

Danh sách địa chỉ gặp đụng độ nhiều nhất.

Cho bảng băm địa chỉ mở, có hàm băm là $h(\text{key}) = \text{key} \% M$, hàm băm lại tuyến tính $h(\text{key}, i) = (h(\text{key}) + i) \% M$. Trong đó, key là giá trị khóa, M là kích thước bảng băm, i là lần thăm dò. Hãy viết chương trình in ra các địa chỉ trên bảng băm mà xảy ra nhiều đụng độ nhất theo một bộ dữ liệu đầu vào.

Đầu vào:

- Dòng đầu tiên là hai số n, M. Cho biết bộ dữ liệu này có n khóa được nhập vào, và bảng băm có kích thước M.
- n số kế tiếp là các giá trị khóa key được thêm vào bảng băm. Giá trị của **key** trong đoạn [10000, 100000].

Đầu ra: Một dòng gồm k địa chỉ đã xảy ra nhiều đụng độ nhất được in theo thứ tự từ nhỏ đến lớn. Mỗi địa chỉ cách nhau một khoảng trắng

Lưu ý:

- 1) Bảng băm không giới hạn số lượng khóa được nhập vào.
- 2) Số lần thăm dò nhiều nhất là M-1 lần.
- 3) Sinh viên được sử dụng bất cứ cấu trúc dữ liệu hay thư viện nào.

Ví dụ:

Input	Output
5 7 10010 10019 10021 10026 10031	0 2 //Địa chỉ 0 và 2 đã xảy ra số đụng độ bằng nhau và là số đụng độ lớn nhất.

Tìm theo mã số

Hãy hoàn thành hàm Search(Hashtable &, int, int &) để tìm một phần tử có mã số kiểu int (maso) trong một bảng băm kiểu Hashtable (ht). Hàm Search sẽ trả về kết quả là chỉ số kiểu int và số lần dò tìm (nprob) kiểu int cần thiết để tìm thấy phần tử đó. Cho biết:

- Hashtable là loại bảng băm xử lý đụng độ bằng phương pháp thăm dò.
- Hàm băm lại sử dụng phương pháp thăm dò bậc hai: $h(\text{key}, i) = ((\text{key} \% M) + i * i) \% M$. Với M là kích thước bảng băm
- Hàm băm là $h(\text{key}) = \text{key} \% M$.
- Hệ số tải của bảng băm được ấn định là 0.7, nghĩa là bảng băm luôn đảm bảo số phần tử được lưu trong bảng băm không quá 70% kích thước của bảng băm.
- Hằng ký hiệu EMPTY 0 là giá trị mã số quy định cho trường hợp vị trí tương ứng trên bảng băm đang còn trống.

- Hằng ký hiệu DELETE -1 là giá trị mã số quy định cho trường hợp vị trí tương ứng trên bảng băm có phần tử trước đó nhưng đã bị xóa.

Lưu ý:

- 1) Chỉ cài đặt hàm Search.
- 2) Các kiểu dữ liệu và các hàm cần thiết của các kiểu dữ liệu đã được cài đặt sẵn. Sinh viên có thể đọc để sử dụng.
- 3) Đầu vào và đầu ra đã được xử lý sẵn và phù hợp với định dạng nhập/xuất.

Thêm học sinh

Hãy hoàn thành hàm Insert(Hashtable &, Hocsinh) để thêm một phần tử có kiểu dữ liệu Hocsinh vào một bảng băm kiểu Hashtable. Hàm Insert sẽ trả về kết quả 1 nếu thêm phần tử thành công và trả về 0 nếu không thêm phần tử được. Cho biết:

