Name : Laukik Nitin Marathe

Roll No : TEAD21153

Subject : Software Lab-1

Class : TE(A)

Branch : AI&DS

Assignment No : 10

TITLE:

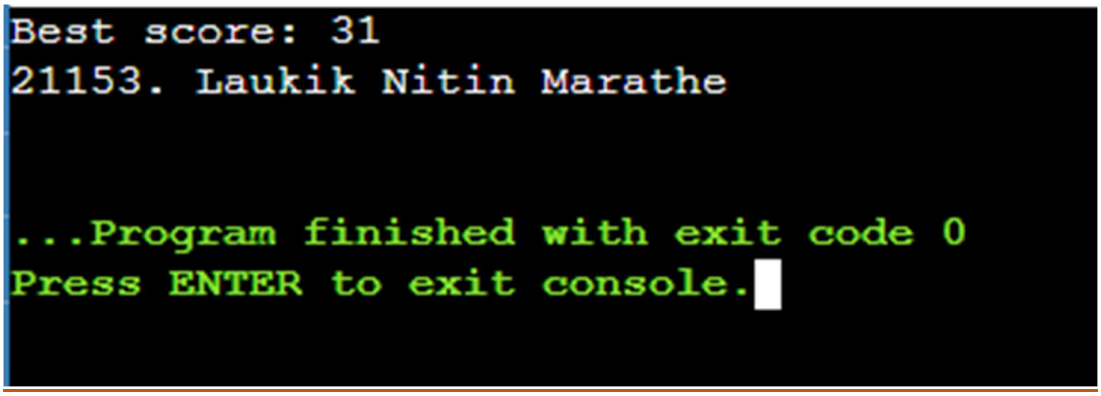
Implement Alpha-Beta Tree search for any game search problem.

CODE:

```
import math
# Example game state
game_state = [3, 5, 2, 9, 12, 5, 23, 23]
# Alpha-Beta Pruning function
def alpha_beta_pruning(state, depth, alpha, beta, is_maximizing_player):
    if depth == 0 or len(state) == 0:
        # Calculate the evaluation function (heuristic) for the current state
        return evaluate_state(state)
    if is_maximizing_player:
        max_eval = -math.inf
        for move in state:
            new_state = state.copy()
            new_state.remove(move)
            eval = alpha_beta_pruning(new_state, depth - 1, alpha, beta, False)
            max_eval = max(max_eval, eval)
            alpha = max(alpha, eval)
            if beta <= alpha:
                break  # Beta cutoff
        return max_eval
    else:
```

```
        min_eval = math.inf
        for move in state:
            new_state = state.copy()
            new_state.remove(move)
            eval = alpha_beta_pruning(new_state, depth - 1, alpha, beta, True)
            min_eval = min(min_eval, eval)
            beta = min(beta, eval)
            if beta <= alpha:
                break  # Alpha cutoff
        return min_eval
# Example evaluation function (heuristic)
def evaluate_state(state):
    return sum(state)
# Main function to start the game search
def main():
    depth = 4  # Depth of the search tree
    best_score = alpha_beta_pruning(game_state, depth, -math.inf, math.inf, True)
    print("Best score:", best_score)
if __name__ == "__main__":
    main()
```

OUTPUT:



Name : Laukik Nitin Marathe

Roll No : TEAD21153

Subject : Software Lab-1

Class : TE(A)

Branch : AI&DS

Assignment No : 14

TITLE:

Mini Project: Implement any one of the following Expert System

- Information management
- Hospitals and medical facilities
- Help desks management
- Employee performance evaluation
- Stock market trading
- Airline scheduling and cargo schedules

CODE:

```
import os

import platform


global studentlist

studentlist = ["jason yap", "Jake ramos", "James Pascual", "Jester Paglinga"]


def studentmanagement():


        print("\n++++++ Welcome to Evanz College Student Management System ++++++\n")

        print("[Choice 1: Showing the List of Student]")

        print("[Choice 2: Add New Student]")

        print("]Choice 3: Searching Student]")

        print("[Choice 4: Deleting a Student]\n")


        try:

                x = int(input("Enter a choice: "))
```

```
        except ValueError:
                exit("\nHy! This is not a Number")
        else:
                print("\n")

        if(x==1):
                print("Student List\n")
                for students in studentlist:
                        print("++ {} ++".format(students))

        elif(x==2):
                studentnew = input("Enter New Student: ")
                if(studentnew in studentlist):
                        print("\nThis Student {} Already In The Table".format(studentnew))
                else:
                        studentlist.append(studentnew)
                        print("\n++ New Student {} Added Successfully ++\n".format(studentnew))
                        for students in studentlist:
                                print("++ {} ++".format(students))

        elif(x==3):
                studentsearching = input("Choose Student Name To Search: ")
                if(studentsearching in studentlist):
                        print("\n++ There is a Record Found of this Student {}
++".format(studentsearching))
                else:
                        print("\n++ There is No Record Found Of this Student {}
++".format(studentsearching))

        elif(x==4):
                studentdelete = input("Choose a Student Name To Delete: ")
                if(studentdelete in studentlist):
```

```
                studentlist.remove(studentdelete)
                for students in studentlist:
                    print("++ {} ++".format(students))
            else:
                print("\n++ There is No Record Found of This Student {}
++".format(studentdelete))

    elif(x < 1 or x > 4):
        print("Please Enter Valid Choice")

studentmanagement()

def continueAgain():
    runningagain = input("\nWant to continue the process yes/no?: ")
    if(runningagain.lower() == 'yes'):
        if(platform.system() == "Windows"):
            print(os.system('cls'))
        else:
            print(os.system('clear'))
        studentmanagement()
        continueAgain()
    else:
        quit()

continueAgain()
```

OUTPUT:

```
++++++ Welcome to DYP College Student Management System ++++++

[Choice 1: Showing the List of Student]
[Choice 2: Add New Student]
]Choice 3: Searching Student]
[Choice 4: Deleting a Student]

Enter a choice: 1


Student List

++ jason yap ++
++ Jake ramos ++
++ James Pascual ++
++ Jester Paglinga ++

Want to continue the process yes/no?: no


...Program finished with exit code 0
Press ENTER to exit console.
```

```
++++++ Welcome to DYP College Student Management System ++++++

[Choice 1: Showing the List of Student]
[Choice 2: Add New Student]
]Choice 3: Searching Student]
[Choice 4: Deleting a Student]

Enter a choice: 2


Enter New Student: Laukik

++ New Student Laukik Added Successfully ++

++ jason yap ++
++ Jake ramos ++
++ James Pascual ++
++ Jester Paglinga ++
++ Laukik ++

Want to continue the process yes/no?: no


...Program finished with exit code 0
Press ENTER to exit console.
```