EXPERIMENT:14 Write the python program to implement Apha & Beta pruning algorithm for gaming

PROGRAM:

```
import math


def print_board(board):
    for row in board:
        print(" | ".join(row))
        print("-"*5)


def check_winner(board):
    for row in board:
        if row.count(row[0]) == 3 and row[0] != " ": return row[0]
    for col in range(3):
        if board[0][col] == board[1][col] == board[2][col] != " ": return
board[0][col]
    if board[0][0] == board[1][1] == board[2][2] != " " or board[0][2] ==
board[1][1] == board[2][0] != " ":
        return board[1][1]
    return None


def alphabeta(board, depth, alpha, beta, is_max):
    winner = check_winner(board)
    if winner == "O": return 1
    if winner == "X": return -1
    if all(cell != " " for row in board for cell in row): return 0
```

```
    if is_max:
        value = -math.inf
        for i in range(3):
            for j in range(3):
                if board[i][j] == " ":
                    board[i][j] = "O"
                    value = max(value, alphabeta(board, depth+1, alpha, beta,
False))
                    board[i][j] = " "
                    alpha = max(alpha, value)
                    if beta <= alpha: break
        return value
    else:
        value = math.inf
        for i in range(3):
            for j in range(3):
                if board[i][j] == " ":
                    board[i][j] = "X"
                    value = min(value, alphabeta(board, depth+1, alpha, beta,
True))
                    board[i][j] = " "
                    beta = min(beta, value)
                    if beta <= alpha: break
        return value


def best_move(board):
```

```
    best_val = -math.inf
    move = (-1, -1)
    for i in range(3):
        for j in range(3):
            if board[i][j] == " ":
                board[i][j] = "O"
                move_val = alphabeta(board, 0, -math.inf, math.inf, False)
                board[i][j] = " "
                if move_val > best_val:
                    move = (i, j)
                    best_val = move_val
    return move

board = [[" "]*3 for _ in range(3)]
while True:
    print_board(board)
    r, c = map(int, input("Enter row and col (0-2): ").split())
    if board[r][c] != " ":
        print("Invalid move!")
        continue
    board[r][c] = "X"
    if check_winner(board) == "X":
        print_board(board); print("You win!"); break
    if all(cell != " " for row in board for cell in row):
        print_board(board); print("Draw!"); break
    ai_r, ai_c = best_move(board)
```

```
    board[ai_r][ai_c] = "O"
    if check_winner(board) == "O":
        print_board(board); print("AI wins!"); break
```

OUTPUT: