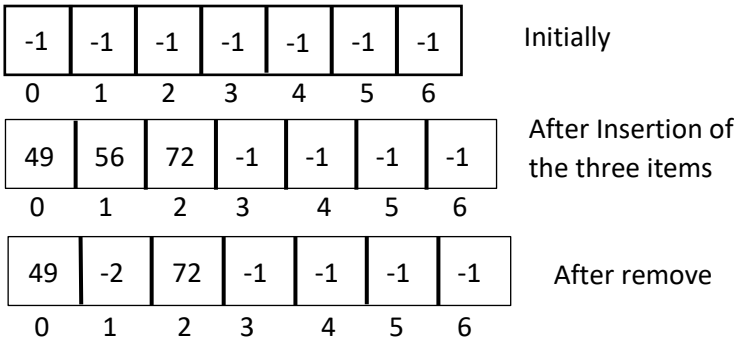
Implementation of Open Addressing in Python

```
h = MyHash(7)
h.insert(49)
h.insert(56)
h.insert(72)
print(h.search(56))
h.remove(56)
print(h.remove(56))
print(h.search(56))
```



O/p: True
False

```
class MyHash:
        def __init__(self, c):
                self.cap = c
                self.table = [-1] * c
                self.size = 0
        def hash(self, x):
                return x % self.cap
        def search(self, x):
                h = self.hash(x)
                t = self.table
                i = h
                while t[i] != -1:
                        if t[i] == x:
                                return True
                        i = (i + 1) % self.cap
                        if i == h:
                                return False
                return False
        def insert(self, x):
                if self.size == self.cap:
                        return False
                if self.search(x) == True:
                        return False
                i = self.hash(x)
                t = self.table
                while t[i] not in (-1, -2):
                        i = (i + 1) % self.cap
                t[i] = x
                self.size = self.size + 1
                return True
        def remove(self, x):
                h = self.hash(x)
                t = self.table
                i = h
                while t[i] != -1:
                        if t[i] == x:
                                t[i] = -2
                                return True
                        i = (i + 1) % self.cap
                        if i == h:
                                return False
                return False
```