**Exp14: Write the Python Program to implement Alpha & Beta pruning algorithm for gaming.**

**Input:**

import math

def print\_board(b):

    for r in b: print(" | ".join(r)); print("-"\*5)

def win(b):

    lines = b + [list(c) for c in zip(\*b)] + [[b[i][i] for i in range(3)], [b[i][2-i] for i in range(3)]]

    return next((l[0] for l in lines if l[0] != " " and all(c == l[0] for c in l)), None)

def full(b): return all(c != " " for r in b for c in r)

def ab(b, a, be, maxp):

    w = win(b)

    if w == "O": return 1

    if w == "X": return -1

    if full(b): return 0

    best = -math.inf if maxp else math.inf

    for i in range(3):

        for j in range(3):

            if b[i][j] == " ":

                b[i][j] = "O" if maxp else "X"

                val = ab(b, a, be, not maxp)

                b[i][j] = " "

                if maxp:

                    best, a = max(best, val), max(a, val)

                    if be <= a: return best

                else:

                    best, be = min(best, val), min(be, val)

                    if be <= a: return best

    return best

def best\_move(b):

    m, best = None, -math.inf

    for i in range(3):

        for j in range(3):

            if b[i][j] == " ":

                b[i][j] = "O"

                val = ab(b, -math.inf, math.inf, False)

                b[i][j] = " "

                if val > best: best, m = val, (i, j)

    return m

def get\_move(b):

    while True:

        try:

            r = int(input("Row (0-2): "))

            c = int(input("Col (0-2): "))

            if 0 <= r < 3 and 0 <= c < 3 and b[r][c] == " ":

                return r, c

        except: pass

        print("Invalid!")

def game():

    b = [[" "]\*3 for \_ in range(3)]

    print("You: X | AI: O (Alpha-Beta)")

    while True:

        print\_board(b)

        if win(b) or full(b): break

        r, c = get\_move(b); b[r][c] = "X"

        if win(b) or full(b): break

        m = best\_move(b)

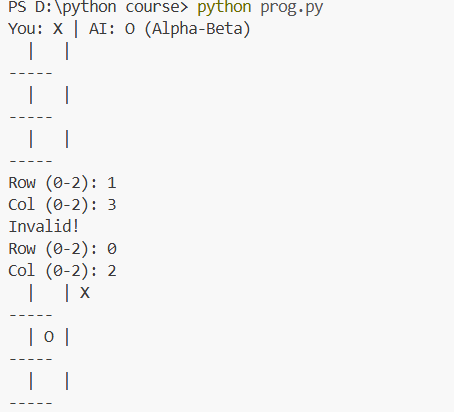
        if m: b[m[0]][m[1]] = "O"

    print\_board(b)

    print("Draw!" if not win(b) else f"{win(b)} wins!")

game()

**Output:**

****