Programiz
C Online Compiler

Programiz PRO ›

main.c

Share | Run

```c
#include <stdio.h>
#include <stdlib.h>

#define SIZE 20

// Define the DataItem structure
struct DataItem {
    int data;
    int key;
};

// Declare the hash array and dummy item
struct DataItem* hashArray[SIZE];
struct DataItem* dummyItem;

// Hash function to calculate index
int hashCode(int key) {
    return key % SIZE;
}

// Function to search for an item with a given key
struct DataItem* search(int key) {
    int hashIndex = hashCode(key);

    while(hashArray[hashIndex] != NULL) {
        if(hashArray[hashIndex]->key == key)
            return hashArray[hashIndex];
        ++hashIndex;
        hashIndex %= SIZE;
    }

    return NULL;
}

// Function to insert an item into the hash table
void insert(int key, int data) {
    struct DataItem *item = (struct DataItem*) malloc(sizeof(struct DataItem));
```

Output | Clear

```
/tmp/XZKCPumolK.o
 ~~  (1,20) (2,70) (42,80) (4,25) ~~  ~~  ~~  ~~  ~~  ~~  ~~  (12,44) (13,78) (14,32) ~~  ~~  (17,11) (37,97) ~~
Element found: 97
Element not found


=== Code Execution Successful ===78-++-t54r
```

Programiz
C Online Compiler

Programiz PRO ›

main.c

Share | Run

```c
void insert(int key, int data) {
    struct DataItem *item = (struct DataItem*) malloc(sizeof(struct DataItem));
    item->data = data;
    item->key = key;

    int hashIndex = hashCode(key);

    while(hashArray[hashIndex] != NULL && hashArray[hashIndex]->key != -1) {
        ++hashIndex;
        hashIndex %= SIZE;
    }

    hashArray[hashIndex] = item;
}

// Function to delete an item from the hash table
struct DataItem* delete(struct DataItem* item) {
    int key = item->key;

    int hashIndex = hashCode(key);

    while(hashArray[hashIndex] != NULL) {
        if(hashArray[hashIndex]->key == key) {
            struct DataItem* temp = hashArray[hashIndex];
            hashArray[hashIndex] = dummyItem;
            return temp;
        }
        ++hashIndex;
        hashIndex %= SIZE;
    }

    return NULL;
}

// Function to display the hash table
void display() {
    int i;
```

Output | Clear

```
/tmp/XZKCPumolK.o
 ~~  (1,20) (2,70) (42,80) (4,25) ~~  ~~  ~~  ~~  ~~  ~~  ~~  (12,44) (13,78) (14,32) ~~  ~~  (17,11) (37,97) ~~
Element found: 97
Element not found


=== Code Execution Successful ===78-++-t54r
```

FREE AI Code Generator: Gener | Online C Compiler - Programiz | Introducing ChatGPT | OpenAI | ChatGPT

programiz.com/c-programming/online-compiler/

Paused

Programiz
C Online Compiler

Programiz PRO ›

main.c

Share | Run

Output | Clear

```c
void display() {
    int i;
    for(i = 0; i < SIZE; i++) {
        if(hashArray[i] != NULL)
            printf(" (%d,%d)", hashArray[i]->key, hashArray[i]->data);
        else
            printf(" ~~ ");
    }
    printf("\n");
}

int main() {
    // Initialize the hash array and dummy item
    int i;
    for(i = 0; i < SIZE; i++) {
        hashArray[i] = NULL;
    }

    dummyItem = (struct DataItem*) malloc(sizeof(struct DataItem));
    dummyItem->data = -1;
    dummyItem->key = -1;

    // Insert items into the hash table
    insert(1, 20);
    insert(2, 70);
    insert(42, 80);
    insert(4, 25);
    insert(12, 44);
    insert(14, 32);
    insert(17, 11);
    insert(13, 78);
    insert(37, 97);

    // Display the hash table
    display();

    // Search for an item and display the result
```

```
/tmp/XZKCPumolK.o
 ~~  (1,20) (2,70) (42,80) (4,25) ~~  ~~  ~~  ~~  ~~  ~~  ~~  (12,44) (13,78) (14,32) ~~  ~~  (17,11) (37,97) ~~
Element found: 97
Element not found


=== Code Execution Successful ===78-++-t54r
```

Search

ENG
IN

2:22 PM
8/3/2024

Programiz
C Online Compiler

Programiz PRO ›

main.c

Share | Run

```
91      dummyItem->key = -1;
92
93      // Insert items into the hash table
94      insert(1, 20);
95      insert(2, 70);
96      insert(42, 80);
97      insert(4, 25);
98      insert(12, 44);
99      insert(14, 32);
100     insert(17, 11);
101     insert(13, 78);
102     insert(37, 97);
103
104     // Display the hash table
105     display();
106
107     // Search for an item and display the result
108     struct DataItem* item = search(37);
109     if(item != NULL)
110         printf("Element found: %d\n", item->data);
111     else
112         printf("Element not found\n");
113
114     // Delete an item and display the result
115     delete(item);
116     item = search(37);
117     if(item != NULL)
118         printf("Element found: %d\n", item->data);
119     else
120         printf("Element not found\n");
121
122     // Free allocated memory
123     free(dummyItem);
124
125     return 0;
126 }
127
```

Output | Clear

```
/tmp/XZKCPumolK.o
 ~~  (1,20) (2,70) (42,80) (4,25) ~~  ~~  ~~  ~~  ~~  ~~  ~~  (12,44) (13,78) (14,32) ~~  ~~  (17,11) (37,97) ~~
Element found: 97
Element not found


=== Code Execution Successful ===78-++-t54r
```