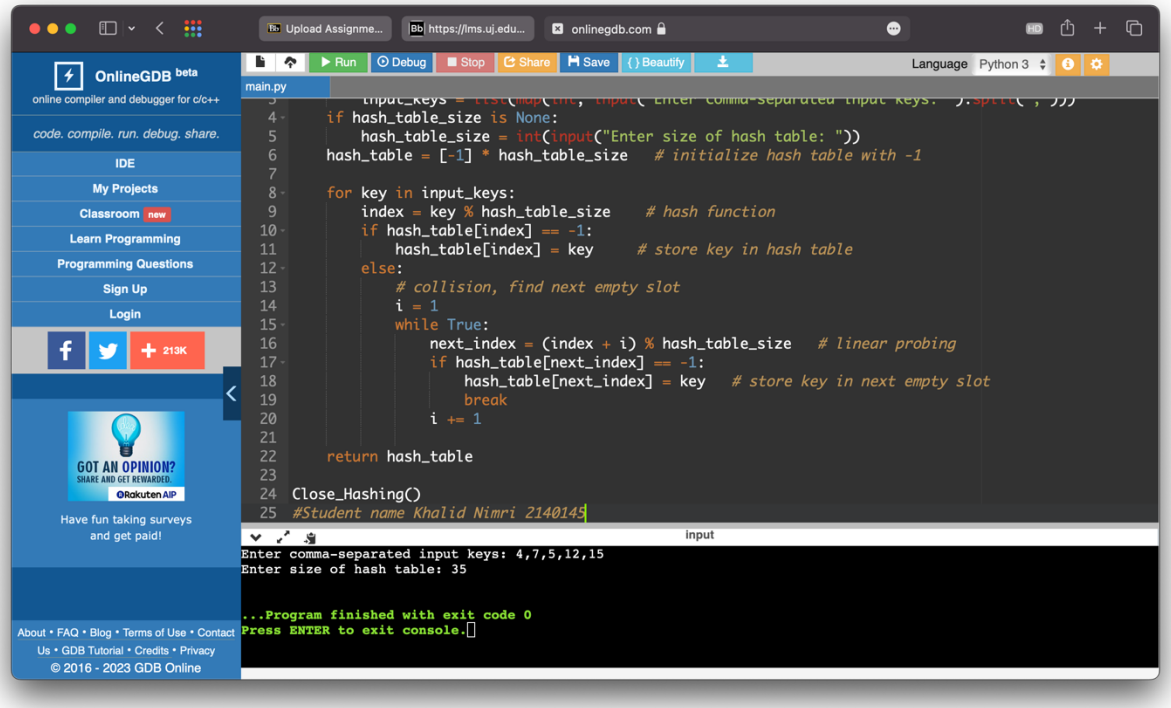
Lab 6
Student name: Khalid Nimri (2140145)

Exercise 1: (Close Hashing)

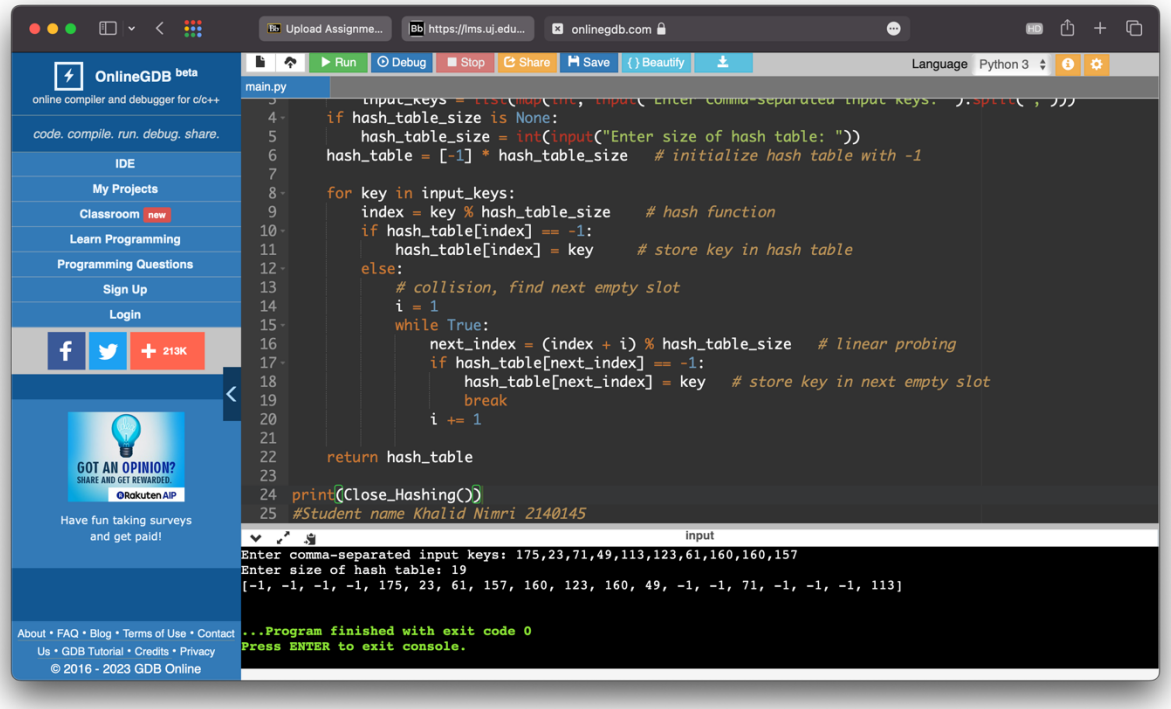
Code:

```
def Close_Hashing(input_keys=None, hash_table_size=None):
    if input_keys is None:
        input_keys = list(map(int, input("Enter comma-separated input keys: ").split(',')))
    if hash_table_size is None:
        hash_table_size = int(input("Enter size of hash table: "))
    hash_table = [-1] * hash_table_size   # initialize hash table with -1

    for key in input_keys:
        index = key % hash_table_size    # hash function
        if hash_table[index] == -1:
            hash_table[index] = key     # store key in hash table
        else:
            # collision, find next empty slot
            i = 1
            while True:
                next_index = (index + i) % hash_table_size   # linear probing
                if hash_table[next_index] == -1:
                    hash_table[next_index] = key   # store key in next empty slot
                    break
                i += 1

    return hash_table

Close_Hashing()
```

Output:

