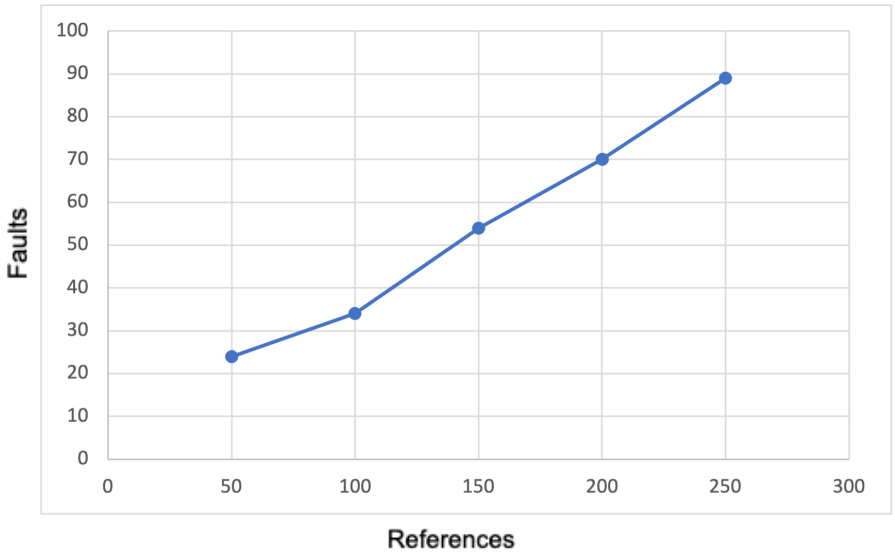
What do you expect to see from the comparison of the different strategies and how you perform this comparison?

Before running the program, I would expect that the LRU (least recently used) strategy would be more efficient in comparison to the FIFO (First in first out). The FIFO strategy will keep what was most recently added, meanwhile LRU is more likely to keep the frequently used items. More often than not there will be items in memory that are generally added once and never used again, so the LRU would take care of that better in comparison to FIFO.

Measure and graph the number of page faults for the following scenarios:

FIFO

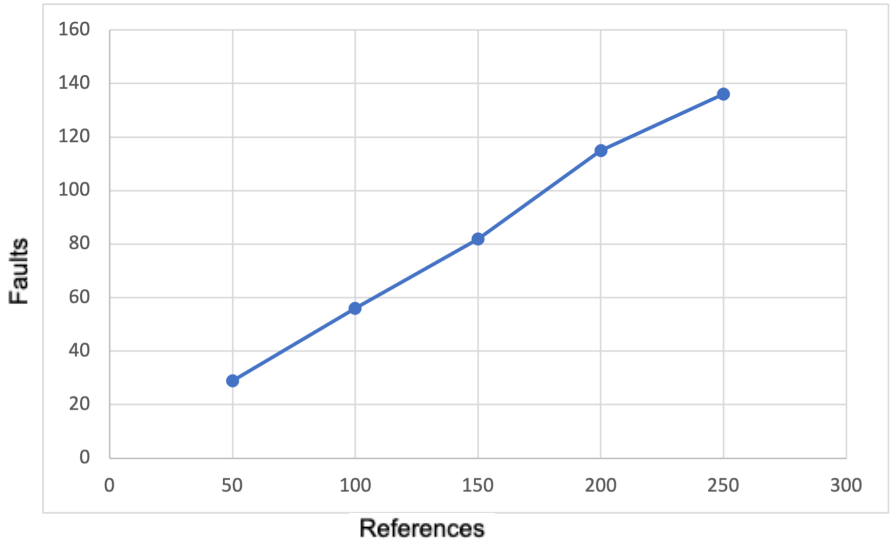
npages = 100, nframes = 50, nrefs = [50..250] in steps of 50



