For this assignment, we will be simulating using two different page replacement algorithms and evaluating the performance of each. These two algorithms will be PFF and VSWS. For this you will simulate a system where the OS has available an infinite amount of physical memory. However, you don’t want to over allocate (is that even possible with infinite memory???) memory to processes so you will do demand paging with the processes in the system.

A filename is given to your program on the command line. The file contains integers only, one per line, how many is up to you but a large number is a good idea (10,000?). The first integer is an indication to you how many pages the process occupies; this will not change throughout the run of the program. All subsequent integers are page references which the program makes during it execution.

In the PFF algorithm, we decide if a new frame should be allocated based on the time since the last page fault. For this, you can use a number of page accesses (remember those integers in the file????). You must decide what a reasonable F value is. Try a few different values. Record the total number of page faults across the run of the program. Put a short comment in your code about how F impacts the total number of page faults.

In the VSWS algorithm, we have M,L and Q to help us decide when to run the algorithm. Determine some reasonable numbers for these values and put a comment in your code to indicate how changes impact the total number of page faults.

Now that both algorithms are done, run the two algorithms on the same data set and determine the performance of each with careful consideration as to the minimum number of frames allocated (probably just 1 when you started, but how often do you have less than 10 frames in memory), the maximum number of frames (the largest amount of memory used), and the total number of page faults.

Submit your SINGLE program code file (please no rar, 7z or other compression, zip only if you absolutely must) in the programming language of your choice on time!