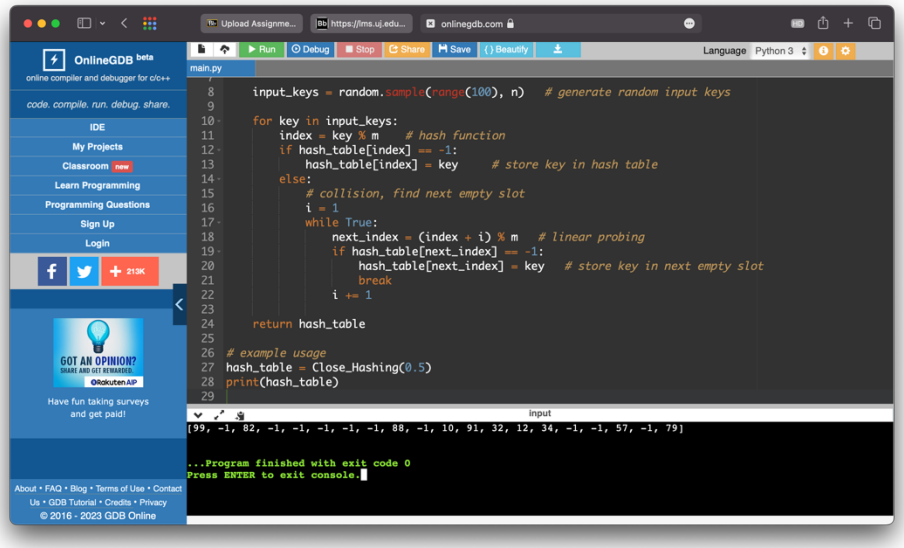


Exercise 2: Consider the following data : { 175 23 71 49 113 123 61 160 160 157}



Exercise 3:

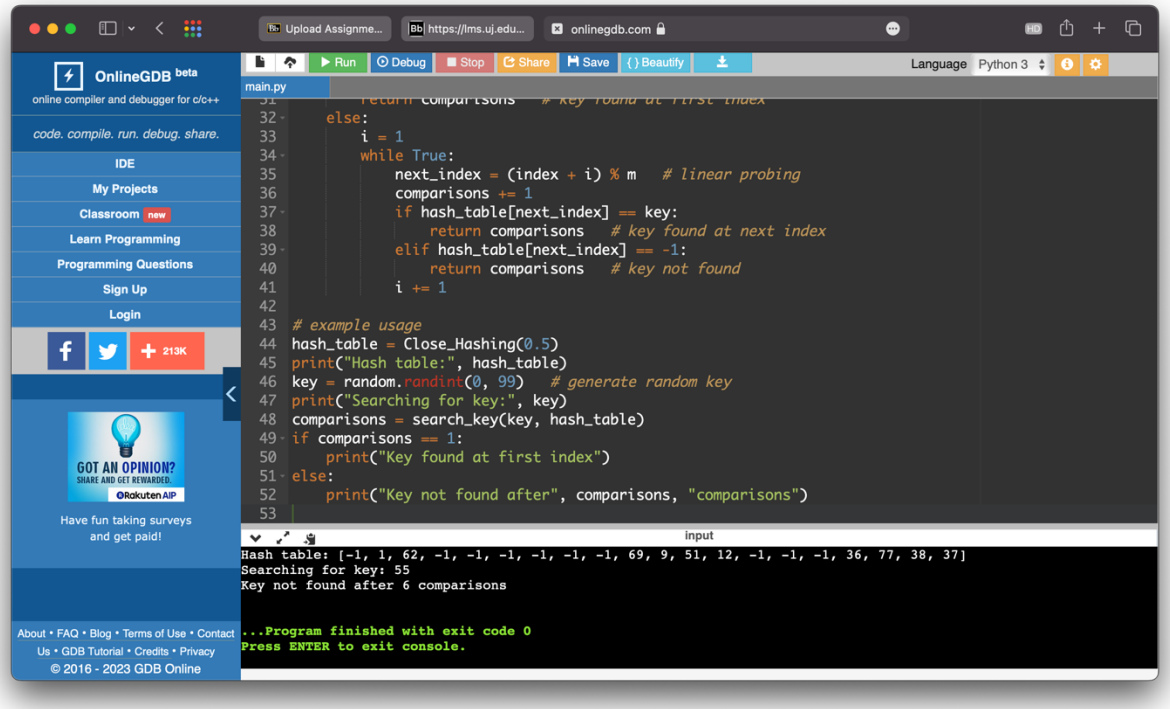
1. Set alpha to 0.5 (which means M = 2N). Use a random input and fill the hash table. attach your complete code.



```
    input_keys = random.sample(range(100), n)  # generate random input keys

    for key in input_keys:
        index = key % m    # hash function
        if hash_table[index] == -1:
            hash_table[index] = key     # store key in hash table
        else:
            # collision, find next empty slot
            i = 1
            while True:
                next_index = (index + i) % m   # linear probing
                if hash_table[next_index] == -1:
                    hash_table[next_index] = key   # store key in next empty slot
                    break
                i += 1

    return hash_table

# example usage
hash_table = Close_Hashing(0.5)
print(hash_table)
```

```
input
[99, -1, 82, -1, -1, -1, -1, -1, 88, -1, 10, 91, 32, 12, 34, -1, -1, 57, -1, 79]

...Program finished with exit code 0
Press ENTER to exit console.
```

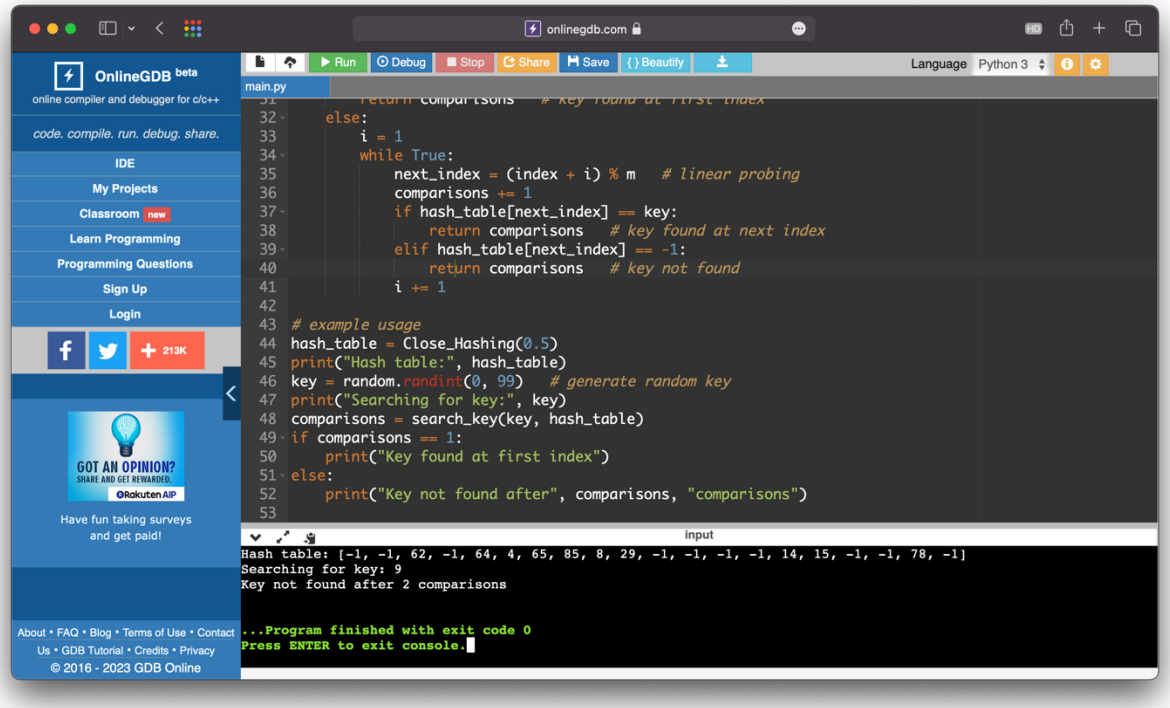
2. Generate a random key and search it in the hash table. Count the number of comparisons you had to make (regardless the search was successful or not). attach your complete code.

