**SOLUTIONS:**

* Base program (direct-mapped 2KB D$):
  + D$ Accesses: 8362
  + D$ Misses: 1033
  + Cycles: 29387
  + Instructions: 19918

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 (direct-mapped 4KB D$):
  + D$ Accesses: 8358
  + D$ Misses: 264
  + Cycles: 26267
  + Instructions: 19918

Discussion: All conflict misses are avoided, thus reducing the misses to ¼ the original number of misses. The number of cycles also reduces.

* Loop interchange (direct-mapped 2KB D$):
  + D$ Accesses: 8362
  + D$ Misses: 271
  + Cycles: 26339
  + Instructions: 19918

Discussion: The amount of misses is reduced to around ¼ the original number of misses, which makes sense, as the optimization removes all D$ conflict misses. All hits are a consequence of spatial locality exploitation.

As for the amount of cycles, we can easily deduct the average miss penalty with the next operation: (29387-26339)/(1033-271) = 4 cycles, which, as we will see in a future lab, is correct.