Question 1:

Write a program to calculate the area of a rectangle. The program should prompt the user to enter the length and width of the rectangle. After receiving the input dimensions, the program should calculate and display the area of the rectangle using the provided lengths.

Input Format:

- Enter the length of the rectangle: [user enters a positive numerical value]
- Enter the width of the rectangle: [user enters a positive numerical value]

Output Format:

- The area of the rectangle is [area_value]

Title for Question 1: Calculate the area of a rectangle using data hiding

Solution:

```
#include <iostream>

class Rectangle {
private:
    double length;
    double width;

public:
    void setDimensions(double len, double wid) {
        if (len > 0 && wid > 0) {
            length = len;
            width = wid;
        } else {
            std::cout << "Invalid dimensions. Please provide positive val
        }
    }

    double getArea() const {
        return length * width;
    }
};

int main() {
    Rectangle rect;
    double len, wid;

   // std::cout << "Enter the length of the rectangle: ";
    std::cin >> len;
```

```
   // std::cout << "Enter the width of the rectangle: ";
    std::cin >> wid;

    rect.setDimensions(len, wid);

    std::cout << "The area of the rectangle is " << rect.getArea() << std

    return 0;
}
```

TestCases:

S.No	Inputs	Outputs
1	7.5 4.2	The area of the rectangle is 31.5
2	10 20	The area of the rectangle is 200
3	0.5 0.8	The area of the rectangle is 0.4
4	-5 7	Invalid dimensions. Please provide positive values. The area of the rectangle is 0
5	12.6 -3.4	Invalid dimensions. Please provide positive values. The area of the rectangle is 0
6	15.2 18.9	The area of the rectangle is 287.28

White List:

Black List:

Question 2:

Write a function to calculateAverage that takes an array of integers and its size as input and returns the average of the numbers.

Input Format:

- The program takes an integer as input, representing the number of elements in the array. Then, the user is prompted to enter the individual elements of the array.

Output Format:

- The program calculates and outputs the average of the entered numbers, displayed as a decimal value.

Title for Question 2: Calculate Average using procedural abstraction

Solution:

```
#include <iostream>

// Function prototype
double calculateAverage(const int arr[], int size);

int main() {
    int size;
   // std::cout << "Enter the size of the array: ";
    std::cin >> size;

    int arr[size];
  //  std::cout << "Enter " << size << " integers:\n";
    for (int i = 0; i < size; ++i) {
        std::cin >> arr[i];
    }

    double average = calculateAverage(arr, size);

    std::cout << "Average: " << average << std::endl;

    return 0;
}

// Function definition
double calculateAverage(const int arr[], int size) {
    // Implementation of procedural abstraction
    int sum = 0;
    for (int i = 0; i < size; ++i) {
        sum += arr[i];
    }
    double average = static_cast<double>(sum) / size;
    return average;
}
```

TestCases:

S.No	Inputs	Outputs
1	5 10 20 30 40 50	Average: 30
2	3 5 15 25	Average: 15
3	4 -10 0 10 20	Average: 5
4	1 100	Average: 100
5	0	Average: -nan
6	6 2 4 6 8 10 12	Average: 7

White List:

Black List:

Question 3:

Imagine a situation where 'n' people are in a room, and each person shakes hands with every other person exactly once. Write a program to calculate the maximum number of handshakes that can occur in such a scenario. Implement a class HandshakeCalculator that encapsulates the calculation logic. The class should provide a public method calculateMaxHandshakes that takes the number of people as input and returns the maximum number of handshakes.

Input format

- The program should allow users to input the number of people

Output format

- Display the maximum number of handshakes using the HandshakeCalculator class.

Title for Question 3: Maximum number of handshakes-data hiding

Solution:

```
#include <iostream>
using namespace std;

class HandshakeCalculator {
public:
    int calculateMaxHandshakes(int n) {
        // Formula to calculate maximum handshakes: n * (n - 1) / 2
        return n * (n - 1) / 2;
    }
};

int main() {
    HandshakeCalculator calculator;
    int numPeople;
  //  cout << "Enter the number of people: ";
    cin >> numPeople;

    if (numPeople <= 0) {
        cout << "Number of people should be a positive integer." << endl;
    } else {
        int maxHandshakes = calculator.calculateMaxHandshakes(numPeople);
        cout << "Maximum number of handshakes: " << maxHandshakes << endl
    }

    return 0;
}
```

TestCases:

S.No	Inputs	Outputs
1	5	Maximum number of handshakes: 10
2	10	Maximum number of handshakes: 45
3	12	Maximum number of handshakes: 66
4	7	Maximum number of handshakes: 21
5	9	Maximum number of handshakes: 36
6	82	Maximum number of handshakes: 3321

White List:

Black List:

Question 4:

Given the coordinates (x, y) of a point, write a program to determine in which quadrant the point lies. Implement a class Point that encapsulates the coordinate values and provides a public method getQuadrant to calculate and return the quadrant in which the point lies.

Input Format

- The program should allow users to input the coordinates of the point.

Output Format

- Display the Quadrants in which coordinates lie.

