

digitSum.c

```
1 #include <stdio.h>
2
3 int main(){
4
5     int n, sum = 0;
6     printf("Enter a number: ");
7     scanf("%d", &n);
8
9
10    while(n != 0){
11        sum = sum + (n % 10); // Add the last digit to sum
12        n = n / 10; // Remove the last digit
13    }
14
15
16    // using for loop
17    /*
18    for(; n != 0; n /= 10){
19        sum += n % 10;
20    }
21    */
22
23    printf("Sum of digits: %d\n", sum);
24    return 0;
25 }
```

function.c

```
1 #include <stdio.h>
2
3 // Create a function
4 void calculateSum(int a, int b) {
5     int sum = a + b;
6     printf("The sum of %d + %d is: %d\n", a, b, sum);
7 }
8
9 int main() {
10
11     int x, y;
12     printf("Enter first number: ");
13     scanf("%d", &x);
14     printf("Enter second number: ");
15     scanf("%d", &y);
16     calculateSum(x, y); // call the function
17
18 //calculateSum(5, 10); // call the function
19 return 0;
20 }
```

array1.c

```
1 #include <stdio.h>
2
3 int main(){
4     // integer array of size 5
5     int arr[5] = {10, 20, 30, 40, 50};
6
7     // for loop to print integer array elements
8     for(int i = 0; i < 5; i++){
9         printf("%d ", arr[i]);
10    }
11
12    // reverse order printing
13    printf("\n");
14    for(int i = 4; i >= 0; i--){
15        printf("%d ", arr[i]);
16    }
17
18    // float array of size 3
19    float arrf[3] = {1.1, 2.2, 3.3};
20    printf("\n");
21    for(int i = 0; i < 3; i++){
22        printf("%.1f ", arrf[i]);
23    }
24
25    // char array of size 4
26    char arrc[4] = {'A', 'B', 'C', 'D'};
27    printf("\n");
28    for(int i = 0; i < 4; i++){
29        printf("%c ", arrc[i]);
30    }
31
32    printf("\n");
33    return 0;
34 }
```

array2.c

```
1 #include <stdio.h>
2
3 int main()
4 {
5     // Declare a 2D array: 4 names, max 20 chars each
6     char arrc[4][20] = {"Apple", "Ball", "Cat", "Dog"};
7
8     for(int i = 0; i < 4; i++){
9         printf("%s ", arrc[i]);
10    }
11
12    printf("\n");
13
14    return 0;
15 }
```

array3.c

```
1 #include <stdio.h>
2
3 // Give an array of marks of 10 students, if the marks of any student is less than
4 // 35, print its roll number. [here roll number refers to index of array]
5
6 int main(){
7     int marks[10] = {45, 32, 67, 29, 50, 80, 22, 90, 33, 55};
8
9     for(int i = 0; i < 10; i++){
10         if(marks[i] < 35){
11             printf("Student with roll number %d has failed.\n", i);
12         }
13     }
14
15     printf("\n");
16
17     // take input of 5 students marks and print them
18     int inputMarks[10];
19     printf("Enter marks of 10 students: \n");
20     for(int i = 0; i < 10; i++){
21         printf("Student %d: ", i+1);
22         scanf("%d", &inputMarks[i]);
23     }
24
25     printf("You entered: ");
26     for(int i = 0; i < 10; i++){
27         printf("%d ", inputMarks[i]);
28     }
29     printf("\n");
30
31     // check which students have less than 35 marks
32     for(int i = 0; i < 10; i++){
33         if(inputMarks[i] < 35){
34             printf("Student with roll number %d has failed.\n", i);
35         }
36     }
37
38     return 0;
39 }
```

