

while_dowhile_loop.c

```
1 #include <stdio.h>
2 // Example of while and do-while loops in C
3 int main() {
4     int n;
5     // Prompt user for input
6     printf("Enter a positive integer: ");
7     scanf("%d", &n);
8     // Print numbers from 1 to n
9     int i = 1;
10    while (i <= n) {
11        printf("%d\n", i);
12        i++;
13    }
14
15    // example of do-while loop
16    int j = 1;
17    printf("Using do-while loop:\n");
18    do {
19        printf("%d\n", j);
20        j++;
21    } while (j <= n);
22
23    return 0;
24}
25
26 ****
27 * Output Example:
28 * Enter a positive integer: 5
29 * 1
30 * 2
31 * 3
32 * 4
33 * 5
34 * Using do-while loop:
35 * 1
36 * 2
37 * 3
38 * 4
39 * 5
40 ****
41 */
42
```

factorial.c

```
1 #include <stdio.h>
2
3 // write a function to calculate factorial of a number using for loop
4
5 int main(){
6
7     int num;
8     printf("Enter a number: ");
9     scanf("%d", &num);
10
11    // Factorial calculation
12    int fact = 1;
13    for(int i = 1; i <= num; i++){
14        fact = fact * i;
15    }
16    printf("Factorial of %d is %d\n", num, fact);
17    return 0;
18}
19
20
21 /**
22 * What happens in the loop?
23
24 Loop runs from i = 1 to i = num
25
26 Each time, multiply the current value of fact by i
27
28 This builds the factorial step-by-step
29
30 Example for num = 5:
31
32 i    fact
33 -----
34 1    1 × 1 = 1
35 2    1 × 2 = 2
36 3    2 × 3 = 6
37 4    6 × 4 = 24
38 5    24 × 5 = 120
39
40 Final value = 120
41 */
42
43
```

```
array_1.c

1 #include <stdio.h>
2
3 // example of array in C
4
5 int main(){
6
7     int marks[5] = {85, 90, 78, 92, 88};
8     for(int i = 0; i < 5; i++){
9         printf("Marks of student %d: %d\n", i+1, marks[i]);
10    }
11    return 0;
12 }
13
14 /**
15 * Example Output:
16 Marks of student 1: 85
17 Marks of student 2: 90
18 Marks of student 3: 78
19 Marks of student 4: 92
20 Marks of student 5: 88
21 */
```

array_2.c

```
1 #include <stdio.h>
2
3 // Two dimensional array example in C
4
5 int main(){
6
7     /*
8     int matrix[3][3] = {
9         {1, 2, 3},
10        {4, 5, 6},
11        {7, 8, 9}
12    };
13
14    // Print the matrix
15    for(int i = 0; i < 3; i++){
16        for(int j = 0; j < 3; j++){
17            printf("%d ", matrix[i][j]);
18        }
19        printf("\n");
20    }
21 */
22
23 int i=0,j=0;
24 int arr[4][3]={{1,2,3},{2,3,4},{3,4,5},{4,5,6}};
25
26 for(i=0;i<4;i++){
27     for(j=0;j<3;j++){
28         printf("arr[%d] [%d] = %d \n",i,j,arr[i][j]);
29     }
30 }
31
32 return 0;
33 }
```

rightHalfPyramid.c

```
1 #include <stdio.h>
2
3 // right half pyramid pattern
4
5 int main() {
6     int rows = 5;
7
8     // This loop for traverse
9     // pyramid from top to bottom
10    for (int i = 0; i < rows; i++) {
11
12        // Inner loop for printing
13        // character in each rows
14        for (int j = 0; j <= i; j++) {
15            printf("* ");
16        }
17        printf("\n");
18    }
19    return 0;
20}
21
22 /**
23 * Output:
24 *
25 * *
26 * * *
27 * * * *
28 * * * * *
29 */
```

