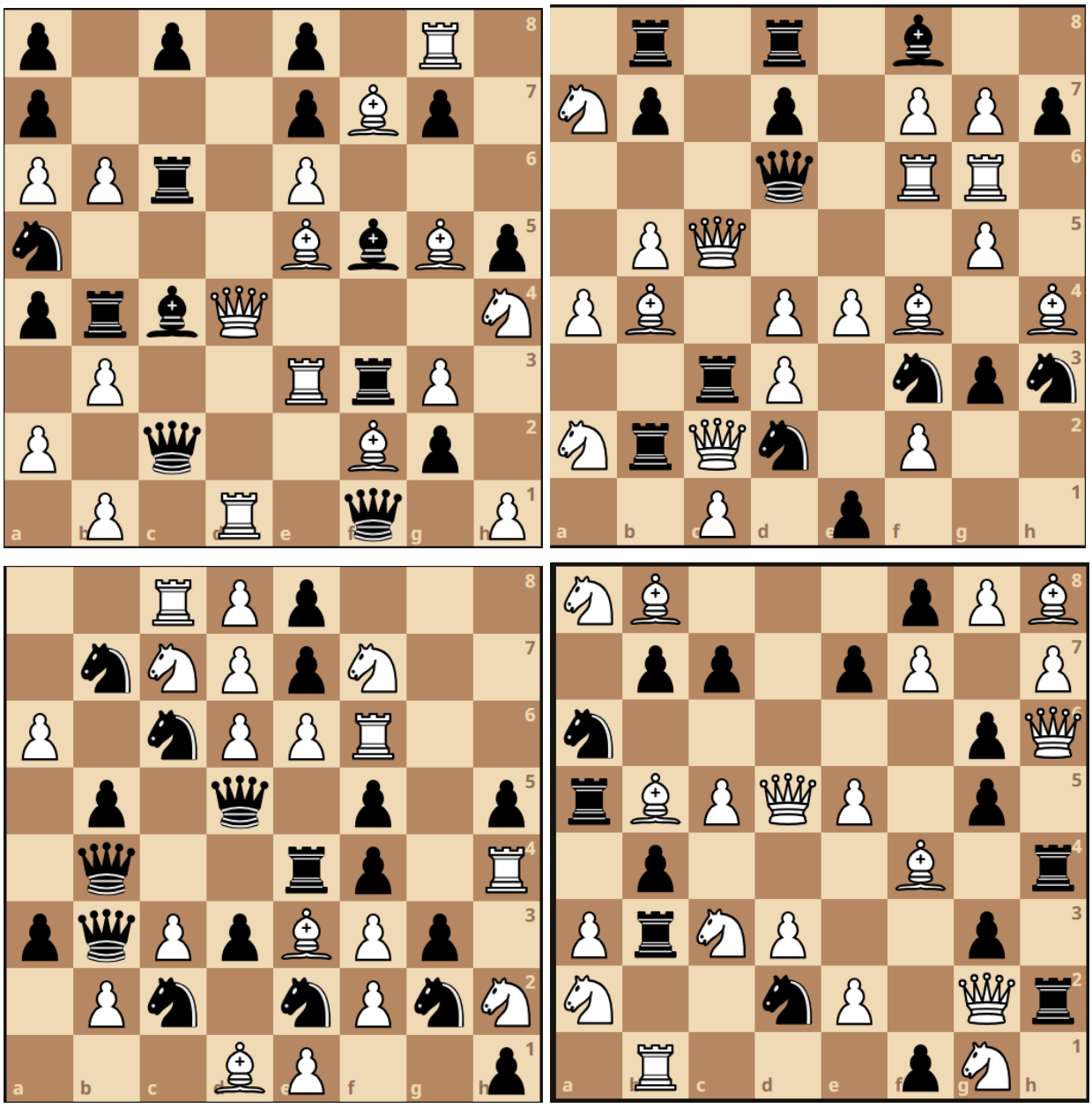
Cryptography CHESSY 2.0 Writeup

p1p1p1R1/p3pBp1/PPr1P3/n3BbBp/prbQ3N/1P2RrP1/P1q2Bp1/1P1R1q1P
1r1r1b2/Np1p1PPp/3q1RR1/1PQ3P1/PB1PPB1B/2rP1npn/NrQn1P2/2P1p3
2RPp3/1nNPpN2/P1nPPR2/1p1q1p1p/1q2rp1R/pqPpBPp1/1Pn1nPnN/3BP2p
NB3pPB/1pp1pP1P/n5pQ/rBPQP1p1/1p3B1r/PrNP2p1/N2nP1Qr/1R3pN1
N1B1rp2/P1npBP1r/R1Ppb1q1/Qp1p1n2/p1p2npB/4P1rb/1qnp1bn1/P1rqP1pB
BP2B3/2Br1pp1/1r1n2BB/RPPb1PPP/pRp1q1RP/Pn1b3n/PQpP1RB1/P2B1p1P
R1pRpQ1p/1Pb3r1/PP1PPNbP/2P1RPRQ/P1Bnp1Pp/1Rnr3B/B1PBPQp1/3BQP1B
bP1Pqb1p/R2P2P1/RBp1R1Pr/Np2PbPp/2Rp2R1/1p1rB1n1/P1pr2B1/3pR1bp
4B1pr/b1Pp2P1/3prb2/n3N3/3b1N1b/1p3P2/5pRp/6P1
P3p1P1/3p2p1/2N3NR/R4p2/1Bn2QP1/6Q1/4N2p/5p1p
p1PpP1Pp/1RnbP2P/pPP1RRr1/2npBPB1/N2p1bp1/nRpB1BBN/bpR1bB1r/PPpBP3
p5pP/Ppb1r1p1/b2Pp2n/3P4/2PPP2B/2qRp1p1/p3P3/1P6
P1bP2Np/1bn2pPP/1PPbP1r1/P1R2R2/RRb1q1NR/qbRN1rpP/r1Pr4/1p5b
pN2pbbQ/rp4qr/Q1B1r1pb/rpP1pr1p/1b3BP1/qPP1p1Bq/1PppNpP1/4p3
rbp1Np1p/1Pb2RNr/1Pp2pR1/PpP1pP1n/N1B1N2b/q1p1nn1r/3Np3/1PPRb2p
1r5p/6P1/3p1P2/Pp1NR3/1Pp1p3/Qb1Pn3/6PN/N1np1b2
1PB1nqP1/1PNp1pP1/rPP1PBpp/rRNB2Rp/pP1b1RNn/1RPpbPpp/rR2rn2/P1BP3B
3Bp3/P7/n7/7r/8/r6q/1pp1R3/2q1r1n1
