書同文

問題敘述

在小高幫助秦始皇規劃通往各地的道路之後，秦始皇現在能夠更方便的前往各地了。

然而，在秦始皇遊歷各處後，發現各地的方言差異實在太大，導致他無法順利的溝通，因此，他又拜託小高幫他製作一個能簡易的翻譯系統，讓他能夠查詢各個單詞的翻譯。

雖然小高將這項工作接了下來，但他還忙著重修服學，因此，他又決定將這項工作交給你來完成。

同樣的，他已經完成了這支程式的大部分內容，你只需要完成他剩下來的部分即可。

簡單來說

這邊是定義：

```
// hash.h
typedef struct Data {
    int key;
    int value;
} Data;

void insert(Data *data);
void remove(int key);
int search(int key);
```

你需要實作三個函式如下：

`void insert(Data *data)`

- 將 Data 中的資料儲存起來

`void remove(int key)`

- 將資料移除

`int search(int key)`

- 尋找資料
- 如果有找到回傳 `Data.value`
- 如果沒找到回傳 `-1`

輸入說明

第一行輸入一個 n ，表示有 n 條命令

接下來 n 行，有可能是以下三種輸入：

- `insert key value`
  - 插入一筆資料
- `delete key`
  - 刪除一筆資料
- `search key`
  - 搜尋一筆資料

輸出說明

當命令為 `search key` 時，請輸出該筆資料的 `value`，若找不到則輸出 -1。

測資限制

- $1 \leq n \leq 10^4$
- $0 \leq key, value \leq 2^{31} - 1$
- 保證不會重複插入相同 key
- 保證不會刪除不存在的 key

如何繳交

請將函式獨立於一個檔案中，並且**一定要包含** `#include "hash.h"`，繳交至 Judge 時，只需要上傳該檔案即可。

例如，這是你的資料夾：

```
├─ main.c
├─ hash.h
└─ hw.c // 或其他你自己命名的名字
```

你需要上傳的只有 `hw.c`，該檔案的內容大致如下:

```
// hw.c
#include "hash.h"    // 必須要有這行
#include <stdlib.h> // 其他你用到的函式庫
.
.
.
// 在這裡實作你的程式
```

檔案中可以包含你其他會用到的東西，如 struct、function 之類的。

請注意，**不要修改** `main.c` 及 `hash.h` 的內容，這兩個檔案會在評分時**由 Judge 系統生成**，請不要在你的電腦上改完之後又來問：**為什麼在 Judge 上會 Compile Error**。

範例測資

範例輸入 1

```
30
insert 443694933 81371000
insert 61757449 1003077674
insert 406793576 1580407251
insert 770352567 904226528
insert 1670144587 503273572
insert 1500193197 1425909884
insert 1864345723 1051933261
insert 369692695 2016047330
insert 1773477031 1865212070
insert 157813624 240920310
search 1401121885
search 1500193197
insert 741252946 193732643
insert 1490170534 406468400
search 773793200
search 61757449
insert 2141123046 1961922344
insert 202945163 192186554
search 1581885387
search 443694933
delete 202945163
insert 1439782279 795187997
insert 1206060612 1927025915
search 770352567
delete 1864345723
search 1864345723
insert 890866943 1105405219
search 202945163
insert 872463718 1887525683
search 2141123046
```

範例輸出 1

```
-1
1425909884
-1
1003077674
-1
81371000
904226528
-1
-1
1961922344
```

附件

View On Github Gist

A. hash.h

```h
// hash.h
typedef struct Data {
    int key;
    int value;
} Data;

void insert(Data *data);
void remove(int key);
int search(int key);
```

B. main.c

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

int main() {
    int command_count;
    scanf("%d", &command_count);

    for (int i = 0; i < command_count; i++) {
        char command[8];
        scanf("%s", command);
        if (command[0] == 'i') {
            Data *data = malloc(sizeof(Data));
            scanf("%d %d", &data->key, &data->value);
            insert(data);
        } else if (command[0] == 'd') {
            int key;
            scanf("%d", &key);
```

