while_dowhile_loop.c

```c
#include <stdio.h>
// Example of while and do-while loops in C
int main() {
    int n;
    // Prompt user for input
    printf("Enter a positive integer: ");
    scanf("%d", &n);
    // Print numbers from 1 to n
    int i = 1;
    while (i <= n) {
        printf("%d\n", i);
        i++;
    }

    // example of do-while loop
    int j = 1;
    printf("Using do-while loop:\n");
    do {
        printf("%d\n", j);
        j++;
    } while (j <= n);

    return 0;
}

/*************************
 * Output Example:
 * Enter a positive integer: 5
 * 1
 * 2
 * 3
 * 4
 * 5
 * Using do-while loop:
 * 1
 * 2
 * 3
 * 4
 * 5
 *************************
 */

```

factorial.c

```c
#include <stdio.h>

// write a function to calculate factorial of a number using for loop

int main(){

    int num;
    printf("Enter a number: ");
    scanf("%d", &num);

    // Factorial calculation
    int fact = 1;
    for(int i = 1; i <= num; i++){
        fact = fact * i;
    }
    printf("Factorial of %d is %d\n", num, fact);
    return 0;
}


/**
 * What happens in the loop?

Loop runs from i = 1 to i = num

Each time, multiply the current value of fact by i

This builds the factorial step-by-step

Example for num = 5:

i   fact
--------------
1   1 × 1 = 1
2   1 × 2 = 2
3   2 × 3 = 6
4   6 × 4 = 24
5   24 × 5 = 120

Final value = 120
 */


```

array_1.c

```c
#include <stdio.h>

// example of array in C

int main(){

    int marks[5] = {85, 90, 78, 92, 88};
    for(int i = 0; i < 5; i++){
        printf("Marks of student %d: %d\n", i+1, marks[i]);
    }
    return 0;
}

/**
 * Example Output:
Marks of student 1: 85
Marks of student 2: 90
Marks of student 3: 78
Marks of student 4: 92
Marks of student 5: 88
 */
```

array_2.c

```c
#include <stdio.h>

// Two dimensional array example in C

int main(){

    /*
    int matrix[3][3] = {
        {1, 2, 3},
        {4, 5, 6},
        {7, 8, 9}
    };

    // Print the matrix
    for(int i = 0; i < 3; i++){
        for(int j = 0; j < 3; j++){
            printf("%d ", matrix[i][j]);
        }
        printf("\n");
    }
    */

    int i=0,j=0;
    int arr[4][3]={{1,2,3},{2,3,4},{3,4,5},{4,5,6}};

    for(i=0;i<4;i++){
        for(j=0;j<3;j++){
            printf("arr[%d] [%d] = %d \n",i,j,arr[i][j]);
        }
    }

    return 0;
}
```

rightHalfPyramid.c

```c
#include <stdio.h>

// right half pyramid pattern

int main() {
    int rows = 5;

    // This loop for traverse
    // pyramid from top to bottom
    for (int i = 0; i < rows; i++) {

        // Inner loop for printing
        // character in each rows
        for (int j = 0; j <= i; j++) {
            printf("* ");
        }
        printf("\n");
    }
    return 0;
}

/**
 * Output:
  *
  * *
  * * *
  * * * *
  * * * * *
 */
```

matrixAddition.c

```c
#include <stdio.h>

// Function to add two matrices of same dimensions

int main(){

    // Declaring two 2x3 matrices and a result matrix
    int a[2][3], b[2][3], sum[2][3], i,j;

    // Taking input for first matrix
    printf("Enter elements of first matrix:\n");
    for(i=0; i<2; i++){
        for(j=0; j<3; j++){
            printf("Element [%d][%d]: ", i, j);
            scanf("%d", &a[i][j]);
        }
    }

    // Taking input for second matrix
    printf("Enter elements of second matrix:\n");
    for(i=0; i<2; i++){
        for(j=0; j<3; j++){
            printf("Element [%d][%d]: ", i, j);
            scanf("%d", &b[i][j]);
        }
    }

    // Displaying the first matrix
    printf("First matrix:\n");
    for(i=0; i<2; i++){
        for(j=0; j<3; j++){
            printf("%d ", a[i][j]);
        }
        printf("\n");
    }

    // Displaying the second matrix
    printf("Second matrix:\n");
    for(i=0; i<2; i++){
        for(j=0; j<3; j++){
            printf("%d ", b[i][j]);
        }
        printf("\n");
    }

    // Adding the two matrices
    for(i=0; i<2; i++){
        for(j=0; j<3; j++){
            sum[i][j] = a[i][j] + b[i][j];
        }
    }
```

```c

    // Displaying the sum
    printf("Sum of the two matrices:\n");
    for(i=0; i<2; i++){
        for(j=0; j<3; j++){
            printf("%d ", sum[i][j]);
        }
        printf("\n");
    }

    return 0;
}


/**
 * *********************************
 * Example Input/Output:
 * *********************************
Enter elements of first matrix:
Element [0][0]: 2
Element [0][1]: 1
Element [0][2]: 3
Element [1][0]: 4
Element [1][1]: 5
Element [1][2]: 6
Enter elements of second matrix:
Element [0][0]: 6
Element [0][1]: 1
Element [0][2]: 9
Element [1][0]: 5
Element [1][1]: 4
Element [1][2]: 5
First matrix:
2 1 3
4 5 6
Second matrix:
6 1 9
5 4 5
Sum of the two matrices:
8 2 12
9 9 11
************************************/
```

matrixSubstraction.c

```c
#include <stdio.h>

int main() {
    int rows, cols;

    // Get dimensions from the user
    printf("Enter the number of rows: ");
    scanf("%d", &rows);
    printf("Enter the number of columns: ");
    scanf("%d", &cols);

    // Declare matrices
    int matrixA[rows][cols];
    int matrixB[rows][cols];
    int resultMatrix[rows][cols];

    // Input elements for matrix A
    printf("\nEnter elements for Matrix A:\n");
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            printf("Enter element A[%d][%d]: ", i, j);
            scanf("%d", &matrixA[i][j]);
        }
    }

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

    // display matrix A
    printf("\nMatrix A:\n");
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            printf("%d\t", matrixA[i][j]);
        }
        printf("\n");
    }

    // display matrix B
    printf("\nMatrix B:\n");
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            printf("%d\t", matrixB[i][j]);
        }
        printf("\n");
    }
```

```c

    // Perform matrix subtraction (A - B)
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            resultMatrix[i][j] = matrixA[i][j] - matrixB[i][j];
        }
    }

    // Print the result matrix
    printf("\nResult of Matrix A - Matrix B:\n");
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            printf("%d\t", resultMatrix[i][j]);
        }
        printf("\n");
    }

    return 0;
}

/**
 * Example Input/Output:
 * Enter the number of rows: 2
Enter the number of columns: 2

Enter elements for Matrix A:
Enter element A[0][0]: 1
Enter element A[0][1]: 4
Enter element A[1][0]: 5
Enter element A[1][1]: 6

Enter elements for Matrix B:
Enter element B[0][0]: 2
Enter element B[0][1]: 3
Enter element B[1][0]: 1
Enter element B[1][1]: 6

Matrix A:
1       4
5       6

Matrix B:
2       3
1       6

Result of Matrix A - Matrix B:
-1      1
4       0
 */
```