# Two-dimensional Array Algorithms

**Program 7-21**  
  
 **1  // This program demonstrates a two-dimensional array.**  
 **2  #include <iostream>**  
 **3  #include <iomanip>**  
 **4  using namespace std;**  
 **5**  
 **6  int main()**  
 **7  {**  
 **8      const int NUM\_DIVS = 3;           // Number of divisions**  
 **9      const int NUM\_QTRS = 4;           // Number of quarters**  
**10      double sales[NUM\_DIVS][NUM\_QTRS]; // Array with 3 rows and 4 columns.**  
**11      double totalSales = 0;            // To hold the total sales.**  
**12      int div, qtr;                     // Loop counters.**  
**13**  
**14      cout << "This program will calculate the total sales of\n";**  
**15      cout << "all the company's divisions.\n";**  
**16      cout << "Enter the following sales information:\n\n";**  
**17**  
**18      // Nested loops to fill the array with quarterly**  
**19      // sales figures for each division.**  
**20      for (div = 0; div < NUM\_DIVS; div++)**  
**21      {**  
**22          for (qtr = 0; qtr < NUM\_QTRS; qtr++)**  
**23          {**  
**24              cout << "Division " << (div + 1);**  
**25              cout << ", Quarter " << (qtr + 1) << ": $";**  
**26              cin >> sales[div][qtr];**  
**27          }**  
**28          cout << endl; // Print blank line.**  
**29      }**  
**30**  
**31      // Nested loops used to add all the elements.**  
**32      for (div = 0; div < NUM\_DIVS; div++)**  
**33      {**  
**34          for (qtr = 0; qtr < NUM\_QTRS; qtr++)**  
**35              totalSales += sales[div][qtr];**  
**36      }**  
**37**  
**38      cout << fixed << showpoint << setprecision(2);**  
**39      cout << "The total sales for the company are: $";**  
**40      cout << totalSales << endl;**  
**41      return 0;**  
**42  }**

Processing a 2-Dimensional array will require nested loops, such as summing all the elements of a 2-D Array:

**1 const int NUM\_ROWS = 5; // Number of rows**  
**2 const int NUM\_COLS = 5; // Number of columns**  
**3 int total = 0; // Accumulator**  
**4 int numbers[NUM\_ROWS][NUM\_COLS] = {{2, 7, 9, 6, 4},**  
**5                                    {6, 1, 8, 9, 4},**  
**6                                    {4, 3, 7, 2, 9},**  
**7                                    {9, 9, 0, 3, 1},**  
**8                                    {6, 2, 7, 4, 1}};**  
**9 // Sum the array elements.**  
**10 for (int row = 0; row < NUM\_ROWS; row++)**  
**11 {**  
**12     for (int col = 0; col < NUM\_COLS; col++)**  
**13          total += numbers[row][col];**  
**14 }**  
**15 // Display the sum.**  
**16 cout << "The total is " << total << endl;**Other algorithms for you to code:

* Summing the rows
* Summing the columns