```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "hash.h"

hash * hash_create() {
	hash * HT;

	if ((HT = (hash *)malloc(sizeof(hash))) == NULL) {
		printf("malloc failed\n");
		return NULL;
	}

	memset(HT, 0, sizeof(hash));

	return HT;
}

int hash_insert(hash *HT, datatype key) {
	linklist p, q;

	if (HT == NULL) {
		printf("HT is NULL\n");
		return -1;
	}

	if ((p = (linklist)malloc(sizeof(listnode))) == NULL) {
		printf("malloc failed\n");
		return -1;
	}
	p->key = key;
	p->value = key % N;
	p->next = NULL;

	q = &(HT->data[key % N]);

	while (q->next && q->next->key < p->key ) {
		q = q->next;
	}

	p->next = q->next;
	q->next = p;

	return 0;

}

linklist  hash_search(hash *HT, datatype key) {
	linklist p;

	if (HT == NULL) {
		printf("HT is NULL\n");
		return NULL;
	}

	p = &(HT->data[key % N]);

	while (p->next && p->next->key != key) {
		p = p->next;
```

malloc → sizeof 接.判

→

初始化 → memset

创

→ 找到对应下标，对应位置（next为NULL或其值更大）

插 P伸手，Q抻手

next为

找 对应下标，对应位置（NULL或等key）

```
        }

        if (p->next == NULL) {
                return NULL;
        } else {
                printf("found\n");
                return p->next;
        }
}
```

返 next null null

非null next