1. To Implement the Median of Medians algorithm ensures that you handle the worst-case time complexity efficiently while finding the k-th smallest element in an unsorted array.
   arr = [12, 3, 5, 7, 19] k = 2 Expected Output:5
   arr = [12, 3, 5, 7, 4, 19, 26] k = 3 Expected Output:5
   arr = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] k = 6 Expected Output:6

2. 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.

3. Write a program to implement Meet in the Middle Technique. Given an array of integers and a target sum, find the subset whose sum is closest to the target. You will use the Meet in the Middle technique to efficiently find this subset.
   a) Set[] = {45, 34, 4, 12, 5, 2} Target Sum : 42

   b) Set[]= {1, 3, 2, 7, 4, 6} Target sum = 10:

4. Write a program to implement Meet in the Middle Technique. Given a large array of integers and an exact sum E, determine if there is any subset that sums exactly to E. Utilize the Meet in the Middle technique to handle the potentially large size of the array. Return true if there is a subset that sums exactly to E, otherwise return false.
   a) E = {1, 3, 9, 2, 7, 12} exact Sum = 15
   b) E = {3, 34, 4, 12, 5, 2} exact Sum = 15

5. Given two 2×2 Matrices A and B
   A=(1 7 B=(1 3
   3 5) 7 5)
   Use Strassen's matrix multiplication algorithm to compute the product matrix C such that C=A×B.

Test Cases:

Consider the following matrices for testing your implementation:

Test Case 1:

A=(1 7 B=(6 8
3 5), 4 2)

Expected Output:
C=(18 14
62 66)

6. Given two integers X=1234 and Y=5678: Use the Karatsuba algorithm to compute the product Z=X x Y

Test Case 1:

Input: x=1234,y=5678
Expected Output: z=1234×5678=7016652