Section 6: And now some questions

Answer to 1.

The approach of separate chaining solves the problem of primary and secondary clustering which are the resulting drawbacks of the linear and quadratic probing. In this approach the problem of high load factors and rehashing does not pose much of a challenge especially when the table size chosen is sufficiently large unlike in the other two mentioned collision algorithm approaches.
This approach is space efficient and uses a combination of an array and linked lists hence proves to be more efficient and preferable one over the other two.

Answer to 2.

No, it would not work better. This would lead to a large number drawbacks that results from these three algorithms. These drawbacks would include long linked lists being created, and this would result in poor retrievals and insertions of items from the hash table. Clusterings that result from the linear and quadratic probing properties will also pose a poor performance and expensive retrievals and insertions. There will be a need for this hybrid algorithm to handle rehashing, which is expensive.

Answer to 3.

Collisions happen when the keys hash to the same index on the table. With separate chaining, when there's collision, the keys are put on same position on the hash table, one after the other(forming an array inside the hash table, on the hash index). So the number of collisions will be one, since it only checks one position on the table. This can be checked using the "findKey(X)" of MyHashTable.

Answer to 4.

No. This problem of rehashing is handled by the use of the combination of an array and a linked list with a large enough size of the table that result in short linked lists. The hash table is never needed to be rehashed as its design combines a data structure that is dynamic and non-dynamic.