Experiment 14: Alpha-Beta Pruning Algorithm

Aim:

Implement an Algorithm in Python for solving Alpha-Beta Pruning

Python Program:

```
MAX, MIN = 1000, -1000

def alph_beta_pru(depth, nodeIndex, maximizingPlayer,
                  values, alpha, beta):

    if depth == 3:
    return values[nodeIndex]

    if maximizingPlayer:
    best = MIN
    for i in range(0, 2):
        val = alph_beta_pru(depth + 1, nodeIndex * 2 + i, False, values, alpha, beta)
        best = max(best, val)
        alpha = max(alpha, best)
        if beta <= alpha:
            break
    return best
    else:
    best = MAX
    for i in range(0, 2):
        val = alph_beta_pru(depth + 1, nodeIndex * 2 + i, True, values, alpha, beta)
        best = min(best, val)
        beta = min(beta, best)
        if beta <= alpha:
            break
    return best

if __name__ == "__main__":

    values = [3, 5, 6, 9, 1, 2, 0, -1]
    print("The optimal value is :", alph_beta_pru(0, 0, True, values, MIN, MAX))
```

Output:

The optimal value is : 5

Result:

Code has been Implemented successfully.