SOLUTIONS:

- Base program (**Direct-mapped** 2KB D$):
  - D$ Accesses: 1025
  - D$ Misses: 1024
  - Cycles: 12837
  - Instructions: 8719

  Discussion: Given the sizes and memory mapping of the two arrays, each new element brought to the D$ for an array will replace the latest block brought for the other array, and as a result each read operation will generate a D$ miss.

- Base program (**2-way** set-associative 2KB D$):
  - D$ Accesses: 1025
  - D$ Misses: 257
  - Cycles: 9777
  - Instructions: 8719

  Discussion: In this configuration all conflict misses are avoided, thus reducing the misses to ¼ the original number of misses.

- Array Enlargement (**Direct-mapped** 2KB D$):
  - D$ Accesses: 1025
  - D$ Misses: 259
  - Cycles: 9777
  - Instructions: 8719

  Discussion: The amount of misses gets reduced to ¼ the original number of misses. All conflict misses are removed, and only compulsory misses remain. All hits are a consequence of spatial locality exploitation. As for the amount of cycles, it gets significantly reduced with the optimization.
  We can easily deduct the average D$ miss penalty with the next operation: (12837-9777)/(1024-259) = 4 cycles, which, as we determined in Exercise 2, is correct.

- Array Merging (**Direct-mapped** 2KB D$):
  - D$ Accesses: 1025
  - D$ Misses: 259
  - Cycles: 9769
  - Instructions: 8718

Discussion: The amount of misses gets reduced to ¼ the original number of misses. All conflict misses are removed, and only compulsory misses remain. All hits are a consequence of spatial locality exploitation.