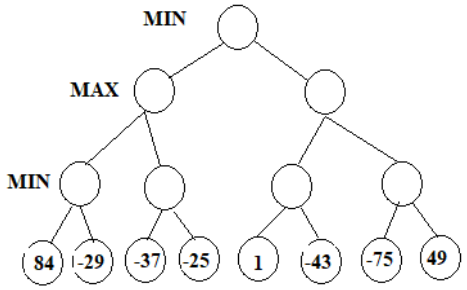
1. Write a python program to solve the given min-max tree and find the optimal path for MIN to win the game.



Program :

```
import math


class Node:
    def __init__(self, value=None):
        self.value = value
        self.children = []


def min_max(node, depth, is_maximizing):
    if depth == 0 or not node.children:
        return node.value


    if is_maximizing:
        max_eval = -math.inf
        for child in node.children:
            eval = min_max(child, depth - 1, False)
            max_eval = max(max_eval, eval)
```

```
        return max_eval
    else:
        min_eval = math.inf
        for child in node.children:
            eval = min_max(child, depth - 1, True)
            min_eval = min(min_eval, eval)
        return min_eval

def create_tree():
    # Example min-max tree
    root = Node(3)
    root.children = [Node(5), Node(6), Node(8)]

    root.children[0].children = [Node(1), Node(2), Node(0)]
    root.children[1].children = [Node(9), Node(7), Node(4)]
    root.children[2].children = [Node(5), Node(3), Node(2)]

    return root

def find_optimal_path(root):
    optimal_path = []
    for i, child in enumerate(root.children):
        eval = min_max(child, math.inf, False)
        if eval == root.value:
```

```
            optimal_path.append(i)
    return optimal_path


def main():
    tree = create_tree()
    optimal_path = find_optimal_path(tree)


    print("Optimal Path for MIN to Win:")
    print(optimal_path)


if __name__ == "__main__":
    main()
```

Output :

```
Optimal Path for MIN to Win:
[2]
```