**HASHMAPS**

#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>

#include <stdint.h>

typedef struct dbEntry

{

  char\* key;

  int64\_t value;

} dbEntry;

int compare\_predicate(const void \*e1, const void \*e2)

{

  dbEntry\* entry1 = (struct dbEntry\*)e1;

  dbEntry\* entry2 = (struct dbEntry\*)e2;

  return strcmp(entry1->key, entry2->key);

}

int main() {

  uint64\_t numEntries = 0;

  scanf("%lu\n", &numEntries);

  dbEntry\* db = (dbEntry\*)malloc(numEntries \* sizeof(dbEntry));

  for (uint64\_t i = 0; i < numEntries; ++i)

  {

    char key[1000] = {};

    int64\_t value = 0;

    scanf("%s %ld\n", key, &value);

    const int keyLength = strlen(key);

    db[i].key = (char\*)malloc(keyLength);

    strcpy(db[i].key, key);

    db[i].value = value;

  }

  qsort(db, numEntries, sizeof(dbEntry), compare\_predicate);

  while (1)

  {

    char key[1000] = {};

    int result = scanf("%s\n", key);

    if (result < 0)

    {

      break;

    }

    dbEntry keyEntry = {key, 0};

    dbEntry\* findEntry = bsearch(&keyEntry, db, numEntries, sizeof(dbEntry), compare\_predicate);

    if (findEntry != 0)

    {

      printf("%s=%ld\n", findEntry->key, findEntry->value);

    }

    else

    {

      printf("Not found\n");

    }

  }

  return 0;

}