

解釋文檔

```
3 #define MAX_KEYS 1000
4 struct KeyValue {
5     char key[256];
6     char value[256];
7 };
8
9 struct KeyValue database[MAX_KEYS];
10 int numKeys = 0;
```

程式包含一個結構 **KeyValue**，用來表示鍵值對，每個鍵值對包括一個鍵（**key**）和一個值（**value**），這些鍵值對被存儲在一個名為 **database** 的陣列中。

有一個整數變數 **numKeys**，用來追蹤當前資料庫中的鍵值對數量，一開始設定為 **0**。

程式包含了五個主要的函式：

```
1 void create(const char* key, const char* value) {
2     if (numKeys < MAX_KEYS) {
3         strcpy(database[numKeys].key, key);
4         strcpy(database[numKeys].value, value);
5         numKeys++;
6         printf("created successfully.\n");
7     }
8     else {
9         printf("Cannot create more pairs.\n");
10    }
11}
12 void read(const char* key) {
13     for (int i = 0; i < numKeys; i++) {
14         if (strcmp(database[i].key, key) == 0) {
15             printf("Key: %s, Value: %s\n", database[i].key, database[i].value);
16             return;
17         }
18     }
19     printf("Key not found.\n");
20}
21 void update(const char* key, const char* value) {
22     for (int i = 0; i < numKeys; i++) {
23         if (strcmp(database[i].key, key) == 0) {
24             strcpy(database[i].value, value);
25             printf("updated successfully.\n");
26             return;
27         }
28     }
29     printf("Key not found.\n");
30}
31 void delet(const char* key) {
32     for (int i = 0; i < numKeys; i++) {
33         if (strcmp(database[i].key, key) == 0) {
34             for (int j = i; j < numKeys - 1; j++) {
35                 strcpy(database[j].key, database[j + 1].key);
36                 strcpy(database[j].value, database[j + 1].value);
37             }
38             numKeys--;
39             printf("Key-Value pair deleted successfully.\n");
40             return;
41         }
42     }
43     printf("Key not found. Cannot delete.\n");
44 }
```

create(const char* key, const char* value): 用來創建新的鍵值對。如果資料庫未滿（即 **numKeys** 小於 **MAX_KEYS**），則將提供的鍵和值複製到資料庫中，並將 **numKeys** 增加 1。如果資料庫已滿，則顯示 "Cannot create more pairs."。

read(const char* key): 用來讀取指定鍵的值。它遍歷資料庫中的鍵值對，如果找到匹配的鍵，則輸出相應的值。如果未找到匹配的鍵，則顯示 "Key not found."。

update(const char* key, const char* value): 用來更新指定鍵的值。它遍歷資料庫中的鍵值對，如果找到匹配的鍵，則將提供的新值複製到資料庫中，並顯示 "updated successfully."。如果未找到匹配的鍵，則顯示 "Cannot update."。

delet(const char* key): 用來刪除指定鍵的鍵值對。它遍歷資料庫中的鍵值對，如果找到匹配的鍵，則將該鍵值對刪除，並調整資料庫中的其他項目。最後，將 **numKeys** 減少 1，並顯示 "Key-Value pair deleted successfully."。如果未找到匹配的鍵，則顯示 "Key not found. Cannot delete."。

```
int main() {
    char command[10];
    char key[256];
    char value[256];

    while (1) {
        printf("Enter command (create/read/update/delete/quit): ");
        scanf("%s", command);

        if (strcmp(command, "create") == 0) {
            printf("Enter key: ");
            scanf("%s", key);
            printf("Enter value: ");
            scanf("%s", value);
            create(key, value);
        }
        else if (strcmp(command, "read") == 0) {
            printf("Enter key: ");
            scanf("%s", key);
            read(key);
        }
        else if (strcmp(command, "update") == 0) {
            printf("Enter key: ");
            scanf("%s", key);
            printf("Enter new value: ");
            scanf("%s", value);
            update(key, value);
        }
        else if (strcmp(command, "delete") == 0) {
            printf("Enter key: ");
            scanf("%s", key);
            delet(key);
        }
        else if (strcmp(command, "quit") == 0) {
            break;
        }
        else {
            printf("Invalid command. Try again.\n");
        }
    }

    return 0;
}
```

main() 函式使用一個無限迴圈，等待使用者輸入。用戶可以輸入以下命令：
"create": "read": "update": "delete": "quit":

如果使用者輸入無效的命令，程式會顯示 "Invalid command. Try again."。

程式結果:

```
wpy@wpy-VirtualBox:~$ ./project
Enter command(create/read/update/delete/quit): create
Enter Key: 1
Enter value: apple
success
Enter command(create/read/update/delete/quit): create
Enter Key: 2
Enter value: hello
success
Enter command(create/read/update/delete/quit): read
Enter Key1
Key: 1, Value: apple
Enter command(create/read/update/delete/quit): read
Enter Key2
Key: 2, Value: hello
Enter command(create/read/update/delete/quit): update
Enter key: 1
Enter value: java
update success
Enter command(create/read/update/delete/quit): read
Enter Key1
Key: 1, Value: java
Enter command(create/read/update/delete/quit): delate
Invalid command
Enter command(create/read/update/delete/quit): delete
Enter key: 1
deleted successfully
Enter command(create/read/update/delete/quit): read
Enter Key1
Key not found.
```