Title for Question 4: Calculate and return the quadrant-data hiding

Solution:

```
#include <iostream>
using namespace std;

class Point {
private:
    double x;
    double y;

public:
    Point(double xCoord, double yCoord) : x(xCoord), y(yCoord) {}

    int getQuadrant() {
```

```
        if (x > 0 && y > 0) {
            return 1;
        } else if (x < 0 && y > 0) {
            return 2;
        } else if (x < 0 && y < 0) {
            return 3;
        } else if (x > 0 && y < 0) {
            return 4;
        } else {
            return 0;  // Origin
        }
    }
};

int main() {
    double xCoord, yCoord;
 //   cout << "Enter the x-coordinate: ";
    cin >> xCoord;
   // cout << "Enter the y-coordinate: ";
    cin >> yCoord;

    Point point(xCoord, yCoord);
    int quadrant = point.getQuadrant();

    if (quadrant == 0) {
        cout << "The point is at the origin." << endl;
    } else {
        cout << "The point lies in Quadrant " << quadrant << endl;
    }

    return 0;
}
```

TestCases:

S.No	Inputs	Outputs
1	3 4	The point lies in Quadrant 1
2	-2 5	The point lies in Quadrant 2
3	-1 -2	The point lies in Quadrant 3
4	4 -3	The point lies in Quadrant 4
5	0 0	The point is at the origin.
6	-5 0	The point is at the origin.

White List:

Black List:

Question 5:

Write a program to determine the number of days in a given month of a given year. The program should prompt the user to enter a year and a month (represented by an integer between 1 and 12). After receiving the inputs, the program should calculate and display the number of days in the specified month of the specified year. Take leap years into account when calculating the number of

days in February.

Input Format:

- Enter the year: [user enters an integer representing the year]
- Enter the month (1-12): [user enters an integer representing the month]

Output Format:

- The number of days in [month name] [year] is [number of days].
- If the input month is not within the range of 1 to 12, display "Invalid month input."

Note: Use January, February, March, ..., December to represent the months.

Title for Question 5: Determine the number of days in a given month of a given year-data abstraction

Solution:

```
#include <iostream>
using namespace std;

class Month {
private:
    int monthNumber;
    int year;

public:
    Month(int month, int yr) : monthNumber(month), year(yr) {}

    bool isLeapYear() {
        return (year % 4 == 0 && year % 100 != 0) || (year % 400 == 0);
    }

    int getDaysInMonth() {
        switch (monthNumber) {
            case 1: case 3: case 5: case 7: case 8: case 10: case 12:
                return 31;
            case 4: case 6: case 9: case 11:
                return 30;
            case 2:
                return isLeapYear() ? 29 : 28;
            default:
                return -1; // Invalid month
        }
    }
};
```

```
int main() {
    int year, month;
    //cout << "Enter the year: ";
    cin >> year;
   // cout << "Enter the month (1-12): ";
    cin >> month;

    Month currentMonth(month, year);

    int days = currentMonth.getDaysInMonth();
    if (days == -1) {
        cout << "Invalid month input." << endl;
    } else {
        string monthNames[] = {
            "", "January", "February", "March", "April", "May", "June", "
            "September", "October", "November", "December"
        };
        cout << "The number of days in " << monthNames[month] << " " << y
    }

    return 0;
}
```

TestCases:

S.No	Inputs	Outputs
1	2023 4	The number of days in April 2023 is 30.
2	1999 13	Invalid month input.
3	2021 6	The number of days in June 2021 is 30.
4	2000 12	The number of days in December 2000 is 31.
5	1998 13	Invalid month input.
6	2100 2	The number of days in February 2100 is 28.

White List:

Black List:

Question 6:

Given a number N. The task is to find the Nth catalan number.

The first few Catalan numbers for N = 0, 1, 2, 3, … are 1, 1, 2, 5, 14, 42, 132, 429, 1430, 4862, … Catalan Number for N is equal to the number of expressions containing N pairs of paranthesis that are correct matched, i.e., for each of the N '(' there exist N ')' on there right and vice versa. Since answer can be huge return answer modulo 1e9+7.

Note: Positions start from 0 as shown above.

Input Format:

- The program expects an integer input, denoted as N, representing the value for which you want to calculate the Nth Catalan number.

Output Format:

- The program outputs an integer, representing the Nth Catalan number modulo 1e9+7.

Constraints:

$1 <= N <= 10^3$

Title for Question 6: Find the Nth catalan number-data abstraction

Solution:

```
#include <iostream>
using namespace std;

const int MOD = 1000000007;

class CatalanCalculator {
public:
    CatalanCalculator(int N) : n(N) {}

    int calculateNthCatalanNumber() {
        int catalan[n + 1];
        catalan[0] = 1;

        for (int i = 1; i <= n; i++) {
            catalan[i] = 0;
            for (int j = 0; j < i; j++) {
                catalan[i] = (catalan[i] + (catalan[j] * catalan[i - j -
            }
        }

        return catalan[n];
    }

private:
    int n;
};

int main() {
    int N;
//    cout << "Enter the value of N: ";
    cin >> N;

    CatalanCalculator calculator(N);
    int nthCatalan = calculator.calculateNthCatalanNumber();

    cout << "Nth Catalan number modulo 1e9+7: " << nthCatalan << endl;

    return 0;
}
```

TestCases:

S.No	Inputs	Outputs
1	0	Nth Catalan number modulo 1e9+7: 1
2	1	Nth Catalan number modulo 1e9+7: 1
3	2	Nth Catalan number modulo 1e9+7: 2
4	3	Nth Catalan number modulo 1e9+7: 5
5	4	Nth Catalan number modulo 1e9+7: 14
6	5	Nth Catalan number modulo 1e9+7: 42

White List:

Black List:
