

Strings, Grids, Vectors: Solutions

1. Mirror (Grid)

Regular:

```
void mirror(Grid<int>& grid) {
    for (int r = 0;
         r < grid.numRows(); r++) {
        // start at r+1 rather than 0
        // to avoid double-swapping
        for (int c = r + 1;
             c < grid.numCols(); c++) {
            int temp = grid[r][c];
            grid[r][c] = grid[c][r];
            grid[c][r] = temp;
        }
    }
}
```

Bonus:

```
void mirror(Grid<int>& grid) {
    Grid<int> result(grid.numCols(),
                    grid.numRows());
    for (int r = 0;
         r < grid.numRows(); r++) {
        for (int c = 0;
             c < grid.numCols(); c++) {
            result[r][c] = grid[c][r];
        }
    }
    grid = result;
}
```

2. Grid Mystery! (Grid)

$\{\{1, 2, 3\}, \{2, 4, 6\}, \{3, 6, 9\}, \{4, 8, 12\}\}$

3. CrossSum (Grid)

```
int crossSum(Grid<int>& grid, int row, int col) {
    int sum = 0;
    for (int c = 0; c < grid.numCols(); c++) {
        sum += grid[row][c];
    }
    for (int r = 0; r < grid.numRows(); r++) {
        sum += grid[r][col];
    }
    sum -= grid[row][col]; // subtract center because it was added twice
    return sum;
}
```

5. Collection Mystery! (Vector)

{0, 10, 20, 10}

{0, 0, 8, 9, -1, 55}

{0, 0, 0, 0, 16, 9, 64, 25, 0}

6. Switch Pairs (Vector)

```
void switchPairs(Vector<int>& v) {
    for (int i = 0; i < v.size() - 1; i += 2) {
        int first = v[i];
        int second = v[i + 1];
        v[i + 1] = first;
        v[i] = second;
    }
}
```