Name: Rameshwar Gosavi
Roll No.: 331
Batch: B

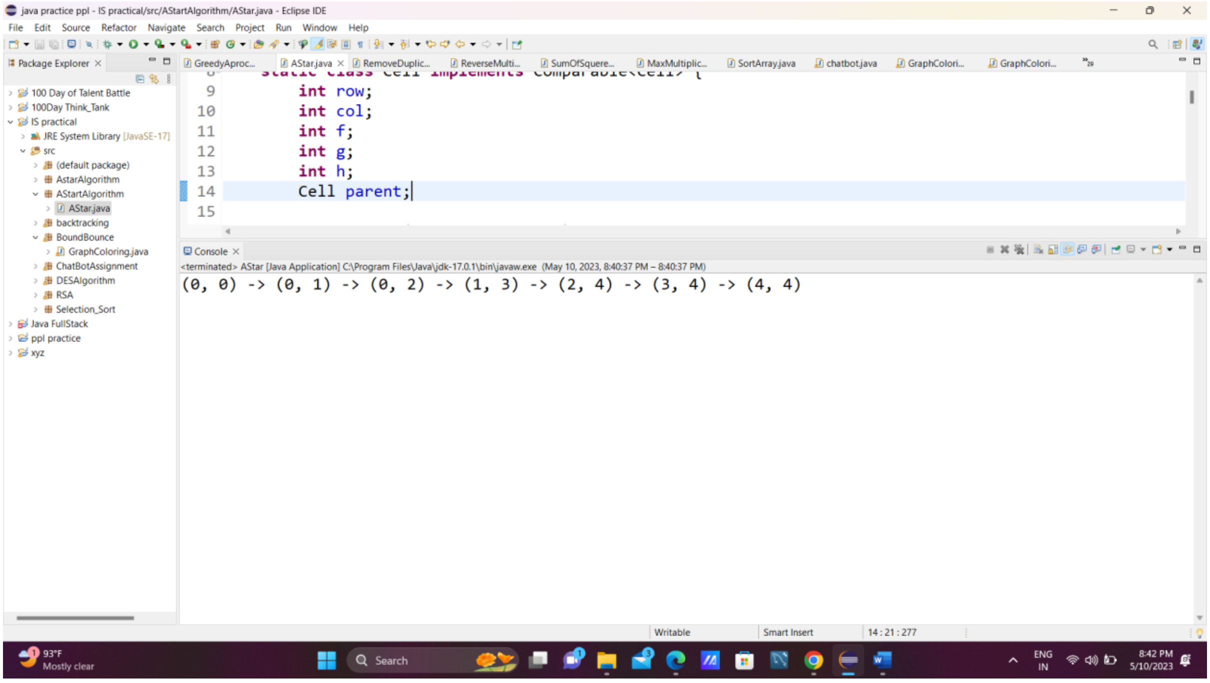
Title: Implement A star Algorithm for any game search problem

```
package AStartAlgorithm;

import java.util.ArrayList;
import java.util.PriorityQueue;

public class AStar {

    static class Cell implements Comparable<Cell> {
        int row;
        int col;
        int f;
        int g;
        int h;
        Cell parent;

        public Cell(int row, int col) {
            this.row = row;
            this.col = col;
        }

        @Override
        public int compareTo(Cell other) {
            return Integer.compare(this.f,
other.f);
        }
    }

    public static ArrayList<Cell> findPath(int[][]
grid, Cell start, Cell end) {
        PriorityQueue<Cell> openList = new
PriorityQueue<>();
```

```
        ArrayList<Cell> closedList = new
ArrayList<>();

        start.g = 0;
        start.h = getDistance(start, end);
        start.f = start.g + start.h;

        openList.offer(start);

        while (!openList.isEmpty()) {
            Cell current = openList.poll();

            if (current.row == end.row &&
current.col == end.col) {
                ArrayList<Cell> path = new
ArrayList<>();
                Cell node = current;

                while (node != null) {
                    path.add(node);
                    node = node.parent;
                }

                return path;
            }

            closedList.add(current);

            for (Cell neighbor : getNeighbors(grid,
current)) {
                if (closedList.contains(neighbor))
{
                    continue;
                }

                int tentativeGScore = current.g +
getDistance(current, neighbor);
```

```
                if (!openList.contains(neighbor)) {
                    openList.offer(neighbor);
                } else if (tentativeGScore >=
neighbor.g) {
                    continue;
                }

                neighbor.parent = current;
                neighbor.g = tentativeGScore;
                neighbor.h = getDistance(neighbor,
end);
                neighbor.f = neighbor.g +
neighbor.h;
            }
        }

        return null;
    }

    public static int getDistance(Cell a, Cell b) {
        int dx = Math.abs(a.col - b.col);
        int dy = Math.abs(a.row - b.row);

        if (dx > dy) {
            return 14 * dy + 10 * (dx - dy);
        } else {
            return 14 * dx + 10 * (dy - dx);
        }
    }

    public static ArrayList<Cell>
getNeighbors(int[][] grid, Cell cell) {
        ArrayList<Cell> neighbors = new
ArrayList<>();

        int[] rows = {-1, 0, 1, -1, 1, -1, 0, 1};
        int[] cols = {-1, -1, -1, 0, 0, 1, 1, 1};

        for (int i = 0; i < rows.length; i++) {
            int row = cell.row + rows[i];
            int col = cell.col + cols[i];

            if (row < 0 || row >= grid.length ||
col < 0 || col >= grid[0].length) {
                continue;
            }

            if (grid[row][col] == 1) {
                continue;
            }

            neighbors.add(new Cell(row, col));
        }

        return neighbors;
    }

    public static void main(String[] args) {
        int[][] grid = {
            {0, 0, 0, 0, 0},
            {0, 1, 1, 0, 0},
            {0, 0, 0, 0, 0},
            {0, 0, 1, 1, 0},
            {0, 0, 0, 0, 0}
        };
        Cell start = new Cell(0, 0);
        Cell end = new Cell(4, 4);

        ArrayList<Cell> path = findPath(grid,
start, end);

        if (path != null) {
            for (int i = path.size() - 1; i >= 0;
i--) {
                Cell cell = path.get(i);
                System.out.print("(" + cell.row +
", " + cell.col + ")");

                if (i != 0) {
                    System.out.print(" -> ");
                }
            }
        } else {
            System.out.println("No path found.");
        }
    }
}
```

Output :



```
    static class Cell implements Comparable<Cell> {
        int row;
        int col;
        int f;
        int g;
        int h;
        Cell parent;

```

```
<terminated> AStar [Java Application] C:\Program Files\Java\jdk-17.0.1\bin\javaw.exe  (May 10, 2023, 8:40:37 PM – 8:40:37 PM)
(0, 0) -> (0, 1) -> (0, 2) -> (1, 3) -> (2, 4) -> (3, 4) -> (4, 4)
```