

```
1  # include <iostream>
2  # include <stdio.h>
3  using namespace std;
4
5  // Steps of the program:
6  // Input Matrix A
7  // Input Matrix B
8  // Display Matrices
9  // Pad the product array
10 // Apply the product
11 // Display C
12
13 int main()
14 {
15     // defining variables and arrays
16     float A[10][10], B[10][10], product[10][10], transA[10][10];
17     int Arows, Acols, Brows, Bcols;
18     int i, j, k, test;
19     float sum = 0;
20
21     // Input Rows and Columns of Matrix A
22     cout << "Enter number of rows for matrix A: ";
23     cin >> Arows;
24     cout << "Enter number of columns for matrix A: ";
25     cin >> Acols;
26     // Input Rows and Columns of Matrix B
27     cout << "Enter number of rows for matrix B: ";
28     cin >> Brows;
29     cout << "Enter number of columns for matrix B: ";
30     cin >> Bcols;
31
32     // Inputting elements into Matrix A
33     cout << endl << "Enter elements of matrix A:" << endl;
34     for (i = 0; i < Arows; i++) // Traversing through rows of A
35     {
36         for (j = 0; j < Acols; j++) // Traversing through columns of A
37         {
38             cout << "Enter element a" << i + 1 << j + 1 << " : "; //
39             // Displaying individual element's index
40             cin >> A[i][j]; // input element values
41         }
42     }
43
44     // Inputting elements into Matrix B
45     cout << endl << "Enter elements of matrix B:" << endl;
46     for (i = 0; i < Brows; i++) // Traversing through rows of B
47     {
48         for (j = 0; j < Bcols; j++) // Traversing through columns of B
49         {
50             cout << "Enter element b" << i + 1 << j + 1 << " : "; //
51             // Displaying individual element's index
52             cin >> B[i][j]; // input element values
53         }
54     }
55
56     // Displaying Matrix A to act as a sanity check
```

```
55     cout << endl << "Entered Matrix A is:" << endl;
56     for (i = 0; i < Arows; i++)        // Traversing through rows of A
57     {
58         for (j = 0; j < Acols; j++)    // Traversing through columns of A
59         {
60             cout << A[i][j] << " ";    // output element values
61         }
62         cout << endl << endl;
63     }
64
65     // Displaying Matrix B
66     cout << "Entered Matrix B is:" << endl;
67     for (i = 0; i < Brows; i++)
68     {
69         for (j = 0; j < Bcols; j++)    // Traversing through rows of A
70         {
71             cout << B[i][j] << " ";    // Traversing through columns of A
72         }
73         cout << endl << endl;          // output element values
74     }
75
76     // Intializing and padding the product matrix to 0 with size of Arows x Bcols
77     for (i = 0; i < Arows; i++)        // Traversing through rows of A
78     {
79         for (j = 0; j < Bcols; j++)    // Traversing through columns of B
80         {
81             product[i][j] = 0;        // initialising to 0
82         }
83     }
84
85     // Multiplying matrix A and B and storing in the product array.
86     for (i = 0; i < Arows; i++)
87     {
88         for (j = 0; j < Bcols; j++)
89             for (k = 0; k < Brows; k++)    // Can also take Acols = Brows,
89                 // allows to simulateanously traverse A and B
90             {
91                 // Accessing elements from A and B matrices and multiplying
92                 // them
92                 product[i][j] += A[i][k] * B[k][j];    // i = row number of
92                 // A, j = column number of B
93             }
94     }
95
96     // Displaying the multiplication of two matrices A and B.
97     cout << endl << "Matrix C= " << endl;
98     for (i = 0; i < Arows; i++)        // traversing the size of the array
99     {
100         for (j = 0; j < Bcols; j++)
101         {
102             cout << " " << product[i][j];    // outputting the product
102             // array elements
103             if (j == Bcols - 1)    // conditions to stop
104                 cout << endl;
105         }
    }
```

```
106     }  
107  
108     cout << endl << "Program complete, press Enter to stop";  
109     cin >> test;  
110  
111     return 0;  
112 }  
113
```