```
            remove(key);
        } else if (command[0] == 's') {
            int key;
            scanf("%d", &key);
            printf("%d\n", search(key));
        }
    }

    return 0;
}
```

Unified Writing System

Problem Description

After Xiao Gao helped Qin Shi Huang plan roads to various places, Qin Shi Huang can now travel more conveniently.

However, after traveling around, Qin Shi Huang discovered that the differences in local dialects are too great, making communication difficult. Therefore, he asked Xiao Gao to create a simple translation system that would allow him to look up translations of various words.

Although Xiao Gao took on this task, he is still busy retaking Service Study. So, he decided to hand this work over to you to complete.

Similarly, he has already completed most of the content of this program; you only need to complete the remaining part.

In simple terms

Here are the definitions:

```
// hash.h
typedef struct Data {
    int key;
    int value;
} Data;

void insert(Data *data);
void remove(int key);
int search(int key);
```

You need to implement three functions as follows:

`void insert(Data *data)`

- Store the data from Data

`void remove(int key)`

- Remove the data

`int search(int key)`

- Search for data
- If found, return `Data.value`
- If not found, return `-1`

Input Format

The first line inputs an n, indicating there are n commands.

The next n lines could be one of the following three types of input:

- `insert key value`
  - Insert a piece of data
- `delete key`
  - Delete a piece of data
- `search key`
  - Search for a piece of data

Output Format

When the command is `search key`, output the `value` of that data. If not found, output -1.

Constraints

- $1 \leq n \leq 10^4$
- $0 \leq key, value \leq 2^{31} - 1$
- Guaranteed not to insert duplicate keys
- Guaranteed not to delete non-existent keys

How to Submit

Please put the function in a separate file and **must include** `#include "hash.h"`. When submitting to Judge, only upload this file.

For example, this is your folder:

```
├─ main.c
├─ hash.h
└─ hw.c // or any other name you choose
```

You only need to upload `hw.c`. The content of this file should roughly be:

```c
// hw.c
#include "hash.h"   // This line must be included
#include <stdlib.h> // Other libraries you use
.
.
.
// Implement your program here
```

The file can include other things you need, such as structs, functions, etc.

Please note, **do not modify** the content of `main.c` and `hash.h`. These two files will be **generated by the Judge system** during grading. Do not modify them on your computer and then ask: **Why does it show Compile Error on Judge?**.

Example Test Case

Sample Input 1

```
30
insert 443694933 81371000
insert 61757449 1003077674
insert 406793576 1580407251
insert 770352567 904226528
insert 1670144587 503273572
insert 1500193197 1425909884
insert 1864345723 1051933261
insert 369692695 2016047330
insert 1773477031 1865212070
insert 157813624 240920310
search 1401121885
search 1500193197
insert 741252946 193732643
insert 1490170534 406468400
search 773793200
search 61757449
insert 2141123046 1961922344
insert 202945163 192186554
search 1581885387
search 443694933
delete 202945163
insert 1439782279 795187997
insert 1206060612 1927025915
search 770352567
delete 1864345723
search 1864345723
insert 890866943 1105405219
search 202945163
insert 872463718 1887525683
search 2141123046
```

Sample Output 1

```
-1
1425909884
-1
1003077674
-1
81371000
904226528
-1
-1
1961922344
```

Attachments

View On Github Gist

A. hash.h

```h
// hash.h
typedef struct Data {
    int key;
    int value;
} Data;

void insert(Data *data);
void remove(int key);
int search(int key);
```

B. main.c

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

int main() {
    int command_count;
    scanf("%d", &command_count);

    for (int i = 0; i < command_count; i++) {
        char command[8];
        scanf("%s", command);
        if (command[0] == 'i') {
            Data *data = malloc(sizeof(Data));
            scanf("%d %d", &data->key, &data->value);
            insert(data);
        } else if (command[0] == 'd') {
            int key;
            scanf("%d", &key);
```

```
            remove(key);
        } else if (command[0] == 's') {
            int key;
            scanf("%d", &key);
            printf("%d\n", search(key));
        }
    }

    return 0;
}
```