Questions:

#1: Which replacement strategy would you choose and why? You should consider both the end results and the effort it took to implement each strategy. Discuss what your results show about the relative merits of FIFO, LRU, LFU, and OPT for the different combinations of parameters.

Considering the difficulty of code and the benefits of each memory management algorithm, I would choose the LRU algorithm. Besides the optimum algorithm, which was much harder to implement and runs much more slowly, LRU consistently got the best results. The FIFO and LRU algorithms seem to perform about the same, but LRU seems like a more practical algorithm, but FIFO is easier to implement. LFU consistently performs the worst. Optimal obviously performs the best, but it is difficult to implement and obviously not possible because you cannot predict future page accesses with 100% accuracy.

#2: What aspect of memory management did you find most difficult to implement?

The optimum algorithm was the most difficult to implement because it had to scan forward in the reference string and not just access the previous columns of the frames array.

#3: What aspect of memory management did you find least difficult to implement?

The FIFO algorithm was the easiest to implement. All that was required was to keep track of the index of the input array that each frame was filled during.

#4: What, if anything, would you change in your current design?

The only change that I would immediately make would be to add functions for scanning the input array and the frame array. This would only function to clean up the code and increase readability.

#5: What, if anything, did you find interesting or surprising about page replacement algorithms that you did not know before doing this project?

The only surprise was how poorly LFU performed. Also, optimal was not that much better, maybe not worth the programming price because of the locality of reference.