KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
227 Nguyễn Văn Cừ, Phường 4, Quận 5, TP.HCM
Điện Thoại: (08) 38.354.266 - Fax:(08) 38.350.096

CẤU TRÚC DỮ LIỆU & GIẢI THUẬT

HDTH - BÀI TẬP TUẦN 9

HKI - 2022 – 2023

Bảng băm – Hash Table

Bài 1 Cài đặt Bảng băm với phương pháp giải quyết đụng độ 'Separate Chaining, sử dụng cấu trúc dữ liệu 'list' của thư viện STL C++. Cấu trúc bảng băm định nghĩa như sau:

```
/* Cau truc Hash Table */
struct Hash {
      int slot;
      // Pointer to an array containing slots
      list<int> *table;
};

/* Ham Khoi Tao Hash Table */
void create_Hash(Hash &h, int b);

/* Ham Hashing */
int hash_func(Hash h, int x);

/* Them Item vao HTable */
void insertItem(Hash& h, int key);

/* Xoa 1 Item */
void deleteItem(Hash h, int key);

// Display hash table
void display_hash(Hash h);
```

Hướng dẫn : Xem các sử dụng cấu trúc 'list' theo sau:

Link: https://www.geeksforgeeks.org/list-cpp-stl/

- Functions used with List:
  .push_back();

  .list::begin(); list::end();

  .erase();

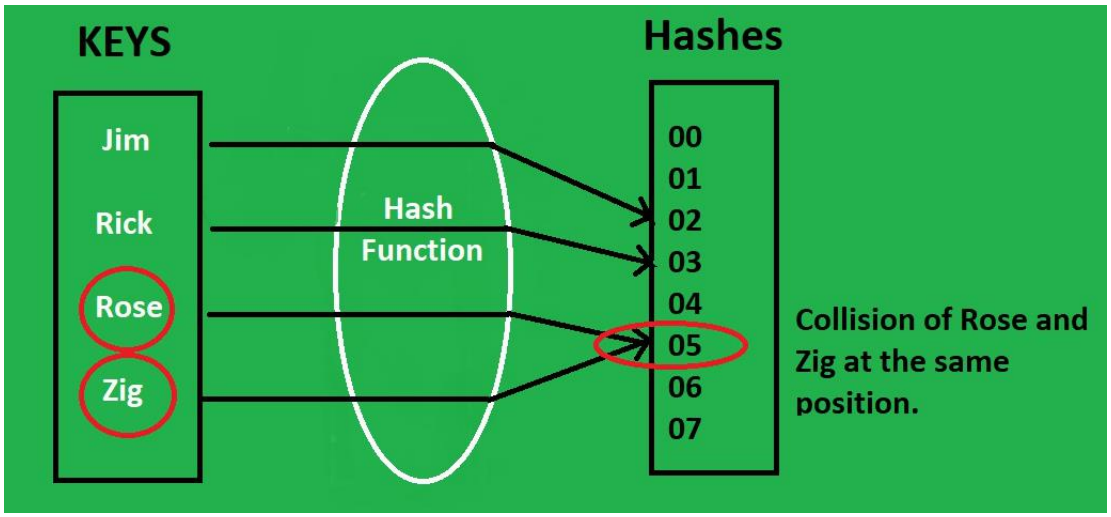
Bài 2 Viết chương trình tạo ứng dụng từ điển sử dụng Bảng băm với phương pháp 'Separate Chaining'.

Hướng dẫn

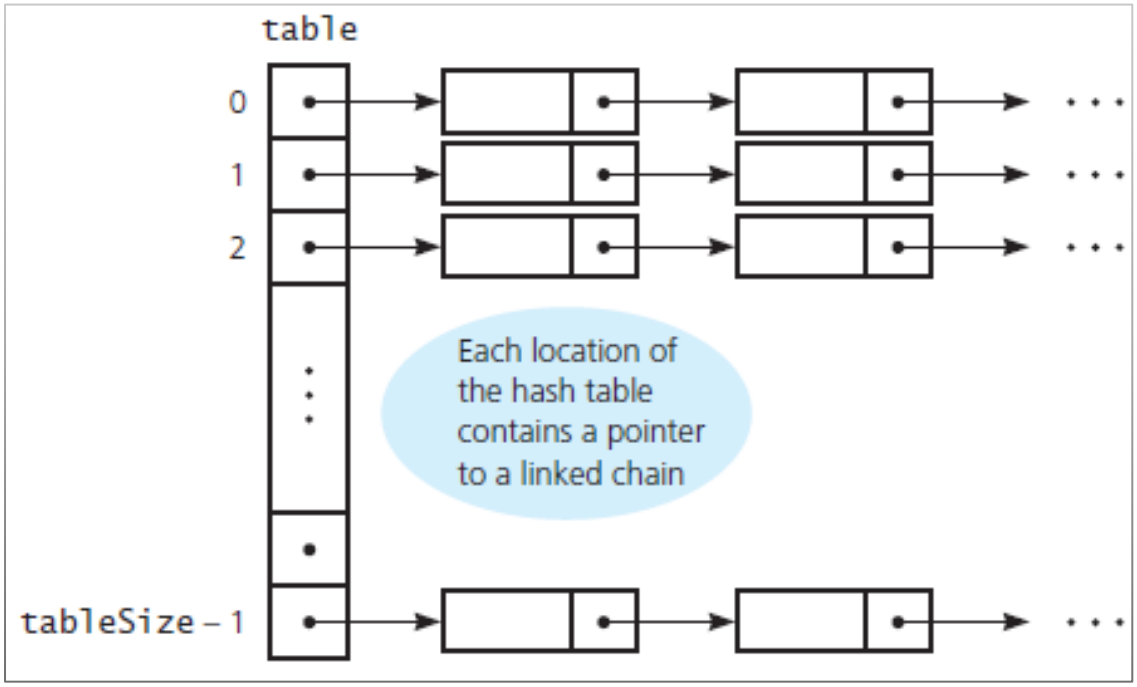
Sinh viên đọc: **Chapter 18 Dictionaries and Their Implementations**, page.525,

- **2014-Data abstraction and Problem Solving with C++.**

☞ **Giải quyết đụng độ**



✍ **Giải quyết đụng độ với phương pháp tiếp cận Separate Chaining**



Đinh nghĩa cấu trúc Bảng Băm hiện thực từ điển:

```
/* Cau truc Bang Bam */
#include<iostream>
using namespace std;
static const int DEFAULT_SIZE = 101;

/* Cau truc node cua Hash Table cai dat tu dien */
template<class KeyType, class ItemType>
struct Node {
      KeyType key;   // search key
      ItemType item; // Data item
      Node<KeyType, ItemType>* nextPtr;
};
/* Cau truc Hash Table */
template<class KeyType, class ItemType>
struct Hash {
      int itemCount;     // Count of dictionary entries
      int hashTableSize; // Table size must be prime
      Node<KeyType, ItemType> **hashTable; // Array of pointers to entries
};

/* Ham khoi tao Bang bam */
template < class KeyType, class ItemType>
void init_Hash(Hash& h);

/* Ham bam */
template < class KeyType, class ItemType>
int hashFunc(Hash h, int x);

/* Ham them 1 item vao bang bam */
template < class KeyType, class ItemType>
void add(Hash& h, const ItemType& newItem, KeyType& searchKey);

/* Ham xoa 1 item tu Hash Table */
template < class KeyType, class ItemType>
void remove(Hash& h, const KeyType& searchKey);

-------------------------------------------------
```