matrixAddition.c

```
1 #include <stdio.h>
2
3 // Function to add two matrices of same dimensions
4
5 int main(){
6
7     // Declaring two 2x3 matrices and a result matrix
8     int a[2][3], b[2][3], sum[2][3], i,j;
9
10    // Taking input for first matrix
11    printf("Enter elements of first matrix:\n");
12    for(i=0; i<2; i++){
13        for(j=0; j<3; j++){
14            printf("Element [%d][%d]: ", i, j);
15            scanf("%d", &a[i][j]);
16        }
17    }
18
19    // Taking input for second matrix
20    printf("Enter elements of second matrix:\n");
21    for(i=0; i<2; i++){
22        for(j=0; j<3; j++){
23            printf("Element [%d][%d]: ", i, j);
24            scanf("%d", &b[i][j]);
25        }
26    }
27
28    // Displaying the first matrix
29    printf("First matrix:\n");
30    for(i=0; i<2; i++){
31        for(j=0; j<3; j++){
32            printf("%d ", a[i][j]);
33        }
34        printf("\n");
35    }
36
37    // Displaying the second matrix
38    printf("Second matrix:\n");
39    for(i=0; i<2; i++){
40        for(j=0; j<3; j++){
41            printf("%d ", b[i][j]);
42        }
43        printf("\n");
44    }
45
46    // Adding the two matrices
47    for(i=0; i<2; i++){
48        for(j=0; j<3; j++){
49            sum[i][j] = a[i][j] + b[i][j];
50        }
51    }
```

```
52 // Displaying the sum
53 printf("Sum of the two matrices:\n");
54 for(i=0; i<2; i++){
55     for(j=0; j<3; j++){
56         printf("%d ", sum[i][j]);
57     }
58     printf("\n");
59 }
60 }
61
62 return 0;
}
63
64
65
66 /**
67 * ****
68 * Example Input/Output:
69 * ****
70 Enter elements of first matrix:
71 Element [0][0]: 2
72 Element [0][1]: 1
73 Element [0][2]: 3
74 Element [1][0]: 4
75 Element [1][1]: 5
76 Element [1][2]: 6
77 Enter elements of second matrix:
78 Element [0][0]: 6
79 Element [0][1]: 1
80 Element [0][2]: 9
81 Element [1][0]: 5
82 Element [1][1]: 4
83 Element [1][2]: 5
84 First matrix:
85 2 1 3
86 4 5 6
87 Second matrix:
88 6 1 9
89 5 4 5
90 Sum of the two matrices:
91 8 2 12
92 9 9 11
93 *****/
```

matrixSubtraction.c

```
1 #include <stdio.h>
2
3 int main() {
4     int rows, cols;
5
6     // Get dimensions from the user
7     printf("Enter the number of rows: ");
8     scanf("%d", &rows);
9     printf("Enter the number of columns: ");
10    scanf("%d", &cols);
11
12    // Declare matrices
13    int matrixA[rows][cols];
14    int matrixB[rows][cols];
15    int resultMatrix[rows][cols];
16
17    // Input elements for matrix A
18    printf("\nEnter elements for Matrix A:\n");
19    for (int i = 0; i < rows; i++) {
20        for (int j = 0; j < cols; j++) {
21            printf("Enter element A[%d][%d]: ", i, j);
22            scanf("%d", &matrixA[i][j]);
23        }
24    }
25
26    // Input elements for matrix B
27    printf("\nEnter elements for Matrix B:\n");
28    for (int i = 0; i < rows; i++) {
29        for (int j = 0; j < cols; j++) {
30            printf("Enter element B[%d][%d]: ", i, j);
31            scanf("%d", &matrixB[i][j]);
32        }
33    }
34
35    // display matrix A
36    printf("\nMatrix A:\n");
37    for (int i = 0; i < rows; i++) {
38        for (int j = 0; j < cols; j++) {
39            printf("%d\t", matrixA[i][j]);
40        }
41        printf("\n");
42    }
43
44    // display matrix B
45    printf("\nMatrix B:\n");
46    for (int i = 0; i < rows; i++) {
47        for (int j = 0; j < cols; j++) {
48            printf("%d\t", matrixB[i][j]);
49        }
50        printf("\n");
51    }
```

```
52
53     // Perform matrix subtraction (A - B)
54     for (int i = 0; i < rows; i++) {
55         for (int j = 0; j < cols; j++) {
56             resultMatrix[i][j] = matrixA[i][j] - matrixB[i][j];
57         }
58     }
59
60     // Print the result matrix
61     printf("\nResult of Matrix A - Matrix B:\n");
62     for (int i = 0; i < rows; i++) {
63         for (int j = 0; j < cols; j++) {
64             printf("%d\t", resultMatrix[i][j]);
65         }
66         printf("\n");
67     }
68
69     return 0;
70 }
71
72 /**
73 * Example Input/Output:
74 * Enter the number of rows: 2
75 Enter the number of columns: 2
76
77 Enter elements for Matrix A:
78 Enter element A[0][0]: 1
79 Enter element A[0][1]: 4
80 Enter element A[1][0]: 5
81 Enter element A[1][1]: 6
82
83 Enter elements for Matrix B:
84 Enter element B[0][0]: 2
85 Enter element B[0][1]: 3
86 Enter element B[1][0]: 1
87 Enter element B[1][1]: 6
88
89 Matrix A:
90 1      4
91 5      6
92
93 Matrix B:
94 2      3
95 1      6
96
97 Result of Matrix A - Matrix B:
98 -1      1
99 4      0
100 */
```