- Hashtable là loại bảng băm xử lý đụng độ bằng phương pháp thăm dò.
- Hàm băm lại sử dụng phương pháp thăm dò tuyến tính: $h(\text{key}, i) = ((\text{key} \% M) + i) \% M$. Với M là kích thước bảng băm
- Hàm băm là $h(\text{key}) = \text{key} \% M$.
- Hệ số tải của bảng băm được ấn định là 0.7, nghĩa là bảng băm luôn đảm bảo số phần tử được lưu trong bảng băm không quá 70% kích thước của bảng băm.
- Hằng ký hiệu EMPTY 0 là giá trị mã số quy định cho trường hợp vị trí tương ứng trên bảng băm đang còn trống.
- Hằng ký hiệu DELETE -1 là giá trị mã số quy định cho trường hợp vị trí tương ứng trên bảng băm có phần tử trước đó nhưng đã bị xóa.

Lưu ý:

- 1) Chỉ cài đặt hàm Insert.
- 2) Các kiểu dữ liệu và các hàm cần thiết của các kiểu dữ liệu đã được cài đặt sẵn. Sinh viên có thể đọc để sử dụng.
- 3) Đầu vào và đầu ra đã được xử lý sẵn và phù hợp với định dạng nhập/xuất.

Thêm phần tử với điều kiện giới hạn.

Hãy cài đặt bảng băm nối kết có kích thước M phần tử, hệ số tải LOAD. Dùng bảng băm này để thêm vào N khóa (là các số nguyên). Sau đó in ra kết quả tìm N' khóa trong bảng băm. Biết rằng:

- Hàm băm có dạng $h(k) = k \% M$.
- Khi thêm các khóa vào bảng băm phải đảm bảo hệ số tải của bảng băm không lớn hơn giá trị LOAD.

Input:

- Dòng đầu tiên là 3 số M, LOAD và N.
- N số nguyên tiếp theo là các giá trị khóa cần thêm, lần lượt được thêm vào theo thứ tự nhập.
- Tiếp theo là một số nguyên N'
- N' số nguyên tiếp theo là các giá trị khóa cần tìm.

Output: dòng thứ i là kết quả tìm khóa thứ i trên bảng băm. Nếu tìm thấy khóa thì in ra số lần so sánh giá trị khóa. Nếu không tìm thấy khóa thì in ra "KHONG"

Ví dụ:

Input	Output
11 0.7 8	2
12 38 27 23 44 8 91 51	KHONG
5	KHONG
23 48 51 27 91	2
	1

Thêm phần tử học sinh.

Hãy hoàn thành hàm Insert(Hashtable &, Hocsinh) để thêm một phần tử có kiểu dữ liệu Hocsinh vào một bảng băm kiểu Hashtable. Hàm Insert sẽ trả về kết quả 1 nếu thêm phần tử thành công và trả về 0 nếu không thêm phần tử được. Cho biết:

- Hashtable là loại bảng băm nối kết trực tiếp.
- Sử dụng liên kết đơn List để giải quyết đụng độ.
- Hàm băm Hash sẽ biến đổi dữ liệu mã số (trường Maso) của học sinh thành chỉ số trong bảng băm
- Hệ số tải của bảng băm được ấn định là 0.7, nghĩa là bảng băm luôn đảm bảo số phần tử được lưu trong bảng băm không quá 70% kích thước của bảng băm.

Lưu ý:

- 1) Chỉ cài đặt hàm Insert.
- 2) Các kiểu dữ liệu và các hàm cần thiết của các kiểu dữ liệu đã được cài đặt sẵn. Sinh viên có thể đọc để sử dụng.
- 3) Đầu vào và đầu ra đã được xử lý sẵn và phù hợp với định dạng nhập/xuất.

Xóa phần tử học sinh.