n2pQnb1/B1NPn2R/Bp1Pp2n/QRP2P1p/2R2P1p/b3n1NP/q2p1q1r/P1P2nNP
P5p1/pP2PP1B/Pp6/1Qr5/p2b2b1/4r3/2PNp2p/2q1n3
2p3rB/qp2NrP1/1n2P3/5PR1/P1bbPpPr/rR1B1Q2/pB2PpPb/1NPPP1np
1nRRbpp1/Ppbp3P/1R1rpPPp/pP1PQ1n1/1rP1RQPB/P1p5/1Nr2Q1n/1b3bp1
PppppN1q/3p1Qp1/r2p1Pp1/1Ppq2RQ/nP1rN1NQ/P1P1Bb2/p3NPpp/pPP1PP1R
1BP5/2P1P1r1/7R/1N4Q1/P3R2P/P2R4/5R2/p1pp4
2n1B3/qprnPp2/1B2N3/6p1/3P1p1R/2B1PpB1/N4N2/8
Pp1qpp2/b1PPb1Pr/r1P1ppNp/rPP2R1p/1P2P2P/rP1Q1Rrp/pP1nPpBp/b1rpppP1
2Pp1B2/RpnPNPp1/PR1rnPR1/2q1Bppn/bp2rPR1/Nrp1n2p/pP1PNprn/1p1RRpbP
4p3/4P1pb/3pR2R/1p4Bp/1p3PBP/2PP1rr1/7P/pQ1p2p1
1B2Pppp/1p1P4/2pPBR2/r1rn1N1P/B1PpbpqQ/2P1n1p1/P3pqnb/p2RP2N
5p2/B3n3/n1P2Pb1/2n1pPrn/6P1/1P2P2R/1R3Rp1/P6B
R1p4p/r3p3/3pB2R/2N3P1/4q2B/2p3Pn/2pR3P/1PP5
3PnP1n/RbpR1rp1/NPp1R1NN/RP1r1Pn1/pr1pBp2/2pprb1P/NR2pPpr/3P2bR
P6Q/7q/6p1/1NB3P1/1p1q4/1P4N1/2p5/2r1n3
2pPP2p/2r1P3/p4p2/1pn4P/p2pp1R1/r3q1P1/P2PP1p1/P1ppP2p
5P2/PR1pB1qP/nPPB1P1Q/1Rp1P1pP/2rr2p1/2pr1b2/p1QR2RN/pPqPr1pp
p2bq1pp/pnP2N1N/1bb1pqb1/nPnPRP1P/P1P5/n2ppPrB/2p4P/pq2QB2
p1r5/5p1p/2p1B3/rbPNR3/R2p1R2/4p1Q1/1PR2P2/8
rp2n1P1/P2p3N/P4B1P/3n2p1/P4p2/1b3n2/Q1PP1ppn/2p4P
2R1q2n/R2Pb3/ppb2bpN/n3pb1r/pp1p1b1q/rN2NnB1/Q1p1rqpp/r1nb1bb1

