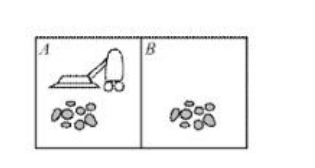
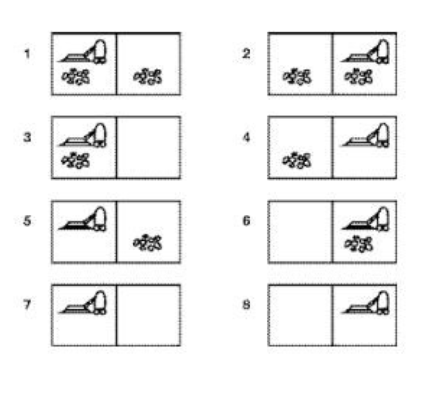
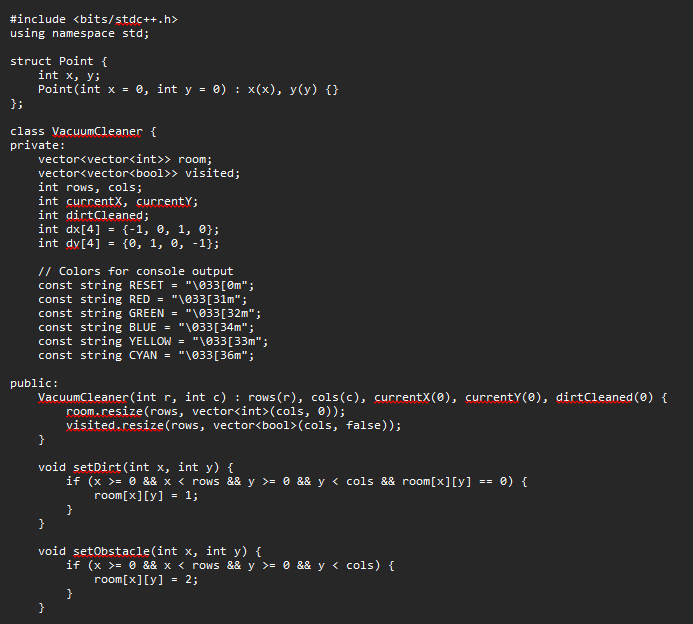
**Practical No: 01**

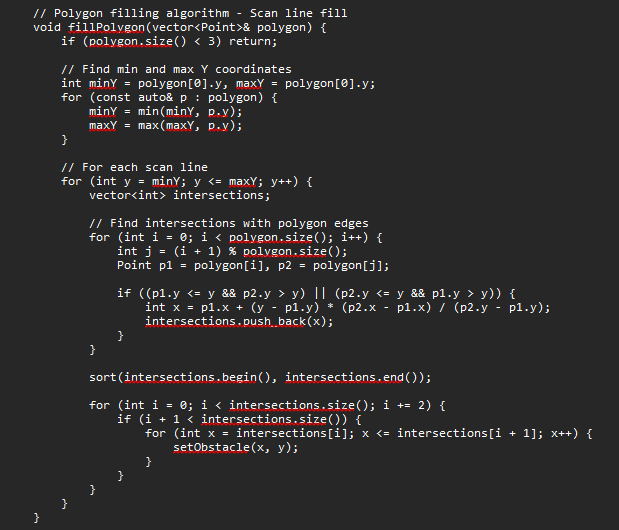
**Aim: Create a vacuum cleaner agent that will detect the dirt and clean it by using c++ or java.**

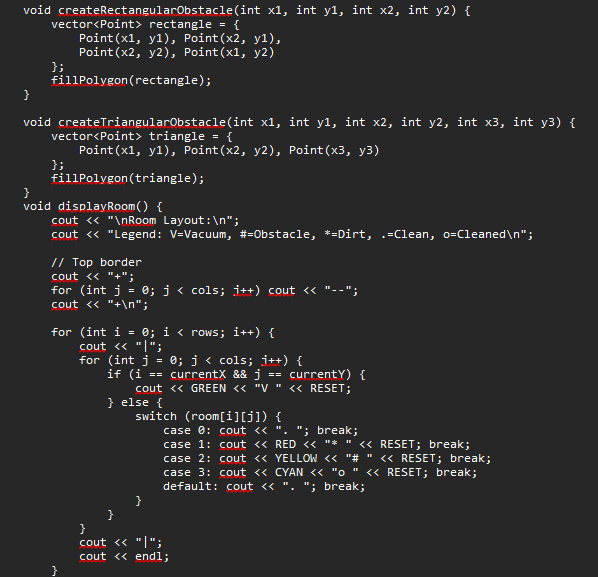
**Vacuum Cleaner Problem in Artificial Intelligence**  
In the simple world, the vacuum cleaner agent has a location sensor and a dirt sensor so that it knows where it is (room A or room B) and whether the room is dirty. It can go left, go right, suck, and idle. A possible performance measure is to maximize the number of clean rooms over a certain period. Each step has an associated cost, be it in power/ energy consumption, noise created, or wear and tear. An efficient and rational agent would maximize performance while mitigating all costs possible. If an agent cleans one square, cleans another, and then cleans the previously cleaned square again, it would be wasting time and power, as well as increasing time and wear & tear. However, if it can efficiently track cleaned and dirty squares, and work out a way of ideally visiting a square only once, this would result in the best-case scenario, mitigated time steps, that would translate into time, noise, power and depreciation savings. Consider the following Vacuum cleaner world:

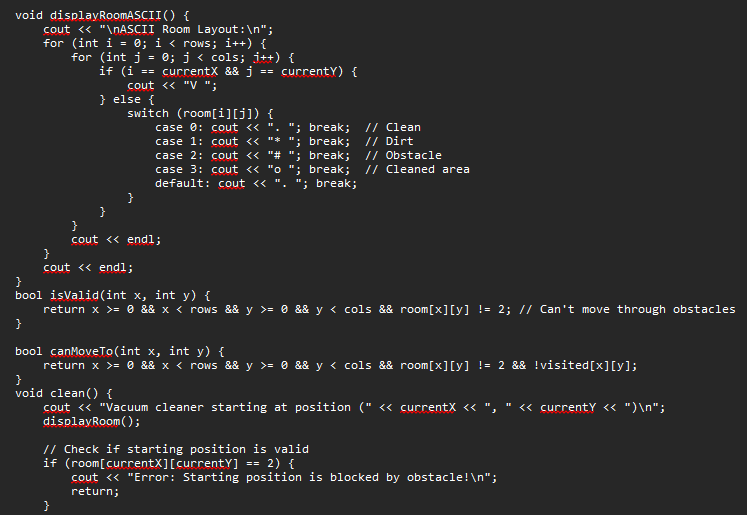
****

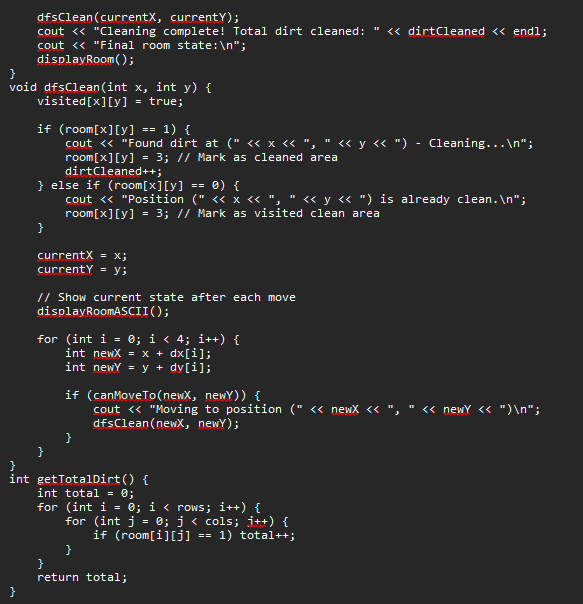
Let's suppose that the world has just two rooms. The robot can be in either room and there can be dirt in zero, one, or two rooms