2darray.c

```
1 #include <stdio.h>
2
3 int main(){
4
5     int arr[3][4] = {
6         {1, 2, 3, 4},
7         {5, 6, 7, 8},
8         {9, 10, 11, 12}
9     };
10
11    for (int i = 0; i < 3; i++){
12        for (int j = 0; j < 4; j++){
13            printf("arr[%d][%d] = %d\n", i, j, arr[i][j]);
14        }
15    }
16
17    /*
18     Output:
19     arr[0][0] = 1
20     arr[0][1] = 2
21     arr[0][2] = 3
22     arr[0][3] = 4
23     arr[1][0] = 5
24     arr[1][1] = 6
25     arr[1][2] = 7
26     arr[1][3] = 8
27     arr[2][0] = 9
28     arr[2][1] = 10;
29     arr[2][2] = 11;
30     arr[2][3] = 12;
31     */
32
33     int ary [2][3];
34
35     ary[0][0] = 100;
36     ary[0][1] = 200;
37     ary[0][2] = 300;
38     ary[1][0] = 400;
39     ary[1][1] = 500;
40     ary[1][2] = 600;
41
42     for(int i = 0; i < 2; i++){
43         for(int j = 0; j < 3; j++){
44             printf("ary[%d][%d] = %d\n", i, j, ary[i][j]);
45         }
46     }
47
48     /* Output:
49     ary[0][0] = 100
50     ary[0][1] = 200
51     ary[0][2] = 300
```

```
52     ary[1][0] = 400
53     ary[1][1] = 500
54     ary[1][2] = 600
55     */
56
57     return 0;
58 }
```

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 */
```

multiplicationMatrix.c

```
1 #include <stdio.h>
2
3 int main()
4 {
5     int A[50][50], B[50][50], C[50][50], i, j, m1, n1, m2, n2;
6
7     printf("\n\nMultiplication of two Matrices :\n");
8     printf("-----\n");
9     // for A matrix
10    printf("\n\n A Matrix :\n");
11    printf("-----\n");
12    printf("Enter the number of rows of A matrix(between 1 to 50 ): ");
13    scanf("%d", &m1);
14    printf("Enter the number of columns of A matrix (between 1 to 50 ): ");
15    scanf("%d", &n1);
16
17    // for B matrix
18    printf("\n\n B Matrix :\n");
19    printf("-----\n");
20    printf("Enter the number of rows of B matrix(between 1 to 50 ): ");
21    scanf("%d", &m2);
22    printf("Enter the number of columns of B matrix (between 1 to 50 ): ");
23    scanf("%d", &n2);
24
25    if (n1 != m2)
26    {
27        printf("\n Error : Number of column of the A matrix should be same as
28        number of rows of B matrix. \n\n");
29    }
30    else
31    {
32
33        /* Stored values into the array*/
34        printf("\n\n");
35
36        printf("Enter elements of A matrix :\n");
37        printf("-----\n");
38
39        for (i = 0; i < m1; i++) // row
40        {
41            for (j = 0; j < n1; j++) // column
42            {
43                printf("element - [%d],[%d] : ", i, j);
44                scanf("%d", &A[i][j]);
45            }
46
47        printf("\n\n");
48
49        printf("Enter elements of B matrix :\n");
50        printf("-----\n");
51    }
```

```

52     for (i = 0; i < m2; i++)
53     {
54         for (j = 0; j < n2; j++)
55         {
56             printf("element - [%d],[%d] : ", i, j);
57             scanf("%d", &B[i][j]);
58         }
59     }
60     printf("\n A matrix is :\n");
61     printf("-----\n");
62
63     for (i = 0; i < m1; i++)
64     {
65         printf("\n\t");
66         for (j = 0; j < n1; j++)
67             printf("%d\t", A[i][j]);
68     }
69
70     printf("\n\n");
71
72     printf("\n B matrix is :\n");
73     printf("-----\n");
74
75     for (i = 0; i < m2; i++)
76     {
77         printf("\n\t");
78         for (j = 0; j < n2; j++)
79             printf("%d\t", B[i][j]);
80     }
81     /* calculate the multiplication of the matrix */
82
83     printf("\n\n");
84
85     for (i = 0; i < m1; i++)
86     {
87         for (j = 0; j < n2; j++)
88         {
89             // Calculate the result
90             for (int k = 0; k < n1; k++)
91             {
92                 C[i][j] += A[i][k] * B[k][j];
93             }
94         }
95     }
96
97     // output
98     printf("\nThe C matrix is : \n");
99     printf("-----\n");
100
101    for (i = 0; i < m1; i++)
102    {
103        printf("\n\t");
104        for (j = 0; j < n2; j++)
105            printf("%d\t", C[i][j]);

```