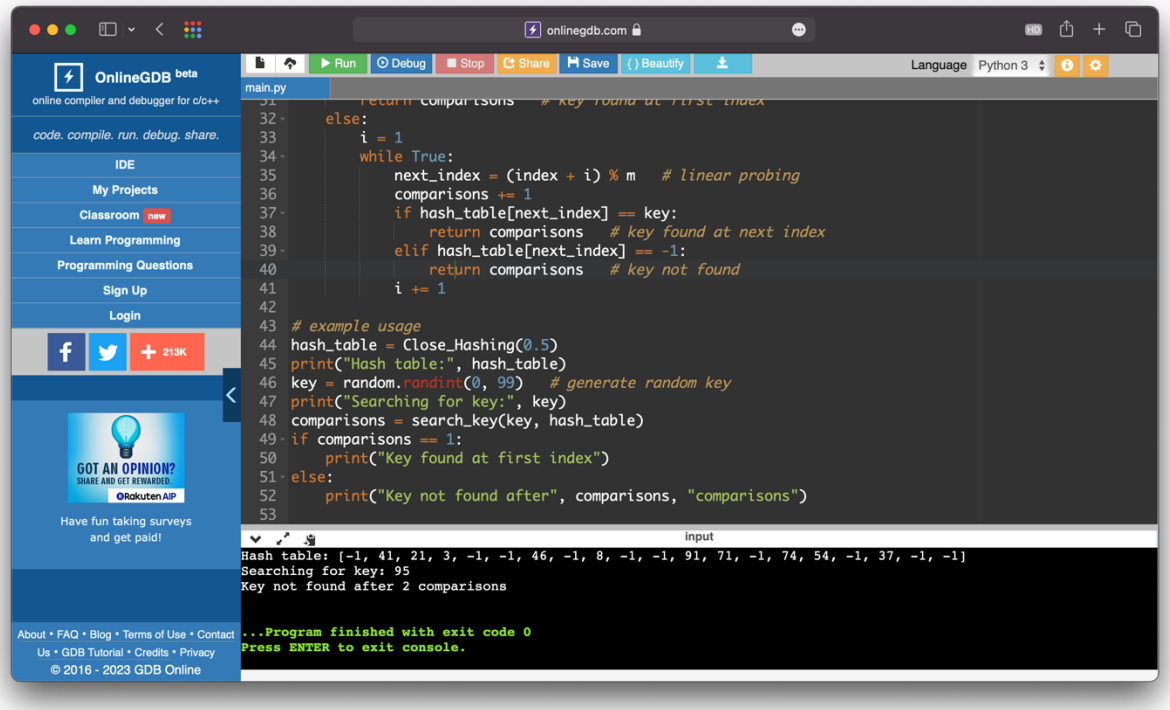


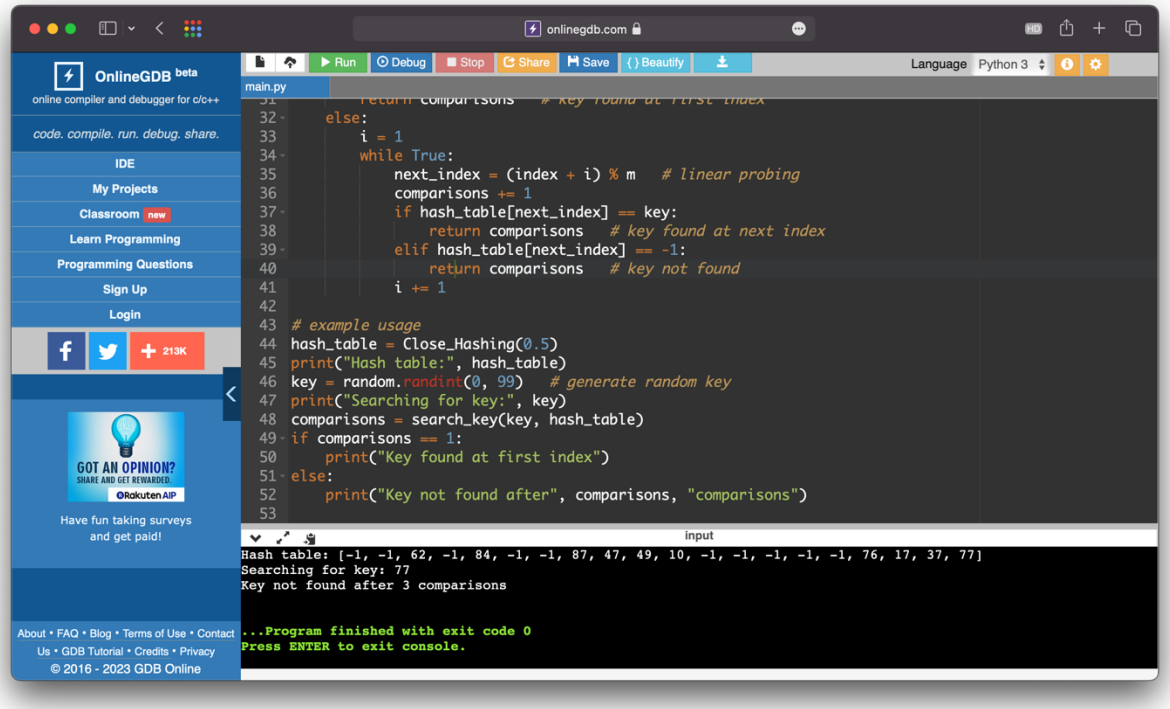
```
    else:
        i = 1
        while True:
            next_index = (index + i) % m   # linear probing
            comparisons += 1
            if hash_table[next_index] == key:
                return comparisons   # key found at next index
            elif hash_table[next_index] == -1:
                return comparisons   # key not found
            i += 1

# example usage
hash_table = Close_Hashing(0.5)
print("Hash table:", hash_table)
key = random.randint(0, 99)   # generate random key
print("Searching for key:", key)
comparisons = search_key(key, hash_table)
if comparisons == 1:
    print("Key found at first index")
else:
    print("Key not found after", comparisons, "comparisons")
```

