CPSC 5610-02, Fall 2023

**Project:** Milestone 1

**Name: Priyadarshini Shanmugasundaram Murugan**

**Pledge:** “I have not received unauthorized aid on this assignment. I understand the answers that I have submitted. The answers submitted have not been directly copied from another source, but instead, are written in my own words.”

---

The objective of this program is to examine a grid of "1s" and identify an emotion based on the path that it takes through it.

This code aims to analyze an 8x8 grid populated with '1s' to identify an emotion based on the path created by traversing these '1s. It employs a breadth-first search (BFS) algorithm to explore and record these paths.

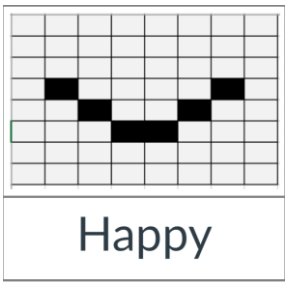
The BFS algorithm maintains a queue and a 'visited' grid to ensure cells are not revisited. It explores adjacent '1s', marking them as visited and extending the path. The direction of movement, either positive (for "happy") or negative (for "sad"), is determined by the initial change in x-coordinates.

The 'determine_emotion' function further scrutinizes the path, considering the path's curvature. It tallies upward and downward curves by assessing changes in both x and y coordinates. If most of the path forms downward curves, it's classified as "happy," and if upward curves prevail, it's labeled as "sad." Otherwise, it's considered "neutral."

This code provides a rudimentary way to interpret emotions based on the spatial arrangement of '1s' in the grid. It finally prints the detected emotion, mapping it to human-readable labels.

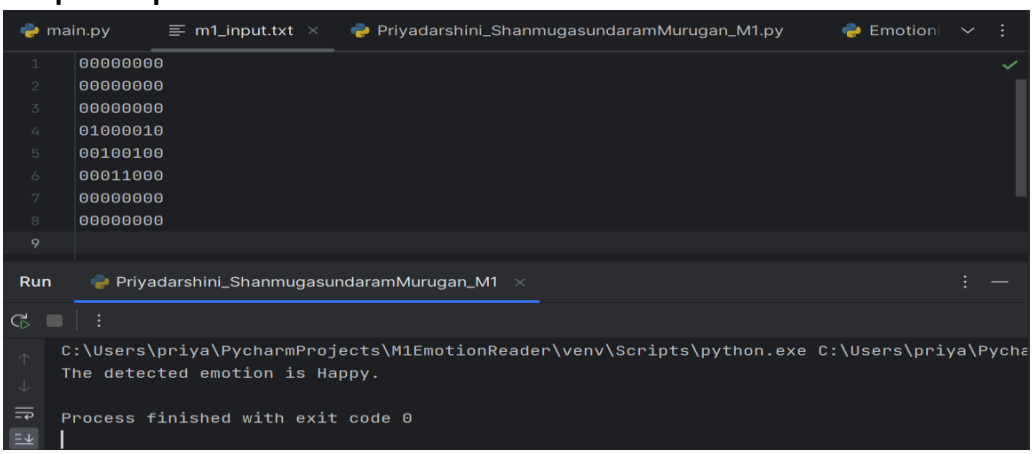
Overall, this code provides a simple emotion detection mechanism by processing a grid's path using BFS and analyzing the path's curvature to determine emotions. It assumes that emotional patterns can be represented in the input grid, with "happy" and "sad" emotions corresponding to specific traversal patterns.

a) **Sample Input: Happy**

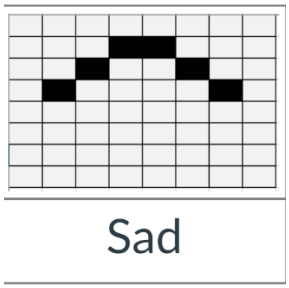



```
00000000
00000000
00000000
01000010
00100100
00011000
00000000
00000000
```

