1. Tabulates the search results' quality in terms of P@5, P@20, Recall@5,
Recall@20, MRR (recip_rank) and MAP for each of the 5 runs:

products/query 1/result_q1_p1.txt"
map all 0.5000
recip_rank all 1.0000
P5 all 0.2000
P20 all 0.0500

products/query 1/result_q1_p2.txt"
map all 0.5000
recip_rank all 1.0000
P5 all 0.2000
P20 all 0.0500

products/query 1/result_q1_p3.txt"
map all 0.5000
recip_rank all 1.0000
P5 all 0.2000
P20 all 0.0500

products/query 1/result_q1_p4.txt"
map all 0.5000
recip_rank all 1.0000
P5 all 0.2000
P20 all 0.0500

products/query 1/result_q1_p5.txt"
map all 0.5000
recip_rank all 1.0000
P5 all 0.2000
P20 all 0.0500

Query 2:
products/query 2/result_q2_p0.txt"
map all 0.3000
recip_rank all 0.5000
P5 all 0.4000
P20 all 0.1000

products/query 2/result_q2_p1.txt"
map all 0.3000
recip_rank all 0.5000
P5 all 0.4000
P20 all 0.1000

products/query 2/result_q2_p2.txt"
map all 0.3000
recip_rank all 0.5000
P5 all 0.4000
P20 all 0.1000

products/query 2/result_q2_p3.txt"
map all 0.3000
recip_rank all 0.5000
P5 all 0.4000
P20 all 0.1000

products/query 2/result_q2_p4.txt"
map all 0.3000
recip_rank all 0.5000
P5 all 0.4000
P20 all 0.1000

products/query 2/result_q2_p5.txt"
map all 0.3000
recip_rank all 0.5000
P5 all 0.4000
P20 all 0.1000

Query 3:
The term speed was not in the document collection.

2. Summarizes your observations about PRF and its parameter x:
For my tests the PRF did not change with parameter x. The documents that were retrieved were the same, only the score changed as x increased. This was unexpected. I suspect that my implementation might have a bug. I regret the amount of complexity I introduced with homework 1, which made the project a little unwieldy.

3. Report all the kappa statistic scores that you computed in Part 1:

beasely-qrels.txt: 0.7
Chuang-qrels.txt: 0.729381443299
kkatipally-qrels.txt: 0.614285714286
ng-qrels.txt: 0.749652294854
saylor-qrels-2.txt: 0.785714285714