def isEmpty(board):
    empty = False
    for row in board:
        for val in row:
            if val == '1':
                empty = True
    return empty

def main(input):
    inputsplit = list(input)
    bin1 = bin(int(inputsplit[0], 16))[2:].zfill(4)
    bin2 = bin(int(inputsplit[1], 16))[2:].zfill(4)
    board = [list(bin1), list(bin2)]
    alist = ['A', 'A', '~A', '~A']
    blist = ['B', '~B']
    clist = ['~C', 'C', 'C', '~C']
    output = ''
    while(isEmpty(board) == True):
        if board[0][0] == '1' and board[0][1] == '1' and board[0][2] == '1' and board[0][3] == '1':
            board[0][0] = 0
            board[0][1] = 0
            board[0][2] = 0
            board[0][3] = 0
            output += 'B+
        elif board[1][0] == '1' and board[1][1] == '1' and board[1][2] == '1' and board[1][3] == '1':
            board[1][0] = 0
            board[1][1] = 0
            board[1][2] = 0
            board[1][3] = 0
            output += '~B+
        elif board[0][0] == '1' and board[0][1] == '1' and board[1][0] == '1' and board[1][1] == '1':
            board[0][0] = 0
            board[1][1] = 0
            board[1][0] = 0
            board[0][1] = 0
            output += 'A+
        elif board[0][1] == '1' and board[0][2] == '1' and board[1][1] == '1' and board[1][2] == '1':
            board[0][1] = 0
            board[0][2] = 0
            board[1][1] = 0
            board[1][2] = 0
            output += 'C+
        elif board[0][2] == '1' and board[0][3] == '1' and board[1][2] == '1' and board[1][3] == '1':
            board[0][2] = 0
            board[1][2] = 0
            board[0][3] = 0
            board[1][3] = 0
            output += '~A+
        elif board[0][0] == '1' and board[1][0] == '1' and board[0][3] == '1' and board[1][3] == '1':
            board[0][0] = 0
            board[1][0] = 0
            board[0][3] = 0
            board[1][3] = 0
            output += '~C+
        else:
            x = []
            y = 0
            found = False
            for r in range (0, 2):
prev = False
for c in range(0, 4):
    if not found:
        if board[r][c] == '1':
            if prev == True:
                x = [c - 1, c]
                y = r
                board[y][c - 1] = 0
                board[y][c] = 0
                found = True
            else:
                prev = True
        else:
            prev = True
    else:
        prev = False
if found:
    out = blist[y]
    if x[0] == 0:
        out = "A" + out
    elif x[0] == 2:
        out = "~A" + out
    else:
        out = out + "C"
    output += out + "+
if found == False:
    found = False
    x = 0
    for c in range(0, 4):
        prev = False
        for r in range(0, 2):
            if not found:
                if board[r][c] == '1':
                    if prev == True:
                        x = c
                        found = True
                        board[0][c] = 0
                        board[1][c] = 0
                    else:
                        prev = True
            else:
                prev = False
            if found:
                output += alist[x] + '" + clist[x] + "+"
        if found == False:
            if board[0][0] == '1' and board[0][3] == '1':
                board[0][0] = 0
                board[0][3] = 0
                output += 'B~C+
            elif board[1][0] == '1' and board[1][3] == '1':
                board[1][0] = 0
                board[1][3] = 0
                output += '~B~C+
            if found == False:
                for r in range(0, 2):
                    for c in range(0, 4):
                        if board[r][c] == '1':
                            output +=alist[c] + blist[r] + clist[c] + "+"
                            board[r][c] = 0
print(output[:-1])

with open('jr-test-input.txt') as fp:
    for i in range(0, 5):
        Input = fp.readline()
        main(Input)