Above is a given text file, “FEN.txt”. Looking at each line, we can see that every single line is a FEN of a singular chess board. Let’s use a chess editor: lichess.org/editor and paste in the first 4 FEN to get 4 chess boards.



What do you notice about the first 4 boards?

None of the boards contain any kings, and the placement of each piece, no matter white or black, has no pattern at all, and the pieces found on the board are much more than you find on a singular board.

So what is the approach to solving the question?
It would definitely not be related to having excessive chess skills (even with my quite high chess rating :D), though I think you may need some super basic chess knowledge. Let's think along the lines of why there are no kings on every single board? The difference between the king and the other pieces are as follows:

- The king is the last piece needed to survive to continue playing the game
- The only possible singular piece of one side (ie black or white)
- The only piece that does not have a piece value

Looking at the possibilities, the most probable outcome to try out for encryption would be using the piece value of the pieces on each board to calculate the ASCII character, which is part of the flag. However, there are black and white pieces on the board. Here is a link on piece values in chess: https://www.masterclass.com/articles/chess-piece-guide

Let's analyze the first board:

For white pieces, there are:
8 pawns → 8 x 1 = 8 piece value
4 bishops → 4 x 3 = 12 piece value
1 knight → 1 x 3 = 3 piece value
3 rooks → 3 x 5 = 15 piece value
1 queen → 1 x 9 = 9 piece value
TOTAL: 47 piece value for white

For black pieces, there are:
9 pawns → 9 x 1 = 9 piece value
2 bishops → 2 x 3 = 6 piece value
1 knight → 1 x 3 = 3 piece value
3 rooks → 3 x 5 = 15 piece value
2 queens → 2 x 9 = 18 piece value
TOTAL: 51 piece value for black

Let's now look at the ASCII Table: https://www.vlsifacts.com/ascii-code/
We can try the following:

1. White piece value + Black piece value (47 + 51 = 98 → b)
2. White piece value - Black piece value (47 - 51 = -4 → ???)
3. Black piece value - White piece value (51 - 47 = 4 → EOT)

Knowing that the flag format for BCACTF 4.0 is bcactf{...}, we know that the 1st option gets us the first letter 'b', we know that we are on the right track to getting the flag, since the first letter of the flag is 'b'. If we do this process again for the next 5 letters, we would get 'bcactf', which

confirms our theory. However, this method is too slow, so we can code it out to get the entire flag.

