

```

#include <stdio.h>

int main() {
    long long binary_number; // To store the input binary number
    int last_digit, decimal_value = 0, base = 1; // Variables for calculations

    // Input: Ask the user for a binary number
    printf("Enter a binary number: ");
    scanf("%lld", &binary_number);

    // Process: Convert binary to decimal
    while (binary_number > 0) {
        last_digit = binary_number % 10; // Get the last digit of the binary number
        decimal_value += last_digit * base; // Add its weighted value to the decimal result
        binary_number /= 10; // Remove the last digit
        base *= 2; // Move to the next power of 2
    }

    // Output: Print the decimal equivalent
    printf("Decimal equivalent: %d\n", decimal_value);
    return 0;
}

```

===== PROGRESS REPORT (CLASS X) =====

Roll No	Name	Term1	Term2	Term3	Term4	Total	Average
1	John	85	90	78	88	341	85.25

```

#include <stdio.h> // Include standard library for input/output

int main() {
    int i, j; // Declare variables for loop counters

    // Outer loop: Controls the number of rows
    for (i = 1; i <= 5; i++) {
        // Inner loop: Controls the number of asterisks in each row
        for (j = 1; j <= i; j++) {
            printf("* "); // Print an asterisk followed by a space
        }
        printf("\n"); // Move to the next line after each row
    }

    return 0; // Indicate successful program termination
}

```

```

#include <stdio.h>

// Define the structure for a student
struct Student {
    int roll_no; // Roll number
    char name[21]; // Name (max 20 characters)
    int marks[4]; // Marks for 4 terms
    int total_marks; // Total marks
    float average_marks; // Average marks
};

int main() {
    struct Student classX[20]; // Array of 20 students
    int i, j; // Loop counters

    // Input: Get data for all students
    printf("Enter data for 20 students:\n");
    for (i = 0; i < 20; i++) {
        printf("\nEnter details for Student %d:\n", i + 1);
        printf("Enter Roll Number: ");
        scanf("%d", &classX[i].roll_no);
        printf("Enter Name: ");
        scanf("%s", classX[i].name);

        classX[i].total_marks = 0; // Initialize total marks

        // Input marks for each term
        for (j = 0; j < 4; j++) {
            printf("Enter marks for Term %d: ", j + 1);
            scanf("%d", &classX[i].marks[j]);
            classX[i].total_marks += classX[i].marks[j]; // Add term marks to total
        }

        // Calculate average marks
        classX[i].average_marks = (float)classX[i].total_marks / 4.0;
    }

    // Output: Display the progress report
    printf("\n==== PROGRESS REPORT (CLASS X) =====\n");
    printf("-----\n");
    printf("Roll No | Name | Term1 | Term2 | Term3 | Term4 | Total | Average\n");
    printf("-----\n");

    for (i = 0; i < 20; i++) {
        printf("%-7d | %-12s |", classX[i].roll_no, classX[i].name); // Roll number and name
        for (j = 0; j < 4; j++) {
            printf(" %-5d |", classX[i].marks[j]); // Marks for each term
        }
        printf(" %-5d | %-7.2f\n", classX[i].total_marks, classX[i].average_marks); // Total and aver
    }

    return 0;
}

```

Successfully opened input.txt for reading.
 Successfully opened even_file.txt for writing.
 Successfully opened odd_file.txt for writing.

