

152. Write a program that finds the convex hull of a set of 2D points using the brute force approach.

Program:-

```
def is_counter_clockwise(p, q, r):
```

```
    """ Check if three points p, q, r form a counter-clockwise turn """
```

```
    return (q[1] - p[1]) * (r[0] - q[0]) - (q[0] - p[0]) * (r[1] - q[1]) > 0
```

```
def convex_hull_brute_force(points):
```

```
    """ Find the convex hull of a set of 2D points using brute force approach """
```

```
    n = len(points)
```

```
    if n < 3:
```

```
        return points
```

```
    # Sort points lexicographically (first by x, then by y)
```

```
    points = sorted(points)
```

```
    hull = []
```

```
    for i in range(n):
```

```
        for j in range(i + 1, n):
```

```
            is_hull_edge = True
```

```
            for k in range(n):
```

```
                if k != i and k != j:
```

```
                    if is_counter_clockwise(points[i], points[j], points[k]):
```

```
                        is_hull_edge = False
```

```
                    break
```

```
            if is_hull_edge:
```

```
if points[i] not in hull:  
    hull.append(points[i])
```

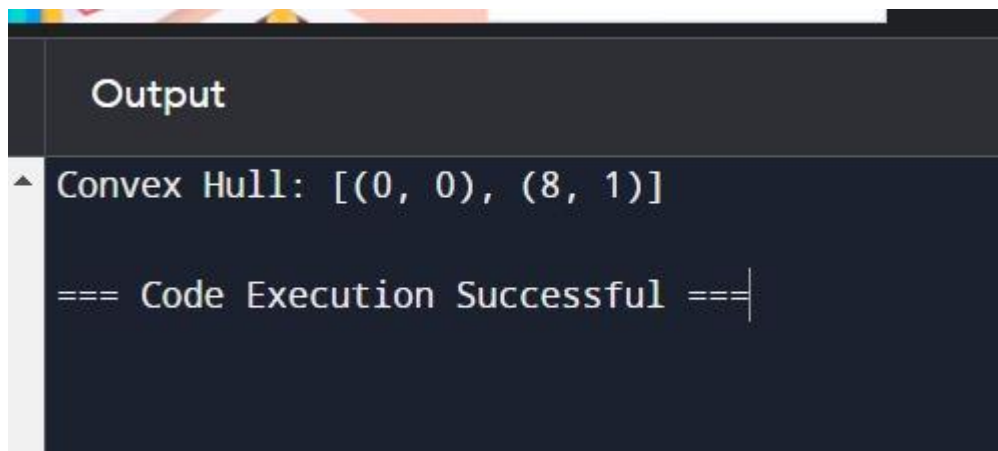
```
if points[j] not in hull:  
    hull.append(points[j])
```

```
return hull
```

input:-

```
points = [(1, 1), (4, 6), (8, 1), (0, 0), (3, 3)]
```

Output:-

A screenshot of a code execution environment. At the top, there is a tab labeled "Output". Below the tab, the text "Convex Hull: [(0, 0), (8, 1)]" is displayed. Below this, a status message "=== Code Execution Successful ===" is shown with a vertical cursor at the end. The background is dark, and the text is light-colored.

```
Output  
^ Convex Hull: [(0, 0), (8, 1)]  
=== Code Execution Successful ===
```

Time complexity:- $O(n^3)$