```
106         }
107
108         printf("\n\n");
109     }
110 }
111
112
113 /*
114 Output:
115 Multiplication of two Matrices :
116 -----
117
118
119 A Matrix :
120 -----
121 Enter the number of rows of A matrix(between 1 to 50 ): 2
122 Enter the number of columns of A matrix (between 1 to 50 ): 3
123
124
125 B Matrix :
126 -----
127 Enter the number of rows of B matrix(between 1 to 50 ): 3
128 Enter the number of columns of B matrix (between 1 to 50 ): 2
129
130
131 Enter elements of A matrix :
132 -----
133 element - [0],[0] : 1
134 element - [0],[1] : 3
135 element - [0],[2] : 4
136 element - [1],[0] : 2
137 element - [1],[1] : 1
138 element - [1],[2] : 6
139
140
141 Enter elements of B matrix :
142 -----
143 element - [0],[0] : 4
144 element - [0],[1] : 6
145 element - [1],[0] : 8
146 element - [1],[1] : 7
147 element - [2],[0] : 9
148 element - [2],[1] : 4
149
150 A matrix is :
151 -----
152
153      1      3      4
154      2      1      6
155
156
157 B matrix is :
158 -----
159
```

```
160      4      6
161      8      7
162      9      4
163
164
165 The C matrix is :
166 -----
167
168      64      43
169      70      43
170 */
```

transposeMatrix.c

```
1 #include <stdio.h>
2
3 int main(){
4
5     int rows, cols;
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 matrix
13    int matrix[rows][cols];
14    int transposedMatrix[cols][rows];
15
16    // Get matrix elements from the user
17    printf("Enter the elements of the matrix:\n");
18    for (int i = 0; i < rows; i++) {
19        for (int j = 0; j < cols; j++) {
20            printf("Element [%d][%d]: ", i, j);
21            scanf("%d", &matrix[i][j]);
22        }
23    }
24    // print original matrix
25    printf("Original matrix:\n");
26    for (int i = 0; i < rows; i++) {
27        for (int j = 0; j < cols; j++) {
28            printf("%d ", matrix[i][j]);
29        }
30        printf("\n");
31    }
32
33    // Transpose the matrix
34    for (int i = 0; i < cols; i++) {
35        for (int j = 0; j < rows; j++) {
36            transposedMatrix[i][j] = matrix[j][i];
37        }
38    }
39
40    // Display the transposed matrix
41    printf("Transposed matrix:\n");
42    for (int i = 0; i < cols; i++) {
43        for (int j = 0; j < rows; j++) {
44            printf("%d ", transposedMatrix[i][j]);
45        }
46        printf("\n");
47    }
48
49    return 0;
50 }
```

transposeSquareMatrix.c

```
1 #include <stdio.h>
2
3 int main(){
4     int rows, cols;
5     // Get dimensions from the user
6     printf("Enter the number of rows/cols: ");
7     scanf("%d", &rows);
8     cols = rows;
9
10    // Declare matrix
11    int matrix[rows][cols];
12    int transposedMatrix[cols][rows];
13
14    // Get matrix elements from the user
15    printf("Enter the elements of the matrix:\n");
16    for (int i = 0; i < rows; i++) {
17        for (int j = 0; j < cols; j++) {
18            printf("Element [%d][%d]: ", i, j);
19            scanf("%d", &matrix[i][j]);
20        }
21    }
22    // print original matrix
23    printf("Original matrix:\n");
24    for (int i = 0; i < rows; i++) {
25        for (int j = 0; j < cols; j++) {
26            printf("%d ", matrix[i][j]);
27        }
28        printf("\n");
29    }
30
31    // Transpose of square matrix
32    for (int i = 0; i < rows; i++) {
33        for (int j = i; j < cols; j++) {
34            int temp = matrix[i][j];
35            matrix[i][j] = matrix[j][i];
36            matrix[j][i] = temp;
37        }
38    }
39
40    // Display the transposed matrix
41    printf("Transposed matrix:\n");
42    for (int i = 0; i < rows; i++) {
43        for (int j = 0; j < cols; j++) {
44            printf("%d ", matrix[i][j]);
45        }
46        printf("\n");
47    }
48
49    return 0;
50 }
```