2D- Arrays

Problem 1: Write a function called *dropDimension* that copies the entries from a 2-dimensional array row by row as the entries of a 1-dimensional array. Assume that the 1-dimensional array has more than enough capacity for these entries. (The function should use capacities of the 2-dimensional array but not the 1-dimensional array as input parameters.)
For example, a program that uses the function follows.

```
int main() {
     int x[100];
     int y[2][3] = {{3,1,4}, {1,5,9}};
     int yrows = 2, ycols = 3;
     dropDimension(y, yrows, ycols, x);
     for (int i = 0; i <= 5; i++) cout << x[i];
       // 314159 is printed
     cout << endl;
     return 0;
}
```

Problem 2: Write a C++ function called range that returns the difference between the largest and smallest elements in a 2-dimensional array (with 3 columns).It should be possible to use your function in the following program. (The output from this program is 10 because the difference between the largest element 13 and the smallest element 3 is: 13-3 = 10)

```
main() {
     int data[2][3] = {{11, 12, 11}, {3, 12, 13}};
     int x;
     x = range (data, 2, 3);
     // data is the 2-d array, 2 and 3 are its capacities
     cout << "The range is: " << x << endl
     return 0;
}
```

Problem 3: Write a C++ function called numEven that returns the number of even elements in a 2-dimensional array (with 3 columns). It should be possible to use your function in the following program. (The output from this program is 2 because only the two 12s are even).

```
main() {
     int data[2][3] = {{15, 12, 19}, {3, 12, 13}};
     int x;
     x = numEven (data, 2, 3);
          // data is the 2-d array, 2 and 3 are its capacities
     cout << "The number of evens is: " << x << endl;
}
```

Problem 4: Write a C++ function called squares that replaces each element of a 2-dimensional array (with two columns) by its square. It should be possible to use your function in the following program.

```
main() {
     int data[2][2] = {{1, 2}, {3, 4}};
     squares (data, 2, 2);
     for (int i = 0; i < 2; i++)
          cout << data[1][i] << " "; // prints 9 16
}
```