3.9 FINDING THE K CLOSEST POINTS TO ORIGIN

Question:

Given an array of points where points[i] = [xi, yi] represents a point on the X-Y plane and an integer k, return the k closest points to the origin (0, 0).

AIM

To design a program that returns the k closest points to the origin using Euclidean distance.

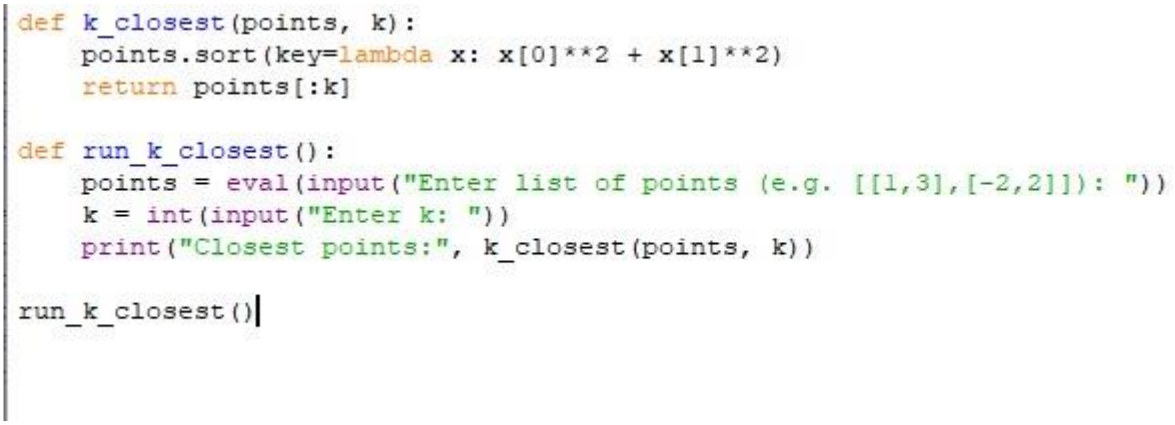
ALGORITHM

1. Start with the given list of points and the integer k.
2. Compute the Euclidean distance of each point from the origin using the formula:

   - Distance = $\sqrt{x^2 + y^2}$

3. Sort the points based on their distance from the origin.
4. Return the first k points from the sorted list

PROGRAM

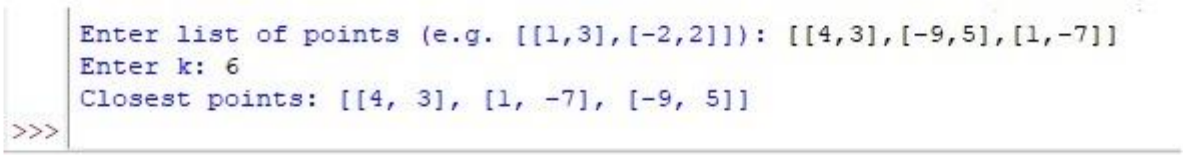


```
def k_closest(points, k):
    points.sort(key=lambda x: x[0]**2 + x[1]**2)
    return points[:k]

def run_k_closest():
    points = eval(input("Enter list of points (e.g. [[1,3],[-2,2]]): "))
    k = int(input("Enter k: "))
    print("Closest points:", k_closest(points, k))

run_k_closest()
```

Input:

Points = [(4,3), (-9,5), (1,-7)]

Output:



```
Enter list of points (e.g. [[1,3],[-2,2]]): [[4,3],[-9,5],[1,-7]]
Enter k: 6
Closest points: [[4, 3], [1, -7], [-9, 5]]
>>>
```

RESULT:

Thus the program is successfully executed and the output is verified.

PERFORMANCE ANALYSIS:

□ Time Complexity: Sorting takes O(n \log n), where n is the number of points.

· Space Complexity: Uses only a few variables, so O(1) (constant extra space).