nrefs	faults
50	24
100	34
150	54
200	70
250	89

FIFO

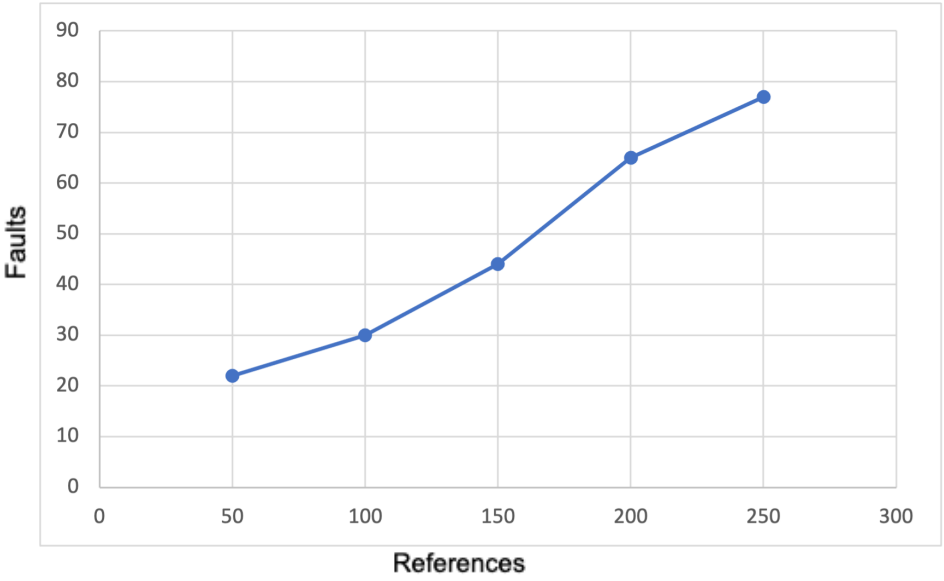
npages = 100, nframes = 10, nrefs = [50..250] in steps of 50



nrefs	faults
50	29
100	56
150	82
200	115
250	136

LRU

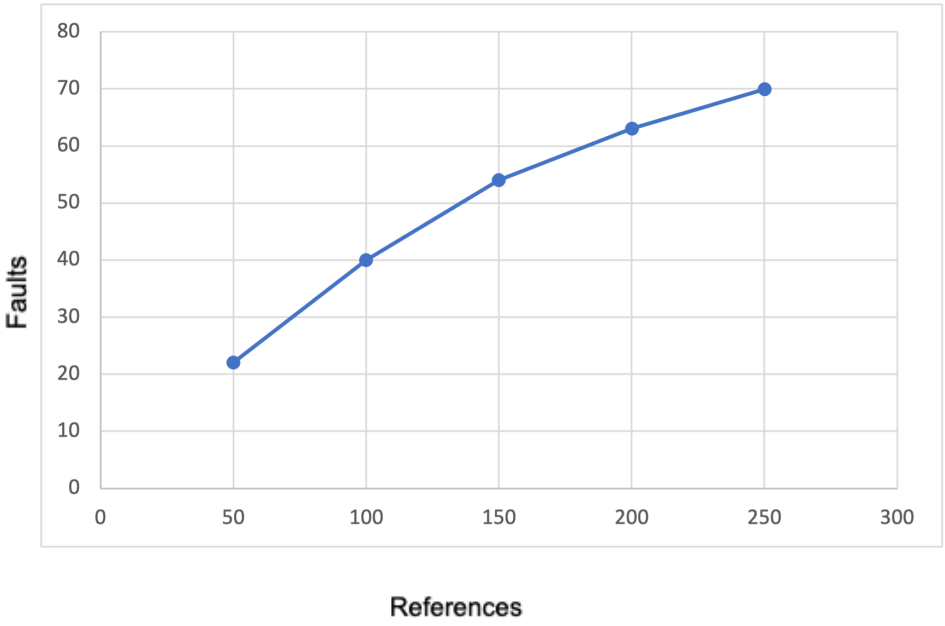
npages = 100, nframes = 50, nrefs = [50..250] in steps of 50



nrefs	faults
50	22
100	30
150	44
200	65
250	77

LRU

npages = 100, nframes = 10, nrefs = [50..250] in steps of 50



nrefs	faults
50	22
100	40
150	54
200	63
250	70

Explain the nature of the results. If one algorithm performs better than another under certain conditions, then point that out, explain the conditions, and explain why it performs better.

As was previously inferred the LRU strategy performed better than the FIFO. This was concluded through analyzing the test results and seeing that the LRU performed with the least amount of faults, it can be assumed as the better strategy. The FIFO performs worse because as the frames increase so does the faults, this is also known as the Belady's anomaly. The LRU takes advantage of the efficiency of picking the page that has faulted and hasn't been used in a while.