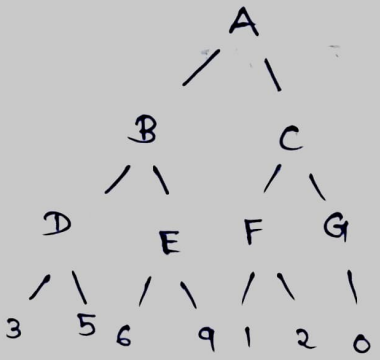
Exp No: 05

Date: 07/09/24 MINIMAX ALGORITHM

AIM:

To implement the minimaximum algorithm in Python.

ALGORITHM:



(i) The function recursively evaluate a tree.

(ii) It takes node depth of tree and a boolean if player is maximum.

(iii) If its a terminal node return node value.

(iv) The function gets child nodes asking the gets child node function.

(v) Computes best score for maximizing A.

PROGRAM:

```
def minimax(node, depth, is_maximizing):
    if depth == 0:
        return node
    if is_maximizing:
        best_value = -math.inf
        for child in get_children(node):
            value = minimax(child, depth-1, false)
        best_value = max(best_value, value)
        return best_value
    else:
        best_value = math.inf
        for child in get_children(node):
            value = minimax(child, depth-1, true)
            best_value = min(best_value, value)
        return best_value

def get_children(node):
    return node.get("children", [])

game_tree = {
    "value": "A",
    "children": [{
        "value": "B"
        children": [{"value": "D", "children": [],
                    "value": "E", "children":
        "terminal_value": {}}]}
                    "terminal_value": [],
```

{ "value": "C", "children": [

{"value": "F", "children", [ ], "terminal_value":

{"value": "G", "children [ ], "terminal_value":

] }] }

if __name__ == "__main":

best_score = minimax (game_tree, 1, True)

print (f "Best score for maximizer (A):

{ best_score } ")

RESULT:

Thus, the minimax algorithm is executed successfully and output is verified.