

Graphical user interface, application

Description automatically generated

Each employee is stored in an index according to the first letter of his name, where A is 0, B is 1, C is 2,…. . The program prints the index of the array along with the employee stored in this index. First, hash 1, which is the dynamic array, is displayed. If the index is not empty (more than 1 employee have a name with the same first letter), the data is put in the next empty index found, and the number of collisions is incremented. For example, Abdallah was supposed to be in index 0, but index 0 contains Ayman, so we move to the next index, but it contains Aya, so Abdallah is placed in the next empty index, which is 3. Then, hash 2, which is the linked list, is displayed. If the index is not empty (more than 1 employee have a name with the same first letter), the data is put in another node of the same linked list in the same index, and the number of collisions is incremented.

I chose this hash function because I think that it is efficient to store employees based on the names. Also, using ascii codes for the first letter is simple to implement.

I believe that using the linked list (separate chaining) is better as it has a lower collision rate. Also, each employee would be found easily as it is stored in its correct index, unlike the array, where the employee could be stored in a different index if its original index is not empty.