```python
class Node:
    def __init__(self, key, mean):
        self.key = key
        self.meaning = mean

class Dict:
    def __init__(self):
        self.l = []
        self.chain = []
        self.length = 0
        self.size = 13
        for i in range(13):
            self.l.append(Node("", ""))
            self.chain.append(-1)

    def hashFunction(self, key):
        i = 0
        s = 0
        for _ in key:
            s = s+ord(_)*i
            i += 1
        return s % self.size

    def insertWithoutReplacement(self, key, value):
        if self.search(key)[1] == False and self.length<self.size:
            self.length += 1
            index = self.hashFunction(key)
            if self.l[index].key == "":
                self.l[index].key = key
                self.l[index].meaning = value
            elif self.hashFunction(self.l[index].key) == self.hashFunction(key):
                while self.chain[index] != -1:
                    index = self.chain[index]
                x = index
                while self.l[x].key != "":
                    x += 1
                    if x>self.size:
                        x = 0
                self.chain[index] = x
                self.l[x].key = key
                self.l[x].meaning = value
            else:
                while self.l[index]!= "":
                    #if condn for hash of key at index
                    index += 1
                self.l[index].key = key
                self.l[index].meaning = value
        else:
            print("Element already exists")

    def insertWithReplacement(self, key, value):
        if self.search(key)[1] == False and self.length < self.size:
            self.length += 1
            index = self.hashFunction(key)
            if self.l[index].key == "":
                self.l[index].key = key
                self.l[index].meaning = value
            elif self.hashFunction(self.l[index].key) == self.hashFunction(key):
                while self.chain[index] != -1:
                    index = self.chain[index]
                x = index
                while self.l[x].key != "":
                    x += 1
                    if x>self.size:
                        x = 0
                self.chain[index] = x
                self.l[x].key = key
                self.l[x].meaning = value
            else:
                kw = self.l[index].key
                mn = self.l[index].meaning
                self.l[index].key = key
                self.l[index].meaning = value
                self.length -= 1 #CHECK
                self.insertWithReplacement(kw, mn)
        else:
            print("HashTable Full")

    def search(self, key):
        index = self.hashFunction(key)
        count = 0
        while self.l[index].key != "" and count<=self.length:
            if self.l[index].key == key:
                return count, True
            index = self.chain[index]
            count += 1
        return count, False


    def delete(self, key):
        if self.search(key)[1] == True:
            count = 0
            index = self.hashFunction(key)
            prev  = -1
            while self.l[index].key != key:
                prev = index
                index = self.chain[index]
            self.l[index].key = ""
            self.l[index].meaning = ""
            x = index
            while self.chain[x]!= -1:
                prev = x
                x = self.chain[x]
            if prev == -1:
                pass
            else:
                self.l[index].key, self.l[index].meaning = self.l[x].key, 
self.l[x].meaning
                print(x)
                self.l[x].key, self.l[x].meaning = "", ""
                self.chain[prev] = -1

    def display(self):
        for i in range(len(self.l)):
            print("Entry :", i ,"Keyword:", self.l[i].key,"-> Meaning:", 
self.l[i].meaning, self.chain[i])



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

    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.display()
    elif (ch ==4):
        word = str(input("enter the word "))
        a.delete(word)
    elif(ch == 5):
        break
```