



main.c

Share Run

Output

Clear

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 #define SIZE 20
5
6 // Define the DataItem structure
7 struct DataItem {
8     int data;
9     int key;
10 };
11
12 // Declare the hash array and dummy item
13 struct DataItem* hashArray[SIZE];
14 struct DataItem* dummyItem;
15
16 // Hash function to calculate index
17 int hashCode(int key) {
18     return key % SIZE;
19 }
20
21 // Function to search for an item with a given key
22 struct DataItem* search(int key) {
23     int hashIndex = hashCode(key);
24
25     while(hashArray[hashIndex] != NULL) {
26         if(hashArray[hashIndex]->key == key)
27             return hashArray[hashIndex];
28         ++hashIndex;
29         hashIndex %= SIZE;
30     }
31
32     return NULL;
33 }
34
35 // Function to insert an item into the hash table
36 void insert(int key, int data) {
37     struct DataItem *item = (struct DataItem*) malloc(sizeof(struct DataItem));
38     item->key = key;
39     item->data = data;
40     int hashIndex = hashCode(key);
41
42     while(hashArray[hashIndex] != NULL) {
43         if(hashArray[hashIndex]->key == key) {
44             // Key already exists, update the data
45             hashArray[hashIndex]->data = data;
46             return;
47         }
48         ++hashIndex;
49         hashIndex %= SIZE;
50     }
51
52     // Key does not exist, insert the new item
53     hashArray[hashIndex] = item;
54 }
```

```
/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
```







main.c

Run

Clear

```
71 void display() {
72     int i;
73     for(i = 0; i < SIZE; i++) {
74         if(hashArray[i] != NULL)
75             printf(" (%d,%d)", hashArray[i]->key, hashArray[i]->data);
76         else
77             printf(" ~ ");
78     }
79     printf("\n");
80 }
81
82 int main() {
83     // Initialize the hash array and dummy item
84     int i;
85     for(i = 0; i < SIZE; i++) {
86         hashArray[i] = NULL;
87     }
88
89     dummyItem = (struct DataItem*) malloc(sizeof(struct DataItem));
90     dummyItem->data = -1;
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
```

Output

```
/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
```

LOOKING TO LEARN PROGRAMMING?  
Start your programming journey with Programiz AT NO COST.

Programiz PRO >

```
main.c
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

/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