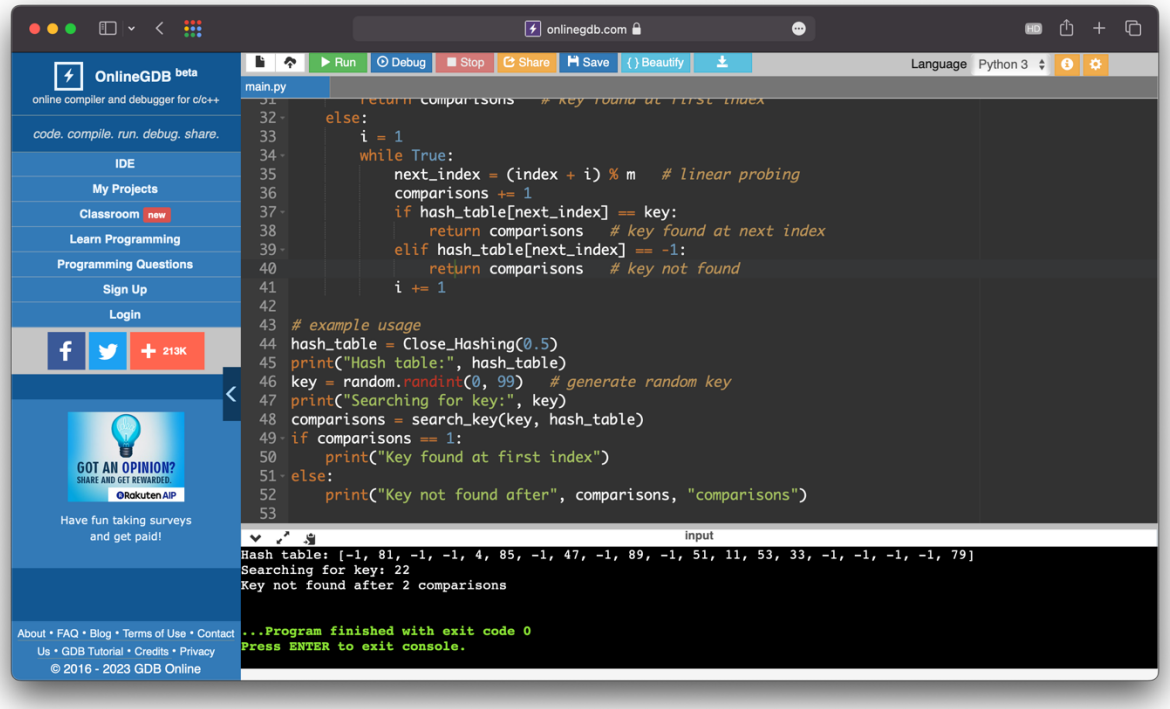
```
input
Hash table: [-1, 1, 62, -1, -1, -1, -1, -1, -1, 69, 9, 51, 12, -1, -1, -1, 36, 77, 38, 37]
Searching for key: 55
Key not found after 6 comparisons

...Program finished with exit code 0
Press ENTER to exit console.
```

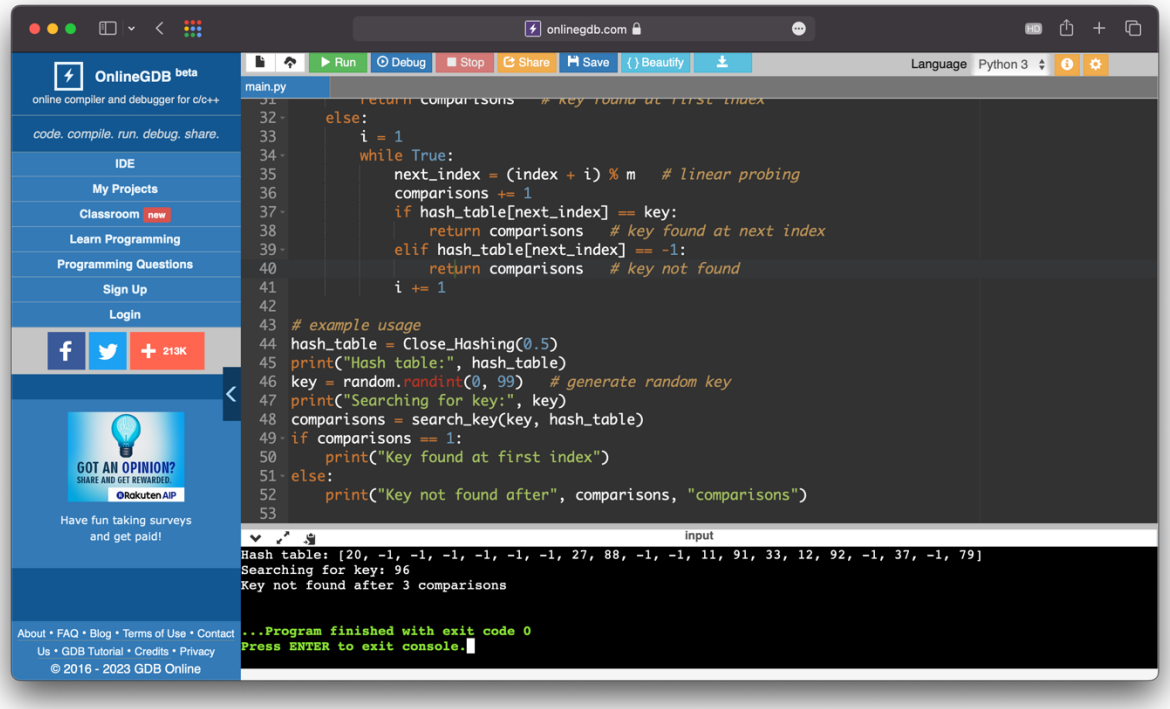
3. Do step 2 (the search operation) several times (at least 10 times) for a different key in the same hash table. Every time note down the number of comparisons. attach your complete code.

