**Purpose:**

The purpose of this project was to compare the two different collision resolution techniques used in hash tables, open address and chaining. To do this a sufficient data set was supplied to see how the two methods work and what is different about them. The data given was from Major League Baseball, all of the player information from the year 1985 all the way through 2016. This is a hefty data set because of the thousand of baseball players that have played in those 31 years. Since each algorithm deals with the data differently, each will have a different amount of collisions while making the hash tables. Because of the sheer amount of data sets, it is possible to get a decent assessment of how each different collision resolution method performs, and the benefits and drawbacks of each different technique.

**Procedure**:

Before anything could actually happen, each technique needed to be understood for how it worked and what actually happened when a collision occurred.

The open addressing method is also known as closed hashing, and when a collision happens, the algorithm sorts the data into an open address in the hash table, as the name implies. This sorting is called probing, and a probing sequence can take place in a few different ways, linearly, quadratic, and double. Linear probing is where the search for an open address progresses at a constant search interval, putting the data into the closest open address to the index value. Quadratic probing is when the distance between probes increases with a constant at each step. Double hashing is definitely the most complex, as when a collision occurs a second hash function is brought into use to determine where the data is to be sorted in the hash table.

The chaining method is a collision resolution technique that is when the hash table values point to singly linked lists containing key-value pairs with the same hash, new key-value pairs are added to the end of the list. The resulting lists are then searched for the appropriate key. This means that not every address in the hash table will be utilized and can remain empty, or rather NULL.

Because of the differences in these algorithms, they have there own benefits and drawbacks. Open addressing does not use as much memory space as chaining does because of the lack of storage of linked lists, but is generally accepted to be slower because of the time it takes more time to place the data into open address slots rather than just add onto/create a list. Chaining can be slow if the linked lists are too long, but it is generally faster. It also has more of an impact on memory usage, as the linked lists require more memory space allocation and with the possibility of having empty spots in the hash table memory space isn’t used as efficiently as in the open addressing method.

Now that the two algorithms are better understood it is possible to implement them with the given data.

**Data:**

The data set provided was quite immense, having over 26,000 data sets. Each set has thirteen sections, these sections being the year, team ID, league ID, player ID, salary, first name, last name, birth year ,birth country, weight, height, bats, and throws. The year is in the range of 1985 through 2016, the team ID is the three letter abbreviation of the team name, the league ID is the the two letter abbreviation of which league the team belongs to, either the American League or the National League, the player ID is the specific players identification number, which is important because there are so many people in the data set a few do have the same name but are not to be mixed up in the data. The first and last name are self explanatory, as well as is the birth year, salary, birth country, weight, and height. The bats and throws sections are simply the direction that the specific player hits and throws from, being either right or left. The first and last name are used in each algorithm to generate the hash table key. This led to the chaining algorithm producing a hash table with linked lists and the open addressing algorithm needing to increment through the table to place every data set in an empty slot.

**Results**:

As table size increased search operations went down - open

As table size ‘’ ‘’ ‘’ ‘’ ‘’ - chain

Both algorithms start with a hash table size of 5147. This size was increased to see how it would change the results. For chaining as the size increased the number of collisions went up and down, indicating that the size of the table is arbitrary, as long as there is enough space to call the minimum. For the open addressing as the size increased the number of collisions went down and up.