The percentage of instructions types remains pretty consistent as the number of integers tested is increased. I believe the consistency is because the instructions are just being performed more times across all instruction types at a mostly even rate (exceptions with jump because those are only called a set number of times throughout the program). As the number of integers tested increased, so did the percentage of correct branch predictions. There was more data to go off of to help predict whether branches would be taken because of the increased size of the test case. Overall, the cache hit ratio got worse with more integers. The program likely had more thrashing as the test case size increased. For associative cache (N-way set associative and fully associative), the increased test size could have caused more cache lines to have memory blocks mapped to them. For direct mapping, with an increased amount of memory used, there was likely an increased chance that the blocks of main memory had to be mapped to the same cache lines.