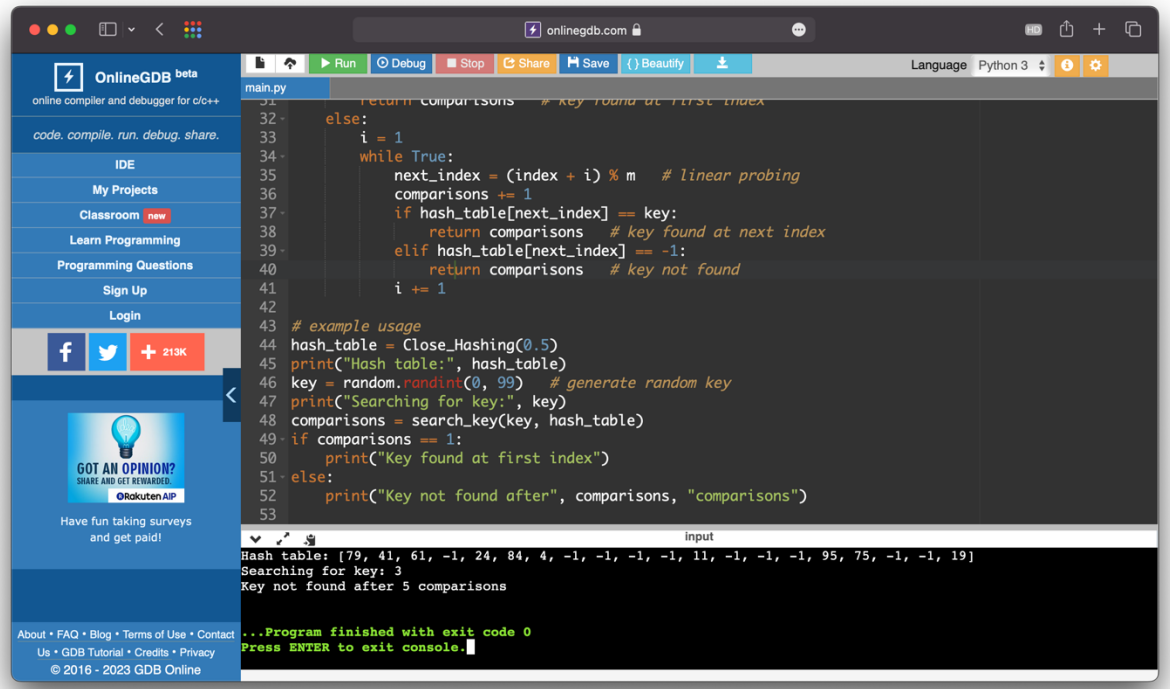


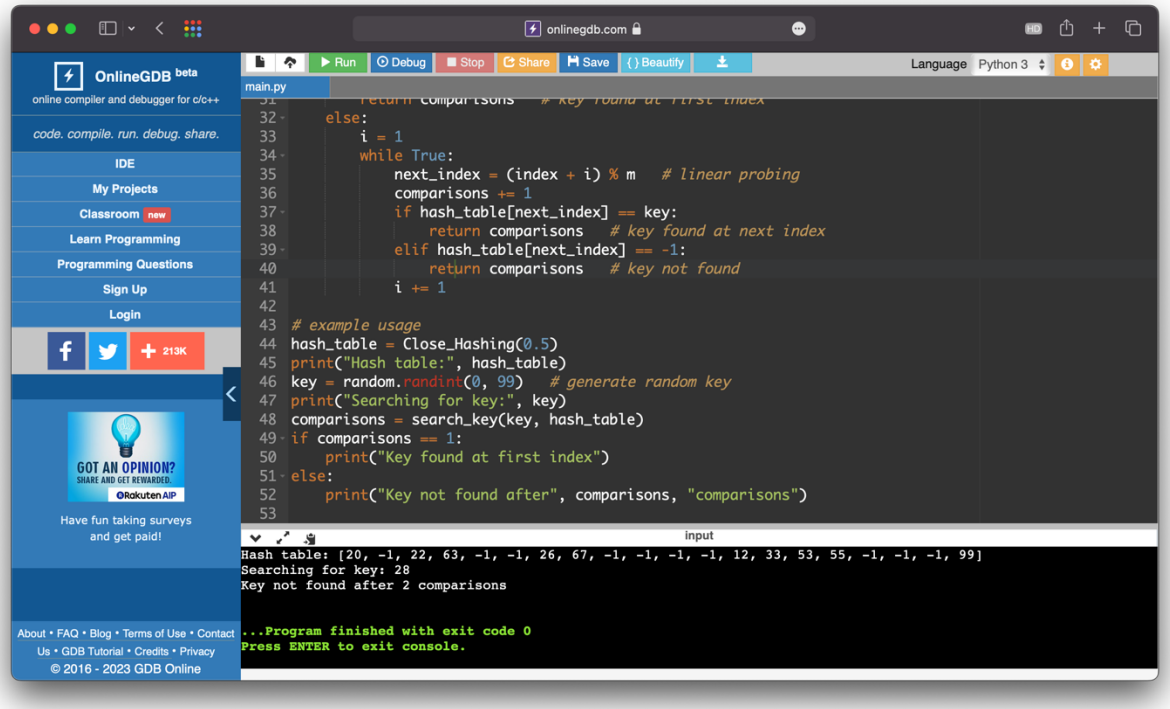


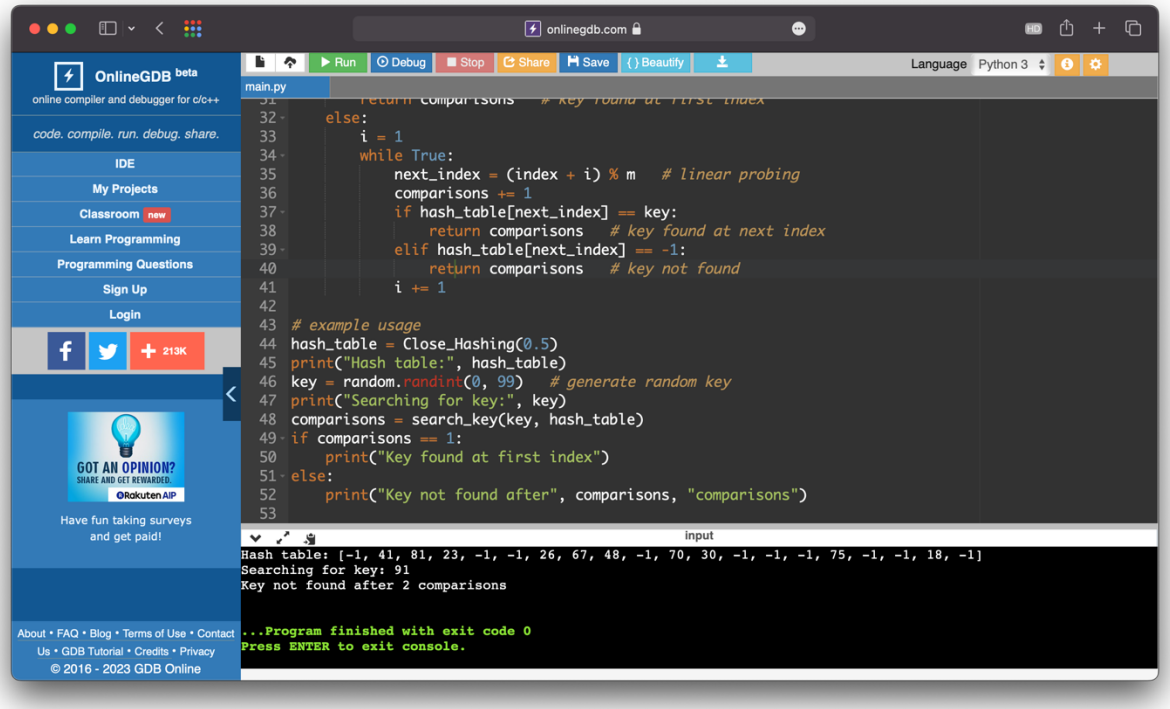

onlinegdb.com
OnlineGDB beta
online compiler and debugger for c/c++
code. compile. run. debug. share.
IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Sign Up
Login
213K
GOT AN OPINION?
SHARE AND GET REWARDED.
Rakuten AIP
Have fun taking surveys and get paid!
About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy
© 2016 - 2023 GDB Online
Run
Debug
Stop
Share
Save
{ } Beautify
Language
Python 3
main.py
32 else:
33 i = 1
34 while True:
35 next_index = (index + i) % m # linear probing
36 comparisons += 1
37 if hash_table[next_index] == key:
38 return comparisons # key found at next index
39 elif hash_table[next_index] == -1:
40 return comparisons # key not found
41 i += 1
42
43 # example usage
44 hash_table = Close_Hashing(0.5)
45 print("Hash table:", hash_table)
46 key = random.randint(0, 99) # generate random key
47 print("Searching for key:", key)
48 comparisons = search_key(key, hash_table)
49 if comparisons == 1:
50 print("Key found at first index")
51 else:
52 print("Key not found after", comparisons, "comparisons")
53
input
Hash table: [-1, -1, 62, -1, 84, -1, -1, 87, 47, 49, 10, -1, -1, -1, -1, -1, 76, 17, 37, 77]
Searching for key: 77
Key not found after 3 comparisons
...Program finished with exit code 0
Press ENTER to exit console.