**Sample Output :**

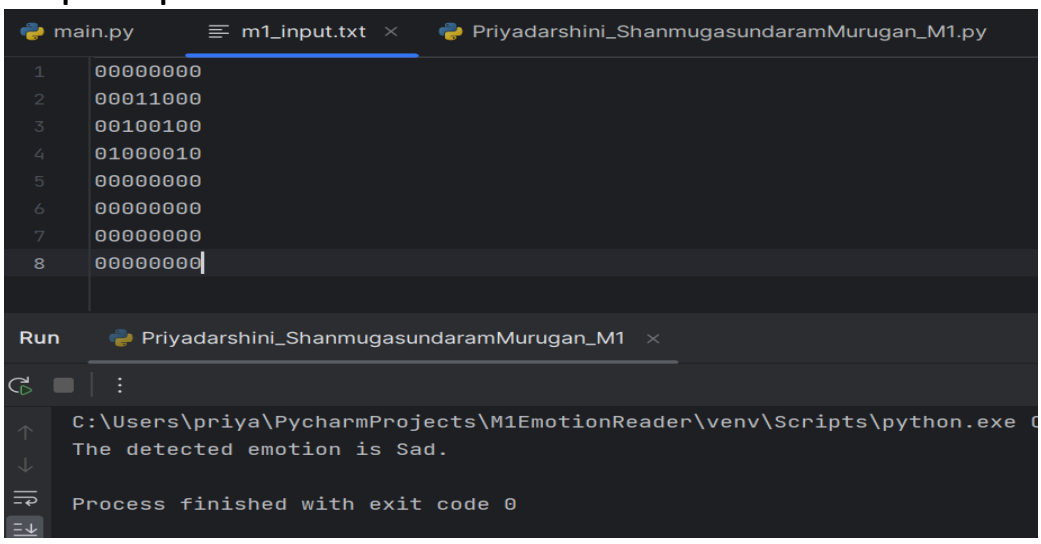



b) **Sample Input: Sad**

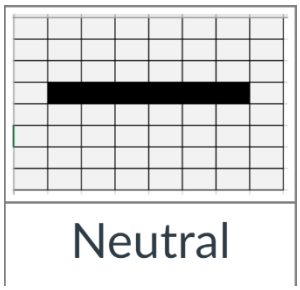



```
00000000
00011000
00100100
01000010
00000000
00000000
00000000
00000000
```

