

Hands on 10

Explanation:

Here's an implementation of a hash table in C++ that follows the given requirements:

- Uses multiplication and division methods for hashing.
- Supports any hash function through function pointers.
- Uses collision resolution by chaining (doubly linked list).
- Implements dynamic resizing (doubling/halving).
- Uses C-style arrays and a custom doubly linked list (no STL containers).

⇒ Doubly Linked List (DoublyLinkedList):

- ◇ Manages chaining of elements within the hash table.
- ◇ Supports insertion, deletion, and search.

⇒ Hash Function (hash(int key)):

- ◇ Uses the multiplication method with a golden ratio fraction (0.6180339887).

⇒ Resizing Mechanism (resize(int newCapacity)):

- ◇ Doubles the capacity when full (load factor ≥ 1.0).
- ◇ Halves the capacity when 75% empty (load factor ≤ 0.25).
- ◇ Rehashes all elements after resizing.

⇒ Operations:

- ◇ insert(int key, int value): Adds key-value pairs.
- ◇ remove(int key): Deletes a key.
- ◇ get(int key): Retrieves the value.
- ◇ display(): Prints hash table buckets.