Solve the following problems using computer with help of Python/C++/Java/C# language as means of communication.

Problem 1: Perfect Numbers

An integer, n, is said to be perfect when the sum of all of the proper divisors of n is equal to n. For example, 28 is a perfect number because its proper divisors are 1, 2, 4, 7 and 14, and 1 + 2 + 4 + 7 + 14 = 28. Create a function named *isPerfect* that determines whether or not a positive integer is perfect. Your function will take one parameter. If that parameter is a perfect number then your function will return true. Otherwise it will return false. In addition, write a main program that uses your function to identify and display all of the perfect numbers between 1 and 10,000.

Problem 2: Collinear Points

Create a function named *isCollinear* that returns true if the three points (x1, y1), (x2, y2), and (x3, y3) lie on the same line, and false otherwise. The three points are collinear if the slope between a pair of points equals to the slope between any other pair of points. Include a main program to demonstrate your function.

Problem 3: Arbitrary Base Conversions

Create a function named *convertBase* that allows the user to convert a number from one base to another. Your program should support bases between 2 and 16 for both the input number and the result number. If the user chooses a base outside of this range then an appropriate error message should be displayed and the program should exit.

Hint: Write sub functions, including a function that converts from an arbitrary base to base 10, a function that converts from base 10 to an arbitrary base, and a main program that reads the bases and input number from the user.

Problem 4: Median of Three Values

Write a function named *medianOfThree* that takes three numbers as parameters, and returns the median value of those parameters as its result. Include a main program that reads three values from the user and displays their median. The median value is the middle of the three values when they are sorted into ascending order.

Problem 5: Approximate PI

The value of π can be approximated by the following infinite series:

$$\pi \approx 3 + \frac{4}{2 \times 3 \times 4} - \frac{4}{4 \times 5 \times 6} + \frac{4}{6 \times 7 \times 8} - \frac{4}{8 \times 9 \times 10} + \frac{4}{10 \times 11 \times 12} - \cdots$$

Create a function *approximatePI* that takes number of approximations required as an input and returns approximated PI value. The first approximation should make use of only the first term from the infinite series. Each additional approximation returned by your function should include one more term in the series, making it a better approximation of π than any of the approximations computed previously. Include a main program to demonstrate your function.