



In this example we are creating a very simple hash table to store integers. For hashing we are using a mod function.

```
class HashTable:
    def __init__(self, M):
        self.M = M          # table size
        self.T = [None] * M # "empty" table
```

When a new **HashTable** object is called it creates a list of size M . In this example **None** means that given slot from the list has no value stored and it can be used.

Adding a new integer to the hash table is done with the hash function:

```
def add(self, x):
    i = x % self.M          # new index
    if self.T[i] == None:
        self.T[i] = x       # store the integer if the slot is free
```

At first we must check if the slot is empty. If it is we can store the new integer there. Note that normally we want to be able to store the new data even if the correct slot is not empty. Therefore, we need a way to handle collisions. See OpenDSA material to learn different approaches for collision resolution.

Removing an integer from the hash table is simply the **add** method reversed. In other words we change the value back to **None**.

```
def remove(self, x):
    i = x % self.M
    if self.T[i] == x:
        self.T[i] = None    # change the value to None only if it was 'x'
```

This time we need to check if the slot stores the correct integer to prevent unwanted deletions.

To print the hash table we can just print **T** as it is, but let's do something fancier:

Example 1.1: a simple hash table program

```

class HahsTable:
    def __init__(self, M):
        self.M = M          # table size
        self.T = [None] * M # "empty table"

    def add(self, x):
        i = x % self.M      # new index
        if self.T[i] == None:
            self.T[i] = x    # store the integer if the position is free

    def remove(self, x):
        i = x % self.M
        if self.T[i] == x:
            self.T[i] = None # change the value to None only if it was 'x'

    def print(self):
        print(4*self.M*"- " + "-\n|", end="")
        for x in self.T:
            if x == None:
                print("  |", end="")
            else:
                print(f"{x:3d}|", end="")
        print("\n" + 4*self.M*"- " + "-")

if __name__ == "__main__":
    HT = HahsTable(8)
    HT.add(312)
    HT.add(755)
    HT.add(994)
    HT.print()
    HT.remove(312)
    HT.add(140)
    HT.add(308)
    HT.add(886)
    HT.print()

```

Example 1.2: the output

```

$ python example.py
-----
|312|  |994|755|  |  |  |  |
-----
|  |  |994|755|140|  |886|  |
-----

```

Last modified: Monday, 3 July 2023, 9:58 AM

You are logged in as Hung Nguyen (Log out)

[Search and Moodle Help](#)
[Course search](#)
[Student Guide \(PDF\)](#)

[Accessibility statement](#)

[Data retention summary](#)

[Get the mobile app](#)

[Policies](#)

Copyright © LUT University