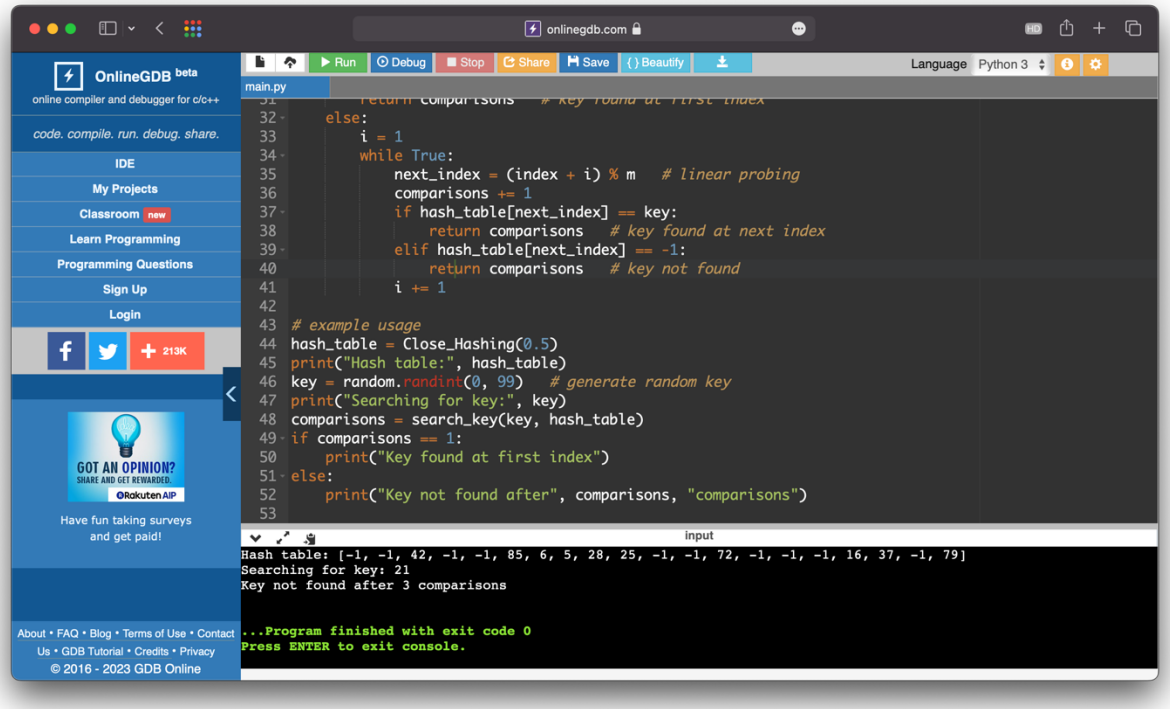

onlinegdb.com
OnlineGDB beta
online compiler and debugger for c/c++
code. compile. run. debug. share.
IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Sign Up
Login
213K
GOT AN OPINION?
SHARE AND GET REWARDED.
Rakuten AIP
Have fun taking surveys and get paid!
About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy
© 2016 - 2023 GDB Online
Run
Debug
Stop
Share
Save
{ } Beautify
Language
Python 3
main.py
32 else:
33 i = 1
34 while True:
35 next_index = (index + i) % m # linear probing
36 comparisons += 1
37 if hash_table[next_index] == key:
38 return comparisons # key found at next index
39 elif hash_table[next_index] == -1:
40 return comparisons # key not found
41 i += 1
42
43 # example usage
44 hash_table = Close_Hashing(0.5)
45 print("Hash table:", hash_table)
46 key = random.randint(0, 99) # generate random key
47 print("Searching for key:", key)
48 comparisons = search_key(key, hash_table)
49 if comparisons == 1:
50 print("Key found at first index")
51 else:
52 print("Key not found after", comparisons, "comparisons")
53
input
Hash table: [-1, 81, -1, -1, 4, 85, -1, 47, -1, 89, -1, 51, 11, 53, 33, -1, -1, -1, -1, 79]
Searching for key: 22
Key not found after 2 comparisons
...Program finished with exit code 0
Press ENTER to exit console.

