

Bài 3.1

1. Biên dịch và chạy chương trình trên
2. Hãy chỉ ra công thức toán của hàm băm trong đoạn code mẫu trên
=> phép toán lấy dư : $k \% M$
3. Trong hàm khởi tạo bảng băm, chỉ ra tập U có bao nhiêu phần tử, tập các khoá k lưu trong bảng băm có bao nhiêu phần tử.
=> có 5 phần tử, tập khóa k có 5 phần tử.
4. Mô tả quy trình các bước chi tiết từ khi xây dựng bảng băm đến khi xuất ra kết quả tìm kiếm.
=> B1: lấy phần dư các phần tử
B2: Xây dựng hàm băm: phần dư của phần tử là vị trí của phần tử đó trong mảng mới
B3: Hàm main: nhập khóa k và xuất ra giá trị nếu có

Bài 3.2

1. Biên dịch đoạn chương trình trên.
2. Chỉ ra những thay đổi của chương trình này so với chương trình cơ bản.
=> chương trình này dữ liệu là một struct chứ không còn là kiểu dữ liệu thông thường.
3. Nếu bỏ đoạn code sau trong hàm băm

```
if (M == 0)
    return 0;
```

 Thì có được không? Giải thích lý do.
 Không vì không thể chia cho 0.
4. Nếu thay `K[i].key` trong đoạn code sau

```
pos = Hash(K[i].key, M);
```

 Thành

```
pos = Hash(K[i], M);
```

 Thì chuyện gì xảy ra? Giải thích lý do tại sao.

=> chương trình sẽ bị lỗi vì đối số cần truyền vào kiểu int không phải kiểu word.

5. Nếu thay giá trị K[2].key chỗ dòng code sau

```
K[2].key = 5;
```

Thành

```
K[2].key = 13;
```

Và lúc chạy, nhập MSSV là 3. Kết quả xuất ra là bao nhiêu? Đúng hay sai? Nếu sai, giải thích lý do tại sao?

=> đúng vì $13 \% 5 = 3$

6. Viết lại chương trình trên, cho phép người dùng tự nhập số lượng giá trị k và thông tin của SV cần đưa vào bảng băm (ở đoạn code trên tác giả chỉ định cứng số lượng là 5).

```
=> void InitHash(Word *&U, int M)
{
    int n;
    cout << "Nhap so luong sinh vien ";
    cin >> n;
    Word *K = new Word[n];
    for (int i = 0; i < n; i++)
    {
        int k;
        string data;
        cout << "Nhap key= ";
        cin >> k;
        K[i].key = k;
        cout << "Nhap ten: ";
        getline(cin, data);
        strcpy(K[i].value, data.c_str());
    }
    int i, pos;
    for (i = 0; i < M; i++)
        U[i].key = 0;
    for (i = 0; i < 5; i++)
```

```
{  
    pos = Hash(K[i].key, M);  
    U[pos] = K[i];  
}  
}
```