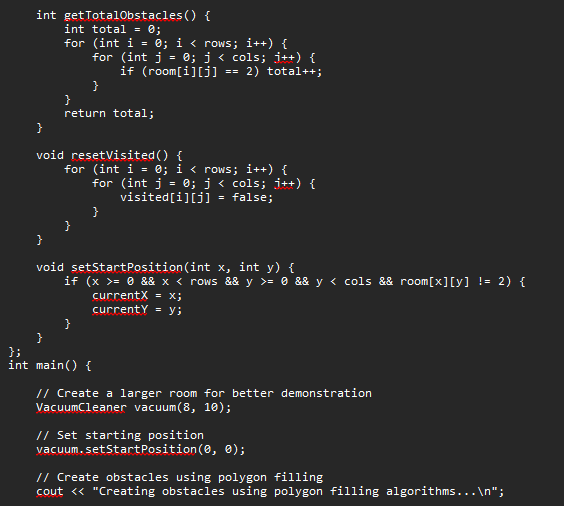
**Code:**

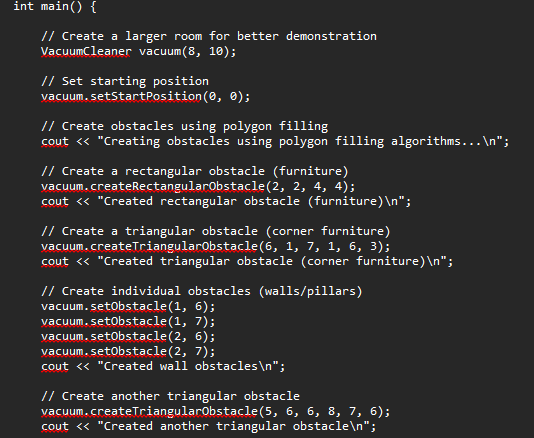
****

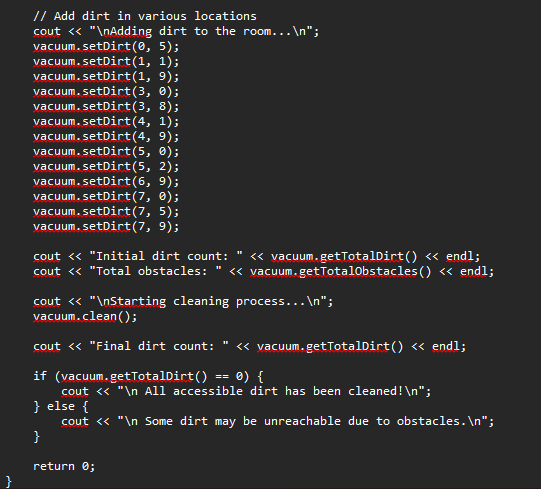
****

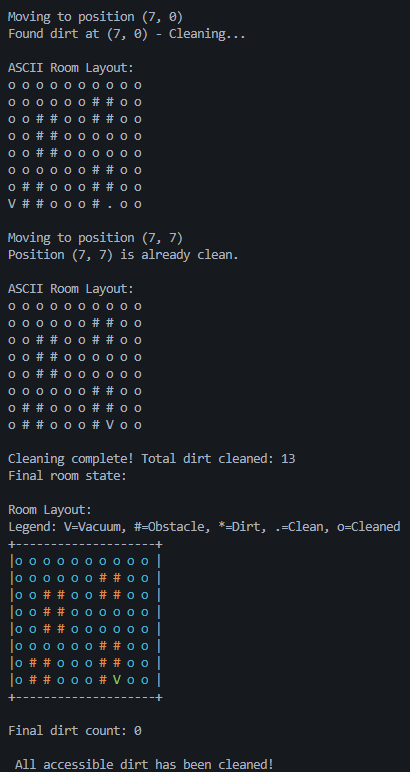
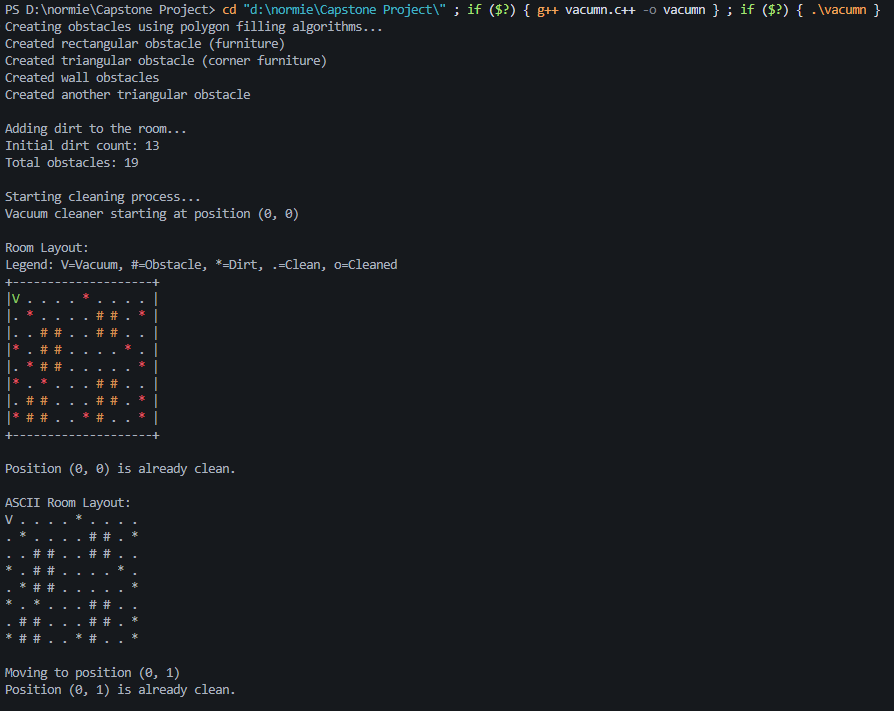
****

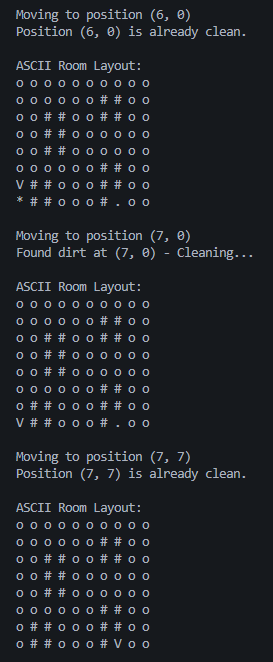
****

****

****

****

**Output:**

****

**Conclusion:**The given C++ program simulates a Vacuum Cleaner Agent using a Depth-First Search (DFS) algorithm to navigate and clean a 2D grid-based room. It starts from the top-left corner (0,0) and recursively visits all reachable cells, checking for and cleaning any dirt (represented by 1). The robot marks visited cells and avoids revisiting them, ensuring efficient traversal. The program visually displays the room layout before and after cleaning, along with real-time updates during the cleaning process. This simulation demonstrates fundamental AI concepts like agent-based movement, environment perception, and autonomous decision-making. It is a practical representation of search algorithms applied in real-world scenarios like robotic vacuum cleaners.