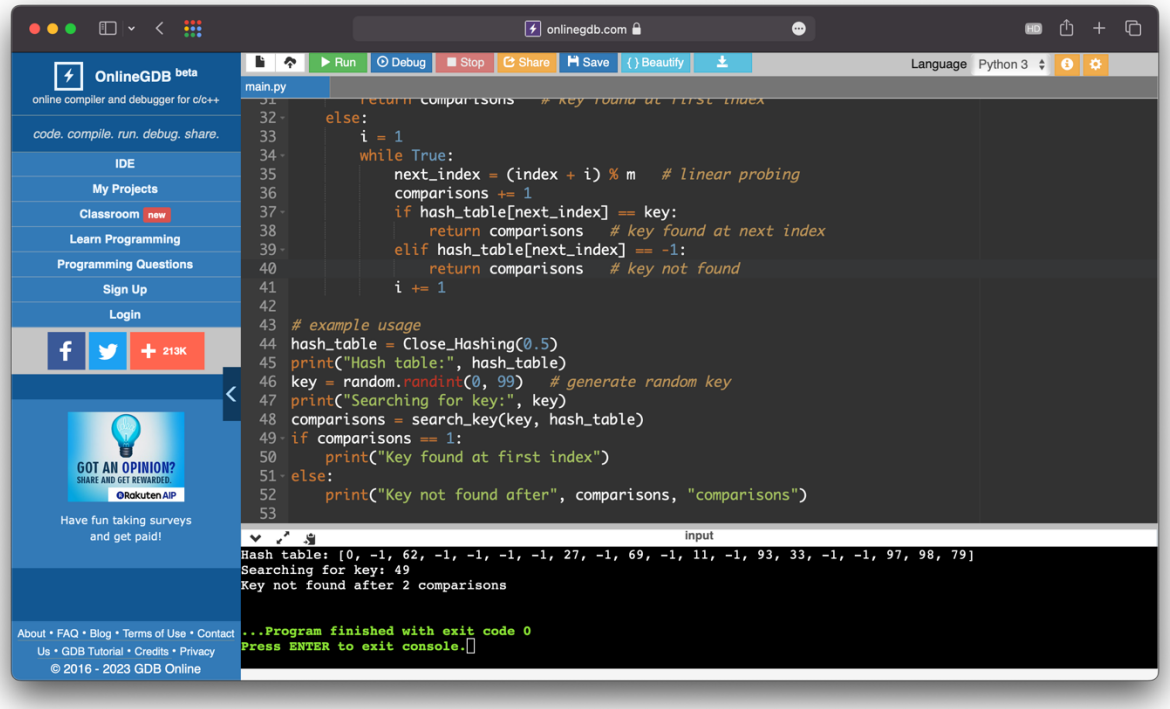

onlinegdb.com
OnlineGDB beta
online compiler and debugger for c/c++
code. compile. run. debug. share.
IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Sign Up
Login
213K
GOT AN OPINION?
SHARE AND GET REWARDED.
Rakuten AIP
Have fun taking surveys and get paid!
About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy
© 2016 - 2023 GDB Online
Run
Debug
Stop
Share
Save
{ } Beautify
Language
Python 3
main.py
else:
i = 1
while True:
next_index = (index + i) % m # linear probing
comparisons += 1
if hash_table[next_index] == key:
return comparisons # key found at next index
elif hash_table[next_index] == -1:
return comparisons # key not found
i += 1
# example usage
hash_table = Close_Hashing(0.5)
print("Hash table:", hash_table)
key = random.randint(0, 99) # generate random key
print("Searching for key:", key)
comparisons = search_key(key, hash_table)
if comparisons == 1:
print("Key found at first index")
else:
print("Key not found after", comparisons, "comparisons")
input
Hash table: [20, -1, -1, -1, -1, -1, -1, 27, 88, -1, -1, 11, 91, 33, 12, 92, -1, 37, -1, 79]
Searching for key: 96
Key not found after 3 comparisons
...Program finished with exit code 0
Press ENTER to exit console.


onlinegdb.com
OnlineGDB beta
online compiler and debugger for c/c++
code. compile. run. debug. share.
IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Sign Up
Login
213K
GOT AN OPINION?
SHARE AND GET REWARDED.
Rakuten AIP
Have fun taking surveys and get paid!
About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy
© 2016 - 2023 GDB Online
Run
Debug
Stop
Share
Save
{ } Beautify
Language
Python 3
main.py
else:
i = 1
while True:
next_index = (index + i) % m # linear probing
comparisons += 1
if hash_table[next_index] == key:
return comparisons # key found at next index
elif hash_table[next_index] == -1:
return comparisons # key not found
i += 1
# example usage
hash_table = Close_Hashing(0.5)
print("Hash table:", hash_table)
key = random.randint(0, 99) # generate random key
print("Searching for key:", key)
comparisons = search_key(key, hash_table)
if comparisons == 1:
print("Key found at first index")
else:
print("Key not found after", comparisons, "comparisons")
input
Hash table: [79, 41, 61, -1, 24, 84, 4, -1, -1, -1, -1, 11, -1, -1, -1, 95, 75, -1, -1, 19]
Searching for key: 3
Key not found after 5 comparisons
...Program finished with exit code 0
Press ENTER to exit console.


onlinegdb.com
OnlineGDB beta
online compiler and debugger for c/c++
code. compile. run. debug. share.
IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Sign Up
Login
213K
GOT AN OPINION?
SHARE AND GET REWARDED.
Rakuten AIP
Have fun taking surveys and get paid!
About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy
© 2016 - 2023 GDB Online
Run
Debug
Stop
Share
Save
{ } Beautify
Language
Python 3
main.py
else:
i = 1
while True:
next_index = (index + i) % m # linear probing
comparisons += 1
if hash_table[next_index] == key:
return comparisons # key found at next index
elif hash_table[next_index] == -1:
return comparisons # key not found
i += 1
# example usage
hash_table = Close_Hashing(0.5)
print("Hash table:", hash_table)
key = random.randint(0, 99) # generate random key
print("Searching for key:", key)
comparisons = search_key(key, hash_table)
if comparisons == 1:
print("Key found at first index")
else:
print("Key not found after", comparisons, "comparisons")
input
Hash table: [20, -1, 22, 63, -1, -1, 26, 67, -1, -1, -1, -1, 12, 33, 53, 55, -1, -1, -1, 99]
Searching for key: 28
Key not found after 2 comparisons
...Program finished with exit code 0
Press ENTER to exit console.

