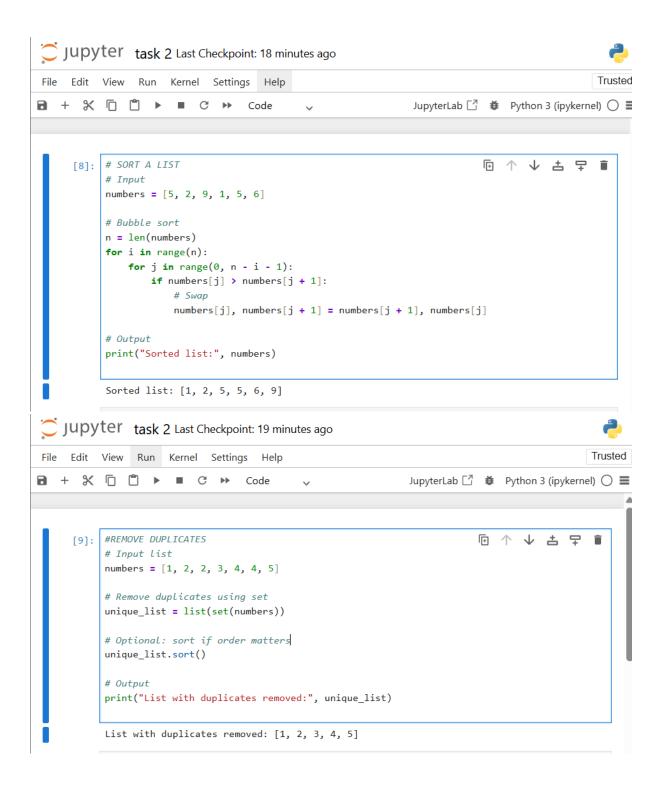
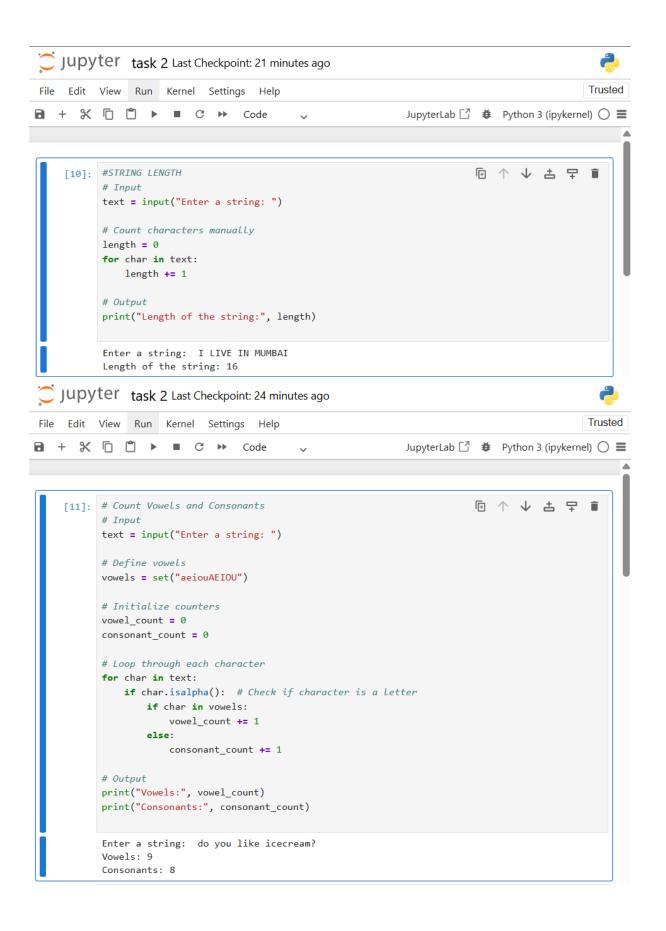
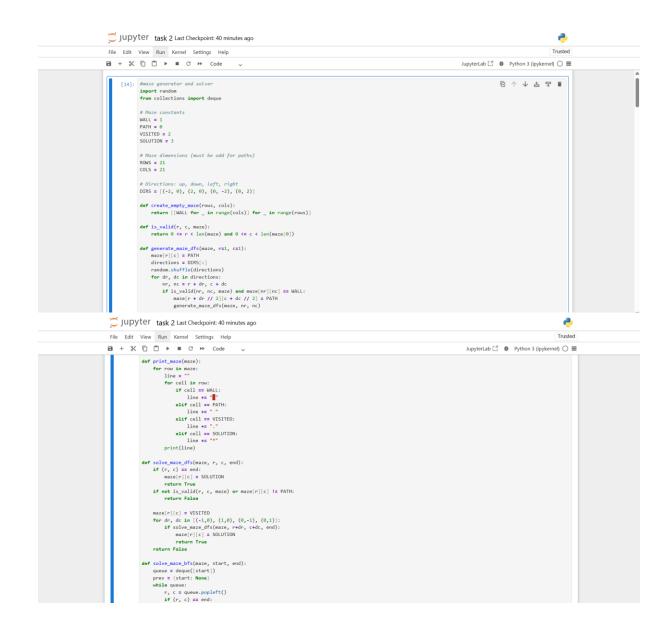


```
[5]: #LCM AND GCD
                                                                           ① ↑ ↓ 古 〒 🗎
           import math
           # Input
           a = int(input("Enter first number (a): "))
           b = int(input("Enter second number (b): "))
           # Calculate GCD
           gcd = math.gcd(a, b)
           # Calculate LCM using the formula: |a * b| / GCD
           lcm = abs(a * b) // gcd
           # Output
           print("GCD:", gcd)
           print("LCM:", lcm)
           Enter first number (a): 18
           Enter second number (b): 24
           GCD: 6
           LCM: 72
Jupyter task 2 Last Checkpoint: 16 minutes ago
                                                                                              Trusted
File Edit View Run Kernel Settings Help
1 + % □ □ ▶ ■ C → Code
                                                               JupyterLab ☐ # Python 3 (ipykernel) ○ ■
      [7]: # LIST REVERSAL
           # Input: Define the list
           original_list = [1, 2, 3, 4, 5]
           # Reversing the list using slicing (no built-in reverse())
           reversed_list = original_list[::-1]
           # Output
           print("Original list:", original_list)
           print("Reversed list:", reversed_list)
           Original list: [1, 2, 3, 4, 5]
           Reversed list: [5, 4, 3, 2, 1]
```







```
break

for dr, dc in [(-1,0), (1,0), (0,-1), (0,1)]:

nr, nc = r + dr, c + dc

if is_valid(nr, nc, maze) and maze[nr][nc] == PATH and (nr, nc) not in prev:

queue.append((nr, nc))

prev[(nr, nc)] = (r, c)

# Reconstruct path
         # Reconstruct path
curr = end
while curr and curr in prev:
    r, c = curr
    maze[r][c] = SOLUTION
    curr = prev[curr]
# Main function
maze = create_empty_maze(ROWS, COLS)
generate_maze_dfs(maze)
start = (1, 1)
end = (ROWS - 2, COLS - 2)
# Choose one of the solvers
# solve_maze_dfs(maze, *start, end)
solve_maze_bfs(maze, start, end)
print_maze(maze)
               print_maze(maze)
```

