167. To Implement a function median_of_medians(arr, k) that takes an unsorted array arr and an integer k, and returns the k-th smallest element in the array.

arr = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] k = 6

arr = [23, 17, 31, 44, 55, 21, 20, 18, 19, 27] k = 5

Output: An integer representing the k-th smallest element in the array.

PROGRAM :-

```
import statistics

def median_of_medians(arr, k):
    if len(arr) <= 5:
        return sorted(arr)[k-1]

    sublists = [arr[j:j+5] for j in range(0, len(arr), 5)]
    medians = [statistics.median(sublist) for sublist in sublists]

    pivot = median_of_medians(medians, len(medians)//2)

    low = [elem for elem in arr if elem < pivot]
    high = [elem for elem in arr if elem > pivot]

    if k <= len(low):
        return median_of_medians(low, k)
    elif k > len(arr) - len(high):
        return median_of_medians(high, k - (len(arr) - len(high)))
    else:
        return pivot

# Example Usage
arr = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
k = 6
result = median_of_medians(arr, k)
print(f"The {k}-th smallest element in the array is: {result}")
```

OUTPUT:-

```
The 6-th smallest element in the array is: 6

=== Code Execution Successful ===
```

TIME COMPLEXITY:-O(n)