Here is my code:

```
def calculate_piece_value(position):
    piece_values = {'P': 1, 'N': 3, 'B': 3, 'R': 5, 'Q': 9}

    total_value = 0

    for row in position.split('/'):
        for char in row:
            if char.isalpha():
                total_value += piece_values[char.upper()]

    return total_value


def calculate_total_piece_value(positions):
    total_values = []

    for position in positions:
        piece_value = calculate_piece_value(position)
        total_values.append(piece_value)

    return total_values


# Example usage
positions =
['p1p1p1R1/p3pBp1/PPr1P3/n3BbBp/prbQ3N/1P2RrP1/P1q2Bp1/1P1R1q1P',
'1r1r1b2/Np1p1PPp/3q1RR1/1PQ3P1/PB1PPB1B/2rP1npn/NrQn1P2/2P1p3',
'2RPp3/1nNPpN2/P1nPPR2/1p1q1p1p/1q2rp1R/pqPpBPp1/1Pn1nPnN/3BP2p',
'NB3pPB/1pp1pP1P/n5pQ/rBPQP1p1/1p3B1r/PrNP2p1/N2nP1Qr/1R3pN1',
'N1B1rp2/P1npBP1r/R1Ppb1q1/Qp1p1n2/p1p2npB/4P1rb/1qnp1bn1/P1rqP1pB',
'BP2B3/2Br1pp1/1r1n2BB/RPPb1PPP/pRp1q1RP/Pn1b3n/PQpP1RB1/P2B1p1P',
'R1pRpQ1p/1Pb3r1/PP1PPNbP/2P1RPRQ/P1Bnp1Pp/1Rnr3B/B1PBPQp1/3BQP1B',
'bP1Pqb1p/R2P2P1/RBp1R1Pr/Np2PbPp/2Rp2R1/1p1rB1n1/P1pr2B1/3pR1bp',
'4B1pr/b1Pp2P1/3prb2/n3N3/3b1N1b/1p3P2/5pRp/6P1',
```

```
'P3p1P1/3p2p1/2N3NR/R4p2/1Bn2QP1/6Q1/4N2p/5p1p',
'p1PpP1Pp/1RnbP2P/pPP1RRr1/2npBPB1/N2p1bp1/nRpB1BBN/bpR1bB1r/PPpBP3',
'p5pP/Ppb1r1p1/b2Pp2n/3P4/2PPP2B/2qRp1p1/p3P3/1P6',
'P1bP2Np/1bn2pPP/1PPbP1r1/P1R2R2/RRb1q1NR/qbRN1rpP/r1Pr4/1p5b',
'pN2pbbQ/rp4qr/Q1B1r1pb/rpP1pr1p/1b3BP1/qPP1p1Bq/1PppNpP1/4p3',
'rbp1Np1p/1Pb2RNr/1Pp2pR1/PpP1pP1n/N1B1N2b/q1p1nn1r/3Np3/1PPRb2p',
'1r5p/6P1/3p1P2/Pp1NR3/1Pp1p3/Qb1Pn3/6PN/N1np1b2',
'1PB1nqP1/1PNp1pP1/rPP1PBpp/rRNB2Rp/pP1b1RNn/1RPpbPpp/rR2rn2/P1BP3B',
'3Bp3/P7/n7/7r/8/r6q/1pp1R3/2q1r1n1',
'n2pQnb1/B1NPn2R/Bp1Pp2n/QRP2P1p/2R2P1p/b3n1NP/q2p1q1r/P1P2nNP',
'P5p1/pP2PP1B/Pp6/1Qr5/p2b2b1/4r3/2PNp2p/2q1n3',
'2p3rB/qp2NrP1/1n2P3/5PR1/P1bbPpPr/rR1B1Q2/pB2PpPb/1NPPP1np',
'1nRRbpp1/Ppbp3P/1R1rpPPp/pP1PQ1n1/1rP1RQPB/P1p5/1Nr2Q1n/1b3bp1',
'PppppN1q/3p1Qp1/r2p1Pp1/1Ppq2RQ/nP1rN1NQ/P1P1Bb2/p3NPpp/pPP1PP1R',
'1BP5/2P1P1r1/7R/1N4Q1/P3R2P/P2R4/5R2/p1pp4',
'2n1B3/qprnPp2/1B2N3/6p1/3P1p1R/2B1PpB1/N4N2/8',
'Pp1qpp2/b1PPb1Pr/r1P1ppNp/rPP2R1p/1P2P2P/rP1Q1Rrp/pP1nPpBp/b1rpppP1',
'2Pp1B2/RpnPNPp1/PR1rnPR1/2q1Bppn/bp2rPR1/Nrp1n2p/pP1PNprn/1p1RRpbP',
'4p3/4P1pb/3pR2R/1p4Bp/1p3PBP/2PP1rr1/7P/pQ1p2p1',
'1B2Pppp/1p1P4/2pPBR2/r1rn1N1P/B1PpbpqQ/2P1n1p1/P3pqnb/p2RP2N',
'5p2/B3n3/n1P2Pb1/2n1pPrn/6P1/1P2P2R/1R3Rp1/P6B',
'R1p4p/r3p3/3pB2R/2N3P1/4q2B/2p3Pn/2pR3P/1PP5',
'3PnP1n/RbpR1rp1/NPp1R1NN/RP1r1Pn1/pr1pBp2/2pprb1P/NR2pPpr/3P2bR',
'P6Q/7q/6p1/1NB3P1/1p1q4/1P4N1/2p5/2r1n3',
'2pPP2p/2r1P3/p4p2/1pn4P/p2pp1R1/r3q1P1/P2PP1p1/P1ppP2p',
'5P2/PR1pB1qP/nPPB1P1Q/1Rp1P1pP/2rr2p1/2pr1b2/p1QR2RN/pPqPr1pp',
'p2bq1pp/pnP2N1N/1bb1pqb1/nPnPRP1P/P1P5/n2ppPrB/2p4P/pq2QB2',
'p1r5/5p1p/2p1B3/rbPNR3/R2p1R2/4p1Q1/1PR2P2/8',
'rp2n1P1/P2p3N/P4B1P/3n2p1/P4p2/1b3n2/Q1PP1ppn/2p4P',
'2R1q2n/R2Pb3/ppb2bpN/n3pb1r/pp1p1b1q/rN2NnB1/Q1p1rqpp/r1nb1bb1'
]


total_values = calculate_total_piece_value(positions)

result = ""
for i, position in enumerate(positions):
    print("Piece value for position", i + 1, ":", total_values[i])
    result += chr(total_values[i])
print(result)
```

