Text

Description automatically generated

Text

Description automatically generated

At first, I thought that separate chaining hash tables would be better as it can be used to group similar elements, for example employees with the same name or experience level, according to how the hashing is done. But after remembering the differences between the structure of an array and that of a linked list, and doing some research to test my hypothesis, I concluded that linear probing is faster than separate chaining when it comes to searching, since the addresses used for an array are closer together than those of a linked list. This would make the linear probing hash more efficient when storing a large amount of data.

I hashed by summing the ASCII values of the name of each employee and dividing it by 10 (the number of employees + 1) using the mod function %. The key generated was the remainder of that operation. I used this function to try and get a variety of keys/ avoid collisions as much as possible, as the sum of ASCII values is different for each name. However, it is difficult to test whether this theory is true or not with this number of employees (much more employees are needed to create a larger hash table and understand whether or not using the ASCII sum of the string would result in a relatively low rate of collision compared to age, salary or experience).