



(<https://www.educba.com/software-development/>)



(<https://www.educba.com/hashing-function-in-c/>)



(<https://www.educba.com/recursive-function-in-c/>)



Introduction to Hash Table in C

C has a data structure called hash table which maps the keys to the values, this data structure is also present in C++. Hash functions are used by this data structure for computing the indexes of a key. The values can be stored at specific locations using the hash table index. If in case





[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

How to Create a Hash Table in C?

- Firstly, we will have to create an array of data, structure which would be a hash table.
- Now, a key has to be taken which would be stored in the hash table as input.
- After this, an index would be generated which would correspond to the key.
- If in case, any data is absent in the array's index then we will have to create the data and insert it then we will have to increase the size of a hash table.
- If in case, the data is already existing then new data is not inserted if the originally present data is not the same as the given key.
- For displaying each and every element of the hash table, we need to extract data of each index and the elements are read and eventually printed.
- For removing a key from the hash table, we need to calculate the index and the data is to be extracted then delete the specific key once the key matches.

Start Your Free Software Development Course

Web development, programming languages, Software testing & others

Working of the hash table in C

In C, a hash function is used by the hash table for computing the index or famously called hash code in an array of slots or buckets, and from these slots or buckets, the required value can be fetched. While doing the lookup, the key gets hashed and the resultant hash represents the

location of the required value stored. Generally, a specific key is assigned by the hash function to a unique slot but mostly the hash table designed has an imperfect hash function which can cause collisions in situations where the generated hash function is the index for multiple keys.





[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

```
#include
#include
#include
#include
#define EUCBACOURSE 50
struct Course* Array[EUCBACOURSE];
struct Course* Content;
struct Course* content;
struct Course {
int one;
int two;
};
int Start(int two) {
return two % EUCBACOURSE;
}
struct Course *search(int two) {
int Product = Start(two);
while(Array[Product] != NULL) {
if(Array[Product]->two == two)
return Array[Product];
++Product;

Product %= EUCBACOURSE;
}
return NULL;
```





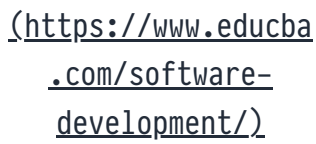
(<https://www.educba.com/software-development/>)

```

course, ,
content->one = one;
content->two = two;
int Product = Start(two);
while(Array[Product] != NULL && Array[Product]->two != +2) {
++Product;
Product %= EUCBACOURSE;
}
Array[Product] = content;
}
struct Course* delete(struct Course* content) {
int two = content->two;
int Product = Start(two);
while(Array[Product] != NULL) {
if(Array[Product]->two == two) {
struct Course* on = Array[Product];
Array[Product] = Content;
return on;
}
++Product;
Product %= EUCBACOURSE;
}
return NULL;
}
void display() {
int n = 1;

```





QUIZ



[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

```
insert(5504
, 5504);
insert(5556
, 6605);
insert(6667
, 7706);
insert(7778
, 8807);
Content->one = +2;
Content->two = +2;
display();
content = search(5566);
if(content != NULL) {
printf("Desired Course Code 1: %d\n", content->one);
} else {
printf("Desired Course Code not avialable\n");
}
delete(content);
content = search(3334);
if(content != NULL) {
printf("Desired Course Code 2: %d\n", content->one);
} else {
printf("Desired Course Code not avialable\n");
}
delete(content);
content = search(2237);
```





(<https://www.educba.com/software-development/>)

```

}
}

```

Output:

```

**..**
**..**
**..**
**..**
(4455,5040) (5556,6605) **..**
**..**
**..**
**..**
**..**
(1112,2201) **..**
**..**
**..**
(5566,6050) (6667,7706) **..**
**..**
**..**
**..**
(1122,2010) (2223,3302) **..**
**..**
**..**
(6677,7060) (7778,8807) **..**
**..**
**..**
**..**
(2233,3020) (3334,4403) **..**
**..**
**..**
**..**
(7788,8070) **..**
**..**
**..**
(9991,1090) **..**
**..**
(3344,4030) (4445,5504) **..**

```

```

**..**
**..**
(8899,9080)
Desired Course Code 1: 6050
Desired Course Code 2: 4403

```





[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

- The hash table should be uniformly distributed and should not be resulting in clusters.
- A perfect hash table has to avoid collisions. A collision can occur in the hash table when a single or multiple pair of elements have the same hash value. This situation is required to be avoided.