What does the code do?
calculate_piece_value takes in the positions of pieces of a board and neatens the FEN to something easier to process (removing the / and the irrelevant numbers) and assigns the piece values of each type of piece no matter white or black:

Pawn (P) is piece value of 1
Knight (N) is piece value of 3
Bishop (B) is piece value of 3
Rook (R) is piece value of 5
Queen (Q) is piece value of 9

*Do note that the uppercase is white pieces and lowercase is black pieces in the FEN, but we will just convert all the lowercase to uppercase because we do not need to know what color the piece is.

The function will then return the total piece value, no matter white or black as an integer output.

calculate_total_piece_value calculates every single board's *calculate_piece_value* and append it into a list and returns the list.

Positions is just the FENs of every board in a list.

The remaining code just prints out the total piece values of each board. It also converts all the piece values into ASCII, so we can see the flag printed out. Here is the output of the code:

Piece value for position 1 : 98
Piece value for position 2 : 99
Piece value for position 3 : 97
Piece value for position 4 : 99
Piece value for position 5 : 116
Piece value for position 6 : 102
Piece value for position 7 : 123
Piece value for position 8 : 98
Piece value for position 9 : 49
Piece value for position 10 : 53
Piece value for position 11 : 104
Piece value for position 12 : 48
Piece value for position 13 : 112
Piece value for position 14 : 115
Piece value for position 15 : 95
Piece value for position 16 : 52
Piece value for position 17 : 114

Piece value for position 18 : 51
Piece value for position 19 : 110
Piece value for position 20 : 55
Piece value for position 21 : 95
Piece value for position 22 : 107
Piece value for position 23 : 110
Piece value for position 24 : 49
Piece value for position 25 : 54
Piece value for position 26 : 104
Piece value for position 27 : 116
Piece value for position 28 : 53
Piece value for position 29 : 95
Piece value for position 30 : 51
Piece value for position 31 : 52
Piece value for position 32 : 106
Piece value for position 33 : 50
Piece value for position 34 : 49
Piece value for position 35 : 110
Piece value for position 36 : 100
Piece value for position 37 : 57
Piece value for position 38 : 50
Piece value for position 39 : 125
bcactf{b15h0ps_4r3n7_kn16ht5_34j21nd92}

Thus, the flag is **bcactf{b15h0ps_4r3n7_kn16ht5_34j21nd92}**