**Sample Output :**

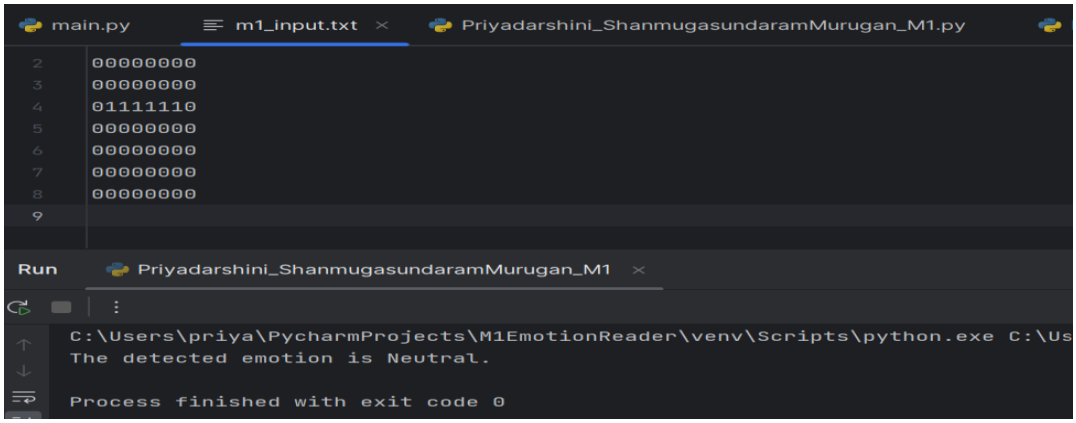


b) **Sample Input: Neutral**

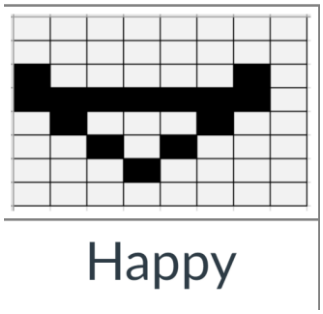



00000000
00000000
00000000
01111110
00000000
00000000
00000000
00000000

**Sample Output :**

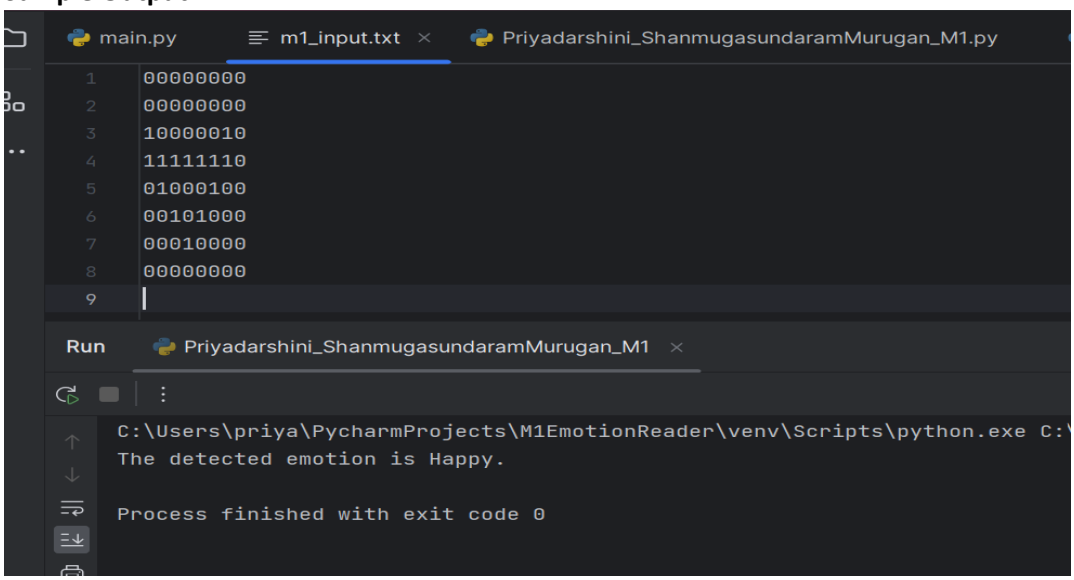



**d) Sample Input: Happy**

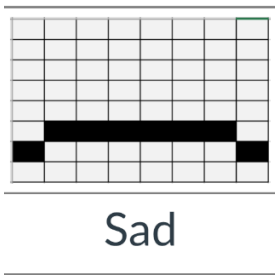



00000000
00000000
10000010
11111110
01000100
00101000
00010000
00000000

**Sample Output :**

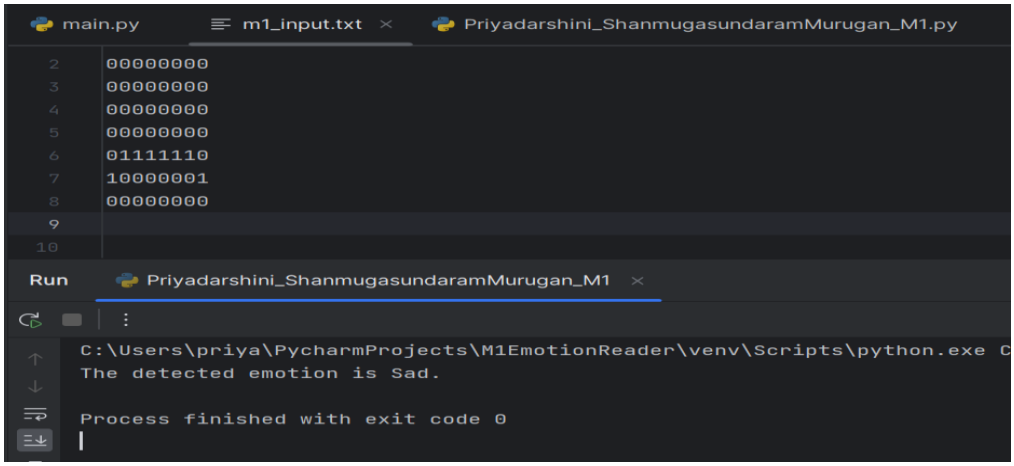


**e) Sample Input: Sad**

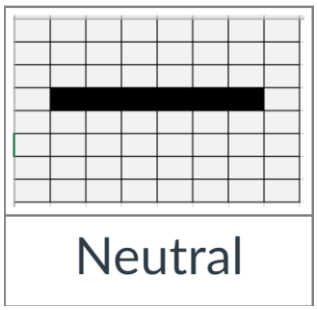



00000000
00000000
00000000
00000000
00000000
01111110
10000001
00000000

**Sample Output :**




**f) Sample Input: Neutral**




00111100
00000000
00000000
00000000
00000000
00000000
00000000
00000000

**Sample Output :**