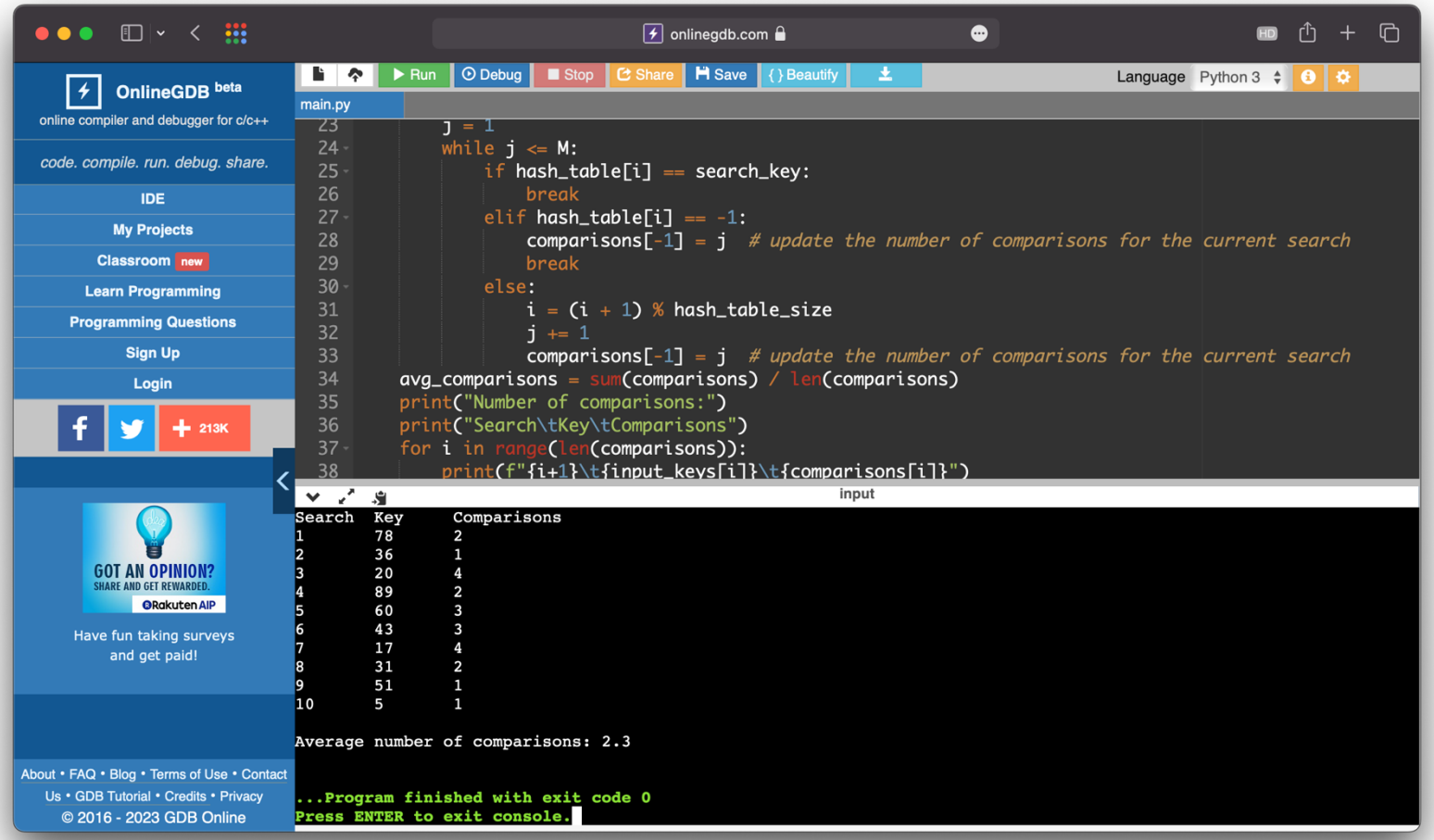

onlinegdb.com
OnlineGDB beta
online compiler and debugger for c/c++
code. compile. run. debug. share.
IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Sign Up
Login
213K
GOT AN OPINION?
SHARE AND GET REWARDED.
Rakuten AIP
Have fun taking surveys and get paid!
About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy
© 2016 - 2023 GDB Online
Run
Debug
Stop
Share
Save
{ } Beautify
Language
Python 3
main.py
else:
i = 1
while True:
next_index = (index + i) % m # linear probing
comparisons += 1
if hash_table[next_index] == key:
return comparisons # key found at next index
elif hash_table[next_index] == -1:
return comparisons # key not found
i += 1
# example usage
hash_table = Close_Hashing(0.5)
print("Hash table:", hash_table)
key = random.randint(0, 99) # generate random key
print("Searching for key:", key)
comparisons = search_key(key, hash_table)
if comparisons == 1:
print("Key found at first index")
else:
print("Key not found after", comparisons, "comparisons")
input
Hash table: [-1, 41, 81, 23, -1, -1, 26, 67, 48, -1, 70, 30, -1, -1, -1, 75, -1, -1, 18, -1]
Searching for key: 91
Key not found after 2 comparisons
...Program finished with exit code 0
Press ENTER to exit console.


onlinegdb.com
OnlineGDB beta
online compiler and debugger for c/c++
code. compile. run. debug. share.
IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Sign Up
Login
213K
GOT AN OPINION?
SHARE AND GET REWARDED.
Rakuten AIP
Have fun taking surveys and get paid!
About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy
© 2016 - 2023 GDB Online
Run
Debug
Stop
Share
Save
{ } Beautify
Language
Python 3
main.py
else:
i = 1
while True:
next_index = (index + i) % m # linear probing
comparisons += 1
if hash_table[next_index] == key:
return comparisons # key found at next index
elif hash_table[next_index] == -1:
return comparisons # key not found
i += 1
# example usage
hash_table = Close_Hashing(0.5)
print("Hash table:", hash_table)
key = random.randint(0, 99) # generate random key
print("Searching for key:", key)
comparisons = search_key(key, hash_table)
if comparisons == 1:
print("Key found at first index")
else:
print("Key not found after", comparisons, "comparisons")
input
Hash table: [-1, -1, 42, -1, -1, 85, 6, 5, 28, 25, -1, -1, 72, -1, -1, -1, 16, 37, -1, 79]
Searching for key: 21
Key not found after 3 comparisons
...Program finished with exit code 0
Press ENTER to exit console.


onlinegdb.com
OnlineGDB beta
online compiler and debugger for c/c++
code. compile. run. debug. share.
IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Sign Up
Login
213K
GOT AN OPINION?
SHARE AND GET REWARDED.
Rakuten AIP
Have fun taking surveys and get paid!
About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy
© 2016 - 2023 GDB Online
Run
Debug
Stop
Share
Save
{ } Beautify
Language
Python 3
main.py
else:
i = 1
while True:
next_index = (index + i) % m # linear probing
comparisons += 1
if hash_table[next_index] == key:
return comparisons # key found at next index
elif hash_table[next_index] == -1:
return comparisons # key not found
i += 1
# example usage
hash_table = Close_Hashing(0.5)
print("Hash table:", hash_table)
key = random.randint(0, 99) # generate random key
print("Searching for key:", key)
comparisons = search_key(key, hash_table)
if comparisons == 1:
print("Key found at first index")
else:
print("Key not found after", comparisons, "comparisons")
input
Hash table: [0, -1, 62, -1, -1, -1, -1, 27, -1, 69, -1, 11, -1, 93, 33, -1, -1, 97, 98, 79]
Searching for key: 49
Key not found after 2 comparisons
...Program finished with exit code 0
Press ENTER to exit console.

4. *Write the number of comparisons in table form and their average in the space provided below.*

Search	Key	Comparisons
1	78	2
2	36	1
3	20	4
4	89	2
5	60	3
6	43	3
7	17	4
8	31	2
9	51	1
10	5	1
Average	2.3	


onlinegdb.com
OnlineGDB beta
online compiler and debugger for c/c++
code. compile. run. debug. share.
IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Sign Up
Login
213K
GOT AN OPINION?
SHARE AND GET REWARDED.
Rakuten AIP
Have fun taking surveys and get paid!
About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy
© 2016 - 2023 GDB Online
Run
Debug
Stop
Share
Save
{ } Beautify
Language
Python 3
main.py
j = 1
while j <= M:
if hash_table[i] == search_key:
break
elif hash_table[i] == -1:
comparisons[-1] = j # update the number of comparisons for the current search
break
else:
i = (i + 1) % hash_table_size
j += 1
comparisons[-1] = j # update the number of comparisons for the current search
avg_comparisons = sum(comparisons) / len(comparisons)
print("Number of comparisons:")
print("Search\tKey\tComparisons")
for i in range(len(comparisons)):
print(f"{i+1}\t{input_keys[i]}\t{comparisons[i]}")
input
Search Key Comparisons
1 78 2
2 36 1
3 20 4
4 89 2
5 60 3
6 43 3
7 17 4
8 31 2
9 51 1
10 5 1
Average number of comparisons: 2.3
...Program finished with exit code 0
Press ENTER to exit console.

```
import random

def Close_Hashing(input_keys, hash_table_size):
    M = 2 * len(input_keys)  # set M to be twice the size of input_keys
    hash_table = [-1] * hash_table_size
    comparisons = []  # initialize a list to store the number of comparisons made during each search
    for k in input_keys:
        i = k % hash_table_size
        j = 1
        while j <= M:
            if hash_table[i] == -1:
                hash_table[i] = k
                break
            else:
                i = (i + 1) % hash_table_size
                j += 1
        if j > M:
            print("Hash table overflow")
    for i in range(10):
        search_key = random.randint(1, 100)
        comparisons.append(1)  # initialize the number of comparisons for the current search to 1
        i = search_key % hash_table_size
        j = 1
        while j <= M:
            if hash_table[i] == search_key:
                break
            elif hash_table[i] == -1:
                comparisons[-1] = j  # update the number of comparisons for the current search
                break
            else:
                i = (i + 1) % hash_table_size
                j += 1
                comparisons[-1] = j  # update the number of comparisons for the current search
    avg_comparisons = sum(comparisons) / len(comparisons)
    print("Number of comparisons:")
    print("Search\tKey\tComparisons")
    for i in range(len(comparisons)):
        print(f"{i+1}\t{input_keys[i]}\t{comparisons[i]}")
    print(f"\nAverage number of comparisons: {avg_comparisons}")

# example usage
input_keys = [78, 36, 20, 89, 60, 43, 17, 31, 51, 5]
hash_table_size = 19
                                    Close_Hashing(input_keys, hash_table_size)
```