Hãy hoàn thành hàm int Delete(Hashtable , int) để xóa phần tử có kiểu dữ liệu Hocsinh có Maso bằng với mã số cần xóa trong một bảng băm kiểu Hashtable. Hàm Delete sẽ trả về kết quả là 1

nếu xóa phần tử thành công và trả về 0 nếu không xóa phần tử được. Cho biết:

- Hashtable là loại bảng băm nối kết trực tiếp.
- Sử dụng liên kết đơn List để giải quyết đụng độ.
- Hàm băm Hash sẽ biến đổi dữ liệu mã số (trường Maso) của học sinh thành chỉ số trong bảng băm
- Hệ số tải của bảng băm được ấn định là 0.7, nghĩa là bảng băm luôn đảm bảo số phần tử được lưu trong bảng băm không quá 70% kích thước của bảng băm.

Lưu ý:

- 1) Chỉ cài đặt hàm Delete.
- 2) Các kiểu dữ liệu và các hàm cần thiết của các kiểu dữ liệu đã được cài đặt sẵn. Sinh viên có thể đọc để sử dụng.
- 3) Đầu vào và đầu ra đã được xử lý sẵn và phù hợp với định dạng nhập/xuất.

Xóa phần tử.

Hãy hoàn thành hàm Delete(Hashtable &, int, int &) để xóa một phần tử có mã số kiểu int (maso) trong một bảng băm kiểu Hashtable (ht). Hàm Delete sẽ trả về kết quả là số nguyên 1 và số lần dò tìm (nprob) kiểu int cần thiết để xóa phần tử đó hoặc trả về kết quả là số 0 nếu không xóa được. Cho biết:

- Hashtable là loại bảng băm xử lý đụng độ bằng phương pháp thăm dò.
- Hàm băm lại sử dụng phương pháp thăm dò bậc hai: $h(\text{key}, i) = ((\text{key} \% M) + i * i) \% M$. Với M là kích thước bảng băm
- Hàm băm là $h(\text{key}) = \text{key} \% M$.
- Hệ số tải của bảng băm được ấn định là 0.7, nghĩa là bảng băm luôn đảm bảo số phần tử được lưu trong bảng băm không quá 70% kích thước của bảng băm.
- Hằng ký hiệu EMPTY 0 là giá trị mã số quy định cho trường hợp vị trí tương ứng trên bảng băm đang còn trống.
- Hằng ký hiệu DELETE -1 là giá trị mã số quy định cho trường hợp vị trí tương ứng trên bảng băm có phần tử trước đó nhưng đã bị xóa.

Lưu ý:

- 1) Chỉ cài đặt hàm Delete.
- 2) Các kiểu dữ liệu và các hàm cần thiết của các kiểu dữ liệu đã được cài đặt sẵn. Sinh viên có thể đọc để sử dụng.
- 3) Đầu vào và đầu ra đã được xử lý sẵn và phù hợp với định dạng nhập/xuất.

Tổng số đụng độ.

Viết chương trình thêm n khóa số nguyên dương vào một bảng băm địa chỉ mở và cho biết tổng số đựng độ khi thêm n khóa này. Biết bảng băm có:

- Kích thước bảng băm M.
- Hàm băm: $h(\text{key}) = \text{key} \% M$.
- Hàm băm lại: $h(\text{key}, i) = (h(\text{key}) + i) \% M$.
- Số lần băm lại từ 1 đến M-1.
- Không có giới hạn số lượng khóa được nhập vào bảng băm.

Chương trình có đầu vào như sau:

- Dòng đầu tiên cặp số M và n.
- n dòng tiếp theo, mỗi dòng chứa một khóa cần thêm vào bảng băm.

Đầu ra: tổng số đựng độ.

Lưu ý: sinh viên được sử dụng cấu trúc dữ liệu và thư viện bất kỳ

Input	Output
7 5 10010 10019 10021 10026 10031	2 //Số 10026 đựng độ 1 lần ở vị trí 2. Số 10031 đựng độ 1 lần ở vị trí 0
7 5 10010 10017 10021 10027 10031	3 //Số 10017 đựng độ 1 lần ở địa chỉ 0. Số 10031 đựng độ 2 lần ở địa chỉ 0 và 1.

Tìm học sinh.

