Q2. Develop a MiniMax algorithm based game agent for the following game description. Provide options for a human player to play with your agent.

Consider a two player game. Assume there are some piles of stones. At each turn, a player can remove any number of stones from any single pile. A player loses, if there are no stones left on his turn. An evaluation score of 1 if Player 1 wins, -1 if Player 2 wins and 0 for a tie (if possible).

Assume that both players play optimally. Assume there are two piles and accept the number of stones in each pile at the beginning of the game. Show the moves taken by the winning player in the game tree.

Ans:- Here to start the game we input count of stones in each piles at beginning of game and also ask who starts the game , agent of user. The final output depends on 2 factors – who start the games and initial configuration. Here we used min-max algorithm to decide the moves given both players play optimally and finally prints winning player.

Sample input and output:-

Example 1:

```
Enter stones in pileA
0
Enter stones in pileB
1
Enter 1 for user play first ,2 for agent to play first
2
Current pile - 0 1
Agent moves 1 Stones from pile 2

Agent is Winner

Exit code: 0 (normal program termination)
```

Example 2:

```
Enter stones in pileA
1
Enter stones in pileB
0
Enter 1 for user play first ,2 for agent to play first
1
Current pile - 1 0
User moves 1 Stones from pile 1

User is Winner

Exit code: 0 (normal program termination)
```

Example 3:

```
Enter stones in pileA
1
Enter stones in pileB
1
Enter 1 for user play first ,2 for agent to play first
1
Current pile - 1 0
Agent moves 1 Stones from pile 1

Agent is Winner

Exit code: 0 (normal program termination)
```

Example 4:

```
Enter stones in pileA
2
Enter stones in pileB
1
Enter 1 for user play first ,2 for agent to play first
2
Current pile - 2 1
Agent moves 1 Stones from pile 1
Current pile - 1 0
Agent moves 1 Stones from pile 1

Agent is Winner

Exit code: 0 (normal program termination)
```

Example 5:

```
Enter stones in pileA
1
Enter stones in pileB
2
Enter 1 for user play first ,2 for agent to play first
1
Current pile - 1 2
User moves 1 Stones from pile 2
Current pile - 1 0
User moves 1 Stones from pile 1

User is Winner

Exit code: 0 (normal program termination)
```

Example 6:

```
Enter stones in pileA
15
Enter stones in pileB
15
Enter 1 for user play first ,2 for agent to play first
2
Current pile - 15 13
User moves 2 Stones from pile 1
Current pile - 13 8
User moves 5 Stones from pile 1
Current pile - 8 0
User moves 8 Stones from pile 1

User is Winner

Exit code: 0 (normal program termination)
```

Example 7:

```
Enter stones in pileA
15
Enter stones in pileB
15
Enter 1 for user play first ,2 for agent to play first
1
Current pile - 15 13
Agent moves 2 Stones from pile 1
Current pile - 13 8
Agent moves 5 Stones from pile 1
Current pile - 8 0
Agent moves 8 Stones from pile 1

Agent is Winner

Exit code: 0 (normal program termination)
```

Example 8:

```
Enter stones in pileA
20
Enter stones in pileB
15
Enter 1 for user play first ,2 for agent to play first
2
Current pile - 20 15
Agent moves 5 Stones from pile 1
Current pile - 15 13
Agent moves 2 Stones from pile 1
Current pile - 13 8
Agent moves 5 Stones from pile 1
Current pile - 8 0
Agent moves 8 Stones from pile 1

Agent is Winner

Exit code: 0 (normal program termination)
```

Example 9:

```
Enter stones in pileA
20
Enter stones in pileB
15
Enter 1 for user play first ,2 for agent to play first
2
Current pile - 20 15
Agent moves 5 Stones from pile 1
Current pile - 15 13
Agent moves 2 Stones from pile 1
Current pile - 13 8
Agent moves 5 Stones from pile 1
Current pile - 8 0
Agent moves 8 Stones from pile 1

Agent is Winner

Exit code: 0 (normal program termination)
```

Example 10:

```
Enter stones in pileA
20
Enter stones in pileB
20
Enter 1 for user play first ,2 for agent to play first
1
Current pile - 20 13
Agent moves 7 Stones from pile 1
Current pile - 13 8
Agent moves 5 Stones from pile 1
Current pile - 8 0
Agent moves 8 Stones from pile 1

Agent is Winner

Exit code: 0 (normal program termination)
```