Hashing function report by Tejas Raval(tr7550@rit.edu)

I have written 2 Hash Function. 1st one is the function def hash_func(self,key) and the second one is def hash_func2(self,key,divBy)

Below are the results of the testing done on Book3 and Book1(from Guttenberg.com), tejas.txt(my own testing file) and words_dict.txt file in tabular format. For testing, the hash functions (2 written by me and 1 Python's hash function) are calculated for load factor of 0.5, 0.6 and 0.7

Book3 Load factor 0.5		
	Collisions	Probes
Function_1	60827	94830
Function_2	62742	148017
Pyhton's hash	59984	91032

Book3 Load factor 0.6		
	Collisions	Probes
Function_1	62043	104663
Function_2	64232	186398
Pyhton's hash	61377	100749

Book3 Load factor 0.7		
	Collisions	Probes
Function_1	63488	120012
Function_2	65875	258896
Pyhton's hash	62654	112441

Book1 Load factor 0.5		
	Collisions	Probes
Function_1	231148	297991
Function_2	234487	422606
Pyhton's hash	229892	294201
Book1 Load factor 0.6		
	Collisions	Probes
Function_1	233620	320683
Function_2	237408	502621
Pyhton's hash	232092	310045
Book1 Load factor 0.7		
	Collisions	Probes
Function_1	233267	330154
Function_2	236037	603168
Pyhton's hash	232212	320632

Words_dict.txt Load factor 0.5		
	Collisions	Probes
Function_1	72422	336293
Function_2	108013	1406087
Pyhton's hash	66714	314125

Words_dict.txt Load factor 0.6		
	Collisions	Probes
Function_1	82409	412753
Function_2	120488	1730682
Pyhton's hash	75288	378451
Words_dict.txt Load factor 0.7		
	Collisions	Probes
Function_1	93969	531693
Function_2	134099	2291732
Pyhton's hash	85785	482810

tejas.txt Load factor 0.5		
	Collisions	Probes
Function_1	24	122
Function_2	20	91
Pyhton's hash	15	98
tejas.txt Load factor 0.6		
	Collisions	Probes
Function_1	27	140
Function_2	22	101
Pyhton's hash	27	106
tejas.txt Load factor 0.7		
	Collisions	Probes
Function_1	31	166
Function_2	26	112
Pyhton's hash	27	153

From above its clear that Python’s hash function gives minimum collisions followed by function1 and function2