

```
//An implementation of Algorithm 4.3 and Algorithm 4.4
//written by Kazutoshi Ando (Shizuoka University)

#include <stdio.h>
#define n 16
#define m 24

int hash(int x) {
    return x%m;
}

void show_array(int A[], int l) { // 配列の中身を表示する関数
    int i;
    printf("[");
    for (i=0;i<l;i++) {
        printf(" %2d",A[i]);
        if(i<l-1) printf(",");
    }
    printf("]\n");
}

void hash_insert(int H[], int D[]) { // データの格納 (Algorithm 4.3)
    int i,k;
    for(i=0;i<n;i++) {
        k = hash(D[i]);
        while (H[k]!=-1) k = (k+1)%m;
        H[k]=D[i];
    }
}

void hash_search(int H[], int x) { // ハッシュ法による探索 (Algorithm 4.4)
    int k = hash(x);
    while (H[k]!=-1) {
        if(H[k]==x) {
            printf(" H[%d] == %d\n", k, H[k]);
            exit(0);
        }
        k = (k+1)%m;
    }
    printf(" %d is not found in H[]\n",x);
}

main(int argc, char *argv[]) {
    int k;
    int D[n]={1,2,5,6,9,11,13,15,17,20,23,24,28,29,33,39};
    int H[m];

    for(k=0;k<m;k++) H[k]=-1;
    // H[k] == -1 のとき, H[k]にはデータが格納されていないと考える.

    hash_insert(H,D);

    show_array(H,m);

    hash_search(H,23);
    //上の1行を下の2行で置き換えれば探索する値をコマンドラインから
    //与えることができる.
    //x=atoi(argv[1]);
    //hash_search(H,x);
}
```