For analyzing the hashing algorithm and assessing their performance. The following assumption can be taken.

Assumption J (uniform hashing assumption).

We will have to assume that the hash function which we are using is uniformly distributing the keys amongst the integer values ranging between 0 to M-1.

Hashing with separate chaining.

Keys are converted to arrays indices by hash function. The second major component of the hashing algorithm is its collision resolution ability. The mainstream way of collision resolution is to build a linked list of the key-value pairs for the M array indices in which the keys are hashing to their index. The major thing is to select the M which is sufficiently large so that the list is short enough for efficiently searching.

Conclusion

On the basis of this article, we understood the basics of hash table in C. We went through the major steps of creating a hash table and how it actually works. Examples are demonstrated in this article that would help the beginners in implementing hash tables.

Recommended Articles

This is a guide to Hash Table in C. Here we discuss definition, syntax, and parameters, How to create a hash table in C? examples with code implementation. You may also have a look at the





[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

3. [Lua Iterate Table \(https://www.educba.com/lua-iterate-table/\)](https://www.educba.com/lua-iterate-table/)
4. [SQL Clear Table \(https://www.educba.com/sql-clear-table/\)](https://www.educba.com/sql-clear-table/)

ALL IN ONE SOFTWARE DEVELOPMENT BUNDLE (600+ COURSES, 50+ PROJECTS)

- ☒ 600+ Online Courses
- ☒ 50+ projects
- ☒ 3000+ Hours
- ☒ Verifiable Certificates
- ☒ Lifetime Access

Learn More

<https://www.educba.com/software-development/courses/software-development-course/?btnz=edu-blg-inline-banner3>

About Us

Blog (<https://www.educba.com/blog/?source=footer>)





[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

source=footer)

Contact Us (<https://www.educba.com/contact-us/?source=footer>)

Verifiable Certificate (<https://www.educba.com/software-development/verifiable-certificate/?source=footer>)

Reviews (<https://www.educba.com/software-development/reviews/?source=footer>)

Terms and Conditions (<https://www.educba.com/terms-and-conditions/?source=footer>)

Privacy Policy (<https://www.educba.com/privacy-policy/?source=footer>)

Apps

iPhone & iPad (<https://itunes.apple.com/in/app/educba-learning-app/id1341654580?mt=8>)

Android (<https://play.google.com/store/apps/details?id=com.educba.www>)

Resources

Free Courses (<https://www.educba.com/software-development/free-courses/?source=footer>)

Java Tutorials (<https://www.educba.com/software-development/software-development-tutorials/java-tutorial/?source=footer>)

Python Tutorials (<https://www.educba.com/software-development/software-development-tutorials/python-tutorial/?source=footer>)

All Tutorials (<https://www.educba.com/software-development/software-development-tutorials/?source=footer>)

Certification Courses





[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

[development/courses/python-certification-course/?source=footer\)](https://www.educba.com/software-development/courses/python-certification-course/?source=footer)

Java Course (<https://www.educba.com/software-development/courses/java-course/?source=footer>)

Become a Selenium Automation Tester (<https://www.educba.com/software-development/courses/selenium-training-certification/?source=footer>)

Become an IoT Developer (<https://www.educba.com/software-development/courses/iot-course/?source=footer>)

ASP.NET Course (<https://www.educba.com/software-development/courses/asp-net-course/?source=footer>)

VB.NET Course (<https://www.educba.com/software-development/courses/vb-net-course/?source=footer>)

PHP Course (<https://www.educba.com/software-development/courses/php-course/?source=footer>)

© 2022 - EDUCBA. ALL RIGHTS RESERVED. THE CERTIFICATION NAMES ARE THE TRADEMARKS OF THEIR RESPECTIVE OWNERS.

