**Worksheet 15.1 – Answer Sheet**

### Magic Square

Solutions:

**int** sumRow(**int**[][]square, **int** row){

// Precondition: square is an initialized matrix, MAX rows x MAX columns

// 0 <= row < MAX

// Postcondition: returns the sum of the values in row

**int** sum = 0, col;

**for** (col=0; col<MAX; col++){

sum += square[row][col];

}

**return** sum;

}

**boolean** unique(**int**[][]square){

// Precondition: square is initialized with integers.

// Action: Inspects every value in square, checking that each one is a unique

// integer ranging from 1..MAX\*MAX

// Postcondition: Returns true if each value is unique from 1..MAX\*MAX,

// otherwise returns false

**boolean**[] list = new **boolean**[MAX\*MAX + 1];

**int** row, col, loop, value;

**for** (row=0; row < MAX; row++){

**for** (col=0; col < MAX; col++){

value = square[row][col];

**if** (value <= MAX \* MAX)

list[value] = true;

}

**for** (loop=1; loop<=MAX\*MAX; loop++){

**if** (false == list[loop])

**return** false; // if we ever hit a false, function returns false

}

**return** true; // if we get here, all values are unique

}

}