Hãy hoàn thành hàm `Node * Search(Hashtable , int)` để trả về kết quả tìm một phần tử có kiểu dữ liệu `Hocsinh` có `Maso` bằng với mã số cần tìm trong một bảng băm kiểu `Hashtable`. Hàm `Search` sẽ trả về kết quả là con trỏ đến phần tử tìm được nếu tìm phần tử thành công và trả về con trỏ `NULL` nếu không tìm phần tử được. Cho biết:

- `Hashtable` là loại bảng băm nối kết trực tiếp.
- Sử dụng liên kết đơn `List` để giải quyết đựng độ.
- Hàm băm `Hash` sẽ biến đổi dữ liệu mã số (trường `Maso`) của học sinh thành chỉ số trong bảng băm
- Hệ số tải của bảng băm được ấn định là 0.7, nghĩa là bảng băm luôn đảm bảo số phần tử được lưu trong bảng băm không quá 70% kích thước của bảng băm.

Lưu ý:

- 1) Chỉ cài đặt hàm `Search`.
- 2) Các kiểu dữ liệu và các hàm cần thiết của các kiểu dữ liệu đã được cài đặt sẵn. Sinh viên có

thể đọc để sử dụng.

3) Đầu vào và đầu ra đã được xử lý sẵn và phù hợp với định dạng nhập/xuất.

Địa chỉ ít đụng độ nhất.

Cho bảng băm nối kết, có hàm băm là $h(\text{key}) = \text{key} \% M$, trong đó key là giá trị khóa, M là kích thước bảng băm. Hãy viết chương trình in ra các địa chỉ trên bảng băm có đụng độ nhưng xảy ra ít đụng độ nhất, theo một bộ dữ liệu đầu vào.

Đầu vào:

- Dòng đầu tiên là hai số n, M . Cho biết bộ dữ liệu này có n khóa được nhập vào, và bảng băm có kích thước M .
- n số kế tiếp là các giá trị khóa key được thêm vào bảng băm. Giá trị của **key** trong đoạn $[10000, 100000]$.

Đầu ra: Một dòng duy nhất, gồm k địa chỉ trên bảng băm đã xảy ra đụng độ nhưng với số lượng ít nhất. **Các địa chỉ này được in theo thứ tự tăng dần.**

Lưu ý:

- 1) Bảng băm không giới hạn số lượng khóa được nhập vào.
- 2) Sinh viên được sử dụng bất cứ cấu trúc dữ liệu hay thư viện nào.

Ví dụ:

Input	Output
5 7 10012 10019 10021 10026 10035	4 //Địa chỉ 2 và 4 đều đã xảy ra đụng độ, nhưng địa chỉ 4 có ít đụng độ xảy ra nhất.

Địa chỉ nhiều đụng độ nhất.

Cho bảng băm nối kết, có hàm băm là $h(\text{key}) = \text{key} \% M$, trong đó key là giá trị khóa, M là kích thước bảng băm. Hãy viết chương trình in ra các địa chỉ trên bảng băm mà xảy ra nhiều đụng độ nhất theo một bộ dữ liệu đầu vào.

Đầu vào:

- Dòng đầu tiên là hai số n, M . Cho biết bộ dữ liệu này có n khóa được nhập vào, và bảng băm có kích thước M .
- n số kế tiếp là các giá trị khóa key được thêm vào bảng băm. Giá trị của **key** trong đoạn $[10000, 100000]$.

Đầu ra: k dòng, mỗi dòng chứa là địa chỉ xảy ra nhiều đụng độ nhất

Lưu ý:

- 1) Bảng băm không giới hạn số lượng khóa được nhập vào.
- 2) Sinh viên được sử dụng bất cứ cấu trúc dữ liệu hay thư viện nào.

Ví dụ:

Input	Output
5 7 10010 10019 10021 10026 10031	0 2 //Địa chỉ 0 và 2 đã xảy ra số đựng độ bằng nhau và là số đựng độ lớn nhất.