⇒ Hashmap

↳ Syntax :- `HashMap< keyDataType , valueDataType> map = new HashMap<>();`

⇒ put and remove functions of HM

```
public static void main(String[] args) {
    HashMap<String, Integer> map = new HashMap<>();
    // add elements as pair of key, value
    map.put( "Bharat", 350 );   // key, value
    map.put("SriLanka", 250);
    map.put("Australia", 300);
    map.put("Pak", 200);
    map.put("NZ", 175);
    System.out.println(map);

    // remove element from HM
    int score = map.remove("NZ");   // key
    System.out.println(map);
    System.out.println(score);
}
```

→ paramters

→ parameter (remove fn will return value)

⇒ Replace function in HM

```
// map.replace("Australia", 310);
map.put("Australia", 310);
System.out.println(map);
```

(work same as put)

⇒ containsKey and containsValue

```
System.out.println(map.containsKey("Bharat"));
System.out.println(map.containsKey("ABC"));

System.out.println(map.containsValue(310));
System.out.println(map.containsValue(10));
```

Note:-

We can't have duplicate key in HM but we can have duplicate value

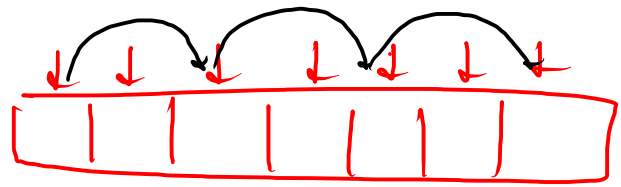
⇒ `map.isEmpty();` // check if HM is empty or not

⇒ `map.size();` // returns the size of HM

⇒ get some value for particular key

↳ map.get(key);

⇒ Iteration in HM



arr

[for(int i : arr)
    syso(i);

```
for (Map.Entry<String, Integer> entry : map.entrySet()) {
    String key = entry.getKey();
    Integer val = entry.getValue();
    System.out.println(key + ", " + val);
}
```

{ BL = 500 }

→ keySet and values

```
for (String s : map.keySet()) {
    System.out.println(s);
}

for (Integer i : map.values()) {
    System.out.println(i);
}
```

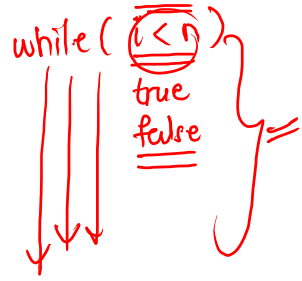
→ provides set of key

→ provides set of values

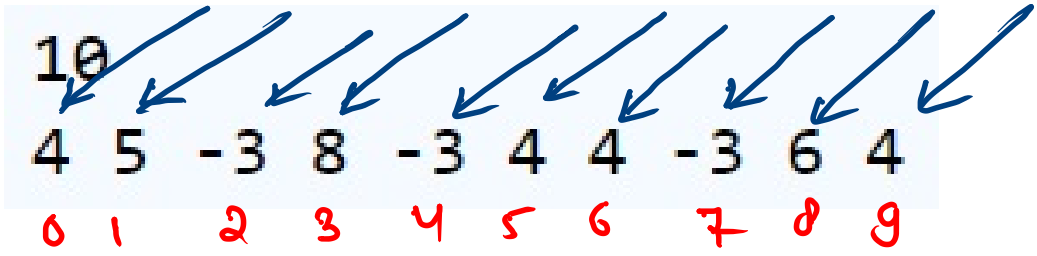
Code

∞ loop

```java
public class Solution {
    public static HashMap<String, String> map;
    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);

        map = new HashMap<>();

        while (true) {
            int n = scn.nextInt();
            if (n == 1) {
                String word = scn.next();
                String meaning = scn.next();
                addElement(word, meaning);
            } else if (n == 2) {
                String word = scn.next();
                printMeaning(word);
            } else if (n == 3) {
                String word = scn.next();
                removeWord(word);
            } else if (n == 4) {
                break;
            }
        }
    }

    public static void addElement(String word, String meaning) {
        map.put(word, meaning);
    }
    public static void printMeaning(String word) {
        String meaning = map.get(word);
        if (meaning == null) {
            System.out.println(-1);
            return;
        }
        System.out.println(meaning);
        // System.out.println(map.get(word) == null ? -1 : map.get(word));
    }
    public static void removeWord(String word) {
        map.remove(word);
    }
}
```



Same Number Same Frequency



check

num = |4| = 4
freq = 4

value vs freq

HM

4 → ~~1~~ ~~2~~ ~~3~~ 4

5 → 1

-3 → ~~1~~ ~~2~~ 3

8 → 1

6 → 1

map

```
{-3=3, 4=4, 5=1, 6=1, 8=1}
```

num = |-3| = 3
freq = 3

Code

```java
public static void SNSF(ArrayList<Integer> arr) {
    HashMap<Integer, Integer> map = new HashMap<Integer, Integer>();
    for (int i = 0; i < arr.size(); i++) {
        int num = arr.get(i);
        if ( map.containsKey(num) ) {
            int freq = map.get(num);
            map.put( num, freq + 1 );
        } else {
            map.put( num, 1 );
        }
    }

    ArrayList<Integer> ans = new ArrayList<>();
    for (Map.Entry<Integer, Integer> entry: map.entrySet()) {
        int num = entry.getKey();
        int freq = entry.getValue();
        if ( Math.abs(num) == freq ) {
            ans.add(num);
        }
    }

    Collections.sort(ans);
    for (int i : ans) {
        System.out.println(i);
    }
}
```

N

O(N)

O(1)

O(N)

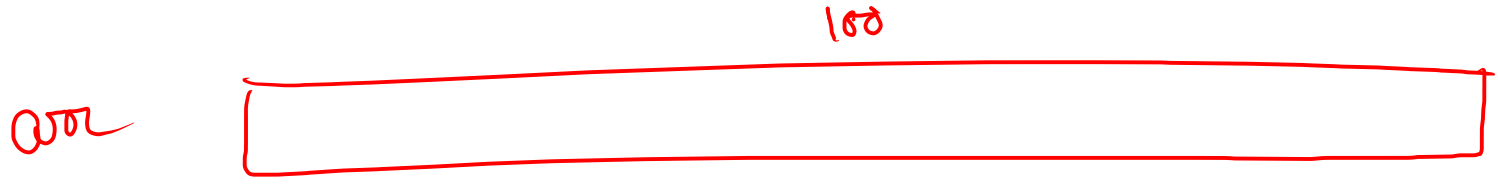
$O(N^2)$

K<<<N

K

(K log K)

Time Complexity of all HM functions are amotised O(1)



O(1) → 1

amotised O(1) → 3 or 4