

```
class WordNode:
    def __init__(self, word="", mean=""):
        self.word = word
        self.meaning = mean
        self.isLink = -1

class HashTable:

    def __init__(self):
        self.dictionary = []
        self.maxSize = 17
        self.length = 0
        for i in range (self.maxSize):
            self.dictionary.append(WordNode( "", ""))

    def hashFunction(self, word):
        k=0
        for i in range (len(word)):
            k+=ord(word[i])
        k = int(k/(len(word)+1) )
        k = k % self.maxSize
        return k%self.maxSize

    def searchForNode(self, key):
        index = self.hashFunction(key)
        count = 0
        while(self.dictionary[index].word != "" and count < self.length):
            if(self.dictionary[index].word != "" and count<=self.length):
                if(self.dictionary[index].word == key):
                    return True
                index = self.dictionary[index].isLink
                count +=1
        return False

    def insertWithoutReplacement(self, word, mean):
        if(self.searchForNode(word)==False and self.length<self.maxSize):
            index = self.hashFunction(word)
            if(self.dictionary[index].word == ""):
                self.dictionary[index].word = word
                self.dictionary[index].meaning = mean
                self.length +=1
            elif (index == self.hashFunction(self.dictionary[index].word)):
                while(self.dictionary[index].link != -1):
                    index = self.dictionary[index].isLink
                    temp = index
                while(self.dictionary[temp].word != ""):
                    index +=1
                self.dictionary[index].key = word
                self.dictionary[index].meaning = mean
            else:
                while(self.dictionary[index] != ""):
                    index +=1
                self.dictionary[index].word = word
                self.dictionary[index].meaning = mean
        else:
            print("element already exist")
```

```

def insertWithReplacement(self, word, mean):
    if self.searchForNode(word) == False and self.length < self.maxSize:
        self.length += 1
        index = self.hashFunction(word)
        if self.dictionary[index].word == "":
            self.dictionary[index].word = word
            self.dictionary[index].meaning = mean
        elif self.hashFunction(self.dictionary[index].word) ==
self.hashFunction(word):
            while self.dictionary[index].link != -1:
                index = self.dictionary[index].link
            x = index
            while self.dictionary[x].word != "":
                x += 1
            self.dictionary[index].link = x
            self.dictionary[x].word = word
            self.dictionary[x].meaning = mean
        else:
            kw = self.dictionary[index].word
            mn = self.dictionary[index].meaning
            self.dictionary[index].word = word
            self.dictionary[index].meaning = mean
            self.insertWithReplacement(kw, mn)
    else:
        print("HashTable Full")

def printHashtable(self):
    for i in self.dictionary:
        print("word : ", i.word, " -> Meanning : ",i.meaning )

def delete(self, key):
    if self.searchForNode(key) is True:
        count = 0
        index = self.hashFunction(key)
        while self.dictionary[index].word != key:
            index = self.dictionary[index].isLink
        self.dictionary[index].word = ""
        self.dictionary[index].meaning = ""
        x = index-1
        while self.hashFunction(self.dictionary[x].word) !=
self.hashFunction(key) and count<self.length:
            x -= 1
            count += 1
            if x<0:
                x = self.maxSize
            self.dictionary[index] = self.dictionary[x]
            self.dictionary[x].word = ""
            self.dictionary[x].meaning = ""
            self.dictionary[x].isLink = -1

a = HashTable()
while True:
    print("1. Insert WR")
    print("2. Insert WOR")
    print("3. print")
    print("4. delete")
    print("5. cclose")

    ch = int(input("enter the choice : "))
    if(ch ==1):
        word = str(input("enter the word "))
        mean = str(input("enter the meaning "))
        a.insertWithoutReplacement(word, mean)

```

```

elif(ch ==2):
    word = str(input("enter the word "))
    mean = str(input("enter the meaning "))
    a.insertWithReplacement(word, mean)
elif (ch ==3):
    a.printHashtable()
elif (ch ==4):
    word = str(input("enter the word "))
    a.delete(word)
elif(ch == 5):
    break

'''

```

Output

```

1. Insert WR
2. Insert WOR
3. print
4. delete
5. close
enter the choice : 1
enter the word ironman
enter the meaning tony stark
1. Insert WR
2. Insert WOR
3. print
4. delete
5. close
enter the choice : 1
enter the word spiderman
enter the meaning peter parker
1. Insert WR
2. Insert WOR
3. print
4. delete
5. close
enter the choice : 1
enter the word captian america
enter the meaning steve rogers
1. Insert WR
2. Insert WOR
3. print
4. delete
5. close
enter the choice : 1
enter the word hulk
enter the meaning bruce banner
1. Insert WR
2. Insert WOR
3. print
4. delete
5. close
enter the choice : 3
word :      -> Meanning :
word :      -> Meanning :
word :  hulk -> Meanning :  bruce banner
word :      -> Meanning :
word :      -> Meanning :
word :      -> Meanning :
word :      -> Meanning :
word :      -> Meanning :
word :  captian america -> Meanning :  steve rogers
word :  ironman -> Meanning :  tony stark
word :      -> Meanning :

```

```
word :  spiderman  -> Meanning :  peter parker
word :      -> Meanning :
word :      -> Meanning :
word :      -> Meanning :
word :      -> Meanning :
word :      -> Meanning :
'''
```