 Reading numbers from input.txt and separating...
 Processing complete.
 All files closed.

```

#include <stdio.h> // Include standard input/output library

int main() {
    // Declare 3x3 matrices A, B, C, the sum matrix SumBC, and the resultant matrix D
    double A[3][3], B[3][3], C[3][3], SumBC[3][3], D[3][3];
    int i, j, k; // Loop counters

    // Input elements for Matrix A
    printf("Enter elements of matrix A (3x3):\n");
    for (i = 0; i < 3; i++) {
        for (j = 0; j < 3; j++) {
            printf("Enter element A[%d][%d]: ", i, j);
            scanf("%lf", &A[i][j]);
        }
    }

    // Input elements for Matrix B
    printf("\nEnter elements of matrix B (3x3):\n");
    for (i = 0; i < 3; i++) {
        for (j = 0; j < 3; j++) {
            printf("Enter element B[%d][%d]: ", i, j);
            scanf("%lf", &B[i][j]);
        }
    }

    // Input elements for Matrix C
    printf("\nEnter elements of matrix C (3x3):\n");
    for (i = 0; i < 3; i++) {
        for (j = 0; j < 3; j++) {
            printf("Enter element C[%d][%d]: ", i, j);
            scanf("%lf", &C[i][j]);
        }
    }

    // Calculate SumBC = B + C (Matrix Addition)
    for (i = 0; i < 3; i++) {
        for (j = 0; j < 3; j++) {
            SumBC[i][j] = B[i][j] + C[i][j];
        }
    }
    printf("\nMatrix (B + C) calculated.\n");

    // Calculate D = A * SumBC (Matrix Multiplication)
    for (i = 0; i < 3; i++) {
        for (j = 0; j < 3; j++) {
            D[i][j] = 0; // Initialize the element D[i][j] to 0
            for (k = 0; k < 3; k++) {
                D[i][j] += A[i][k] * SumBC[k][j];
            }
        }
    }
    printf("Matrix D = A * (B + C) calculated.\n");

    // Display the resultant Matrix D
    printf("\nResultant Matrix D (A * (B + C)):\n");
    for (i = 0; i < 3; i++) {
        for (j = 0; j < 3; j++) {
            printf("%10.2lf ", D[i][j]); // Print the element with formatting
        }
        printf("\n"); // Print newline after each row
    }

    return 0; // Indicate successful program termination
}

```

```

#include <stdio.h> // For file handling functions like fopen(), fclose(), fprintf(), fscanf()
#include <stdlib.h> // For exit()

int main() {
    FILE *input_file_ptr; // Pointer for the input file
    FILE *even_file_ptr; // Pointer for the file to store even numbers
    FILE *odd_file_ptr; // Pointer for the file to store odd numbers
    int number; // Variable to hold the number read from the file

    // Step 1: Open the input file for reading
    input_file_ptr = fopen("input.txt", "r");
    if (input_file_ptr == NULL) {
        printf("Error: Could not open input.txt for reading.\n");
        exit(1);
    }
    printf("Successfully opened input.txt for reading.\n");

    // Step 2: Open the output file for even numbers in write mode
    even_file_ptr = fopen("even_file.txt", "w");
    if (even_file_ptr == NULL) {
        printf("Error: Could not open even_file.txt for writing.\n");
        fclose(input_file_ptr);
        exit(1);
    }
    printf("Successfully opened even_file.txt for writing.\n");

    // Step 3: Open the output file for odd numbers in write mode
    odd_file_ptr = fopen("odd_file.txt", "w");
    if (odd_file_ptr == NULL) {
        printf("Error: Could not open odd_file.txt for writing.\n");
        fclose(input_file_ptr);
        fclose(even_file_ptr);
        exit(1);
    }
    printf("Successfully opened odd_file.txt for writing.\n");

    printf("\nReading numbers from input.txt and separating...\n");

    // Step 4: Read numbers from the input file and write to the respective output files
    while (fscanf(input_file_ptr, "%d", &number) == 1) {
        if (number % 2 == 0) {
            // Write even number to even_file.txt
            fprintf(even_file_ptr, "%d\n", number);
        } else {
            // Write odd number to odd_file.txt
            fprintf(odd_file_ptr, "%d\n", number);
        }
    }
    printf("Processing complete.\n");

    // Step 5: Close all opened files
    fclose(input_file_ptr);
    fclose(even_file_ptr);
    fclose(odd_file_ptr);
    printf("All files closed.\n");

    return 0; // Indicate successful program termination
}

```