

# CSCI 232:

# Data Structures and Algorithms

More HashMap Practice

Reese Pearsall  
Spring 2025

# Announcements

Program 1 due **tonight** at 11:59 PM

Lab 5 due tomorrow

No class next Tuesday (Reese is gone)

First Quiz is next friday



# Program 2

# LeetCode #383 Ransomnote

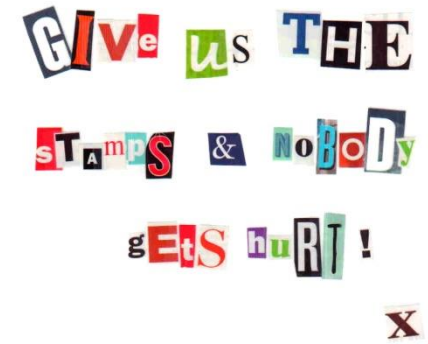
Given two strings **ransomNote** and **magazine**, return true if **ransomNote** can be constructed by using the letters from **magazine** and false otherwise.

ransomNote = "hello"

magazine = "there is a hill down there."



# LeetCode #383 Ransomnote



Given two strings **ransomNote** and **magazine**, return true if **ransomNote** can be constructed by using the letters from **magazine** and false otherwise.

ransomNote = "hello"

magazine = "there is a hill down there."

## 1. Make Character Frequency Table

e: 4

h: 3

t: 2

r: 2

i: 2

l: 2

s: 1

a: 1

d: 1

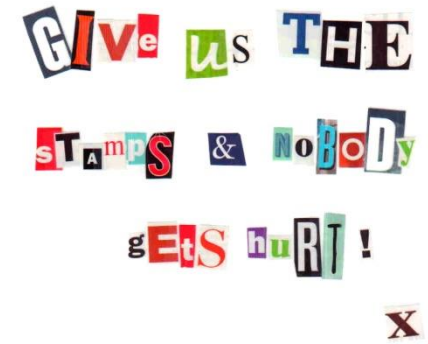
o: 1

w: 1

n: 1

HashMap<Character,Integer>

# LeetCode #383 Ransomnote



Given two strings **ransomNote** and **magazine**, return true if **ransomNote** can be constructed by using the letters from **magazine** and false otherwise.

ransomNote = "hello"

magazine = "there is a hill down there."

1. Make Character Frequency Table

2. Iterate through ransomNote and see if it can be created

e: 4 ~~3~~  
h: 1 ~~2~~  
t: 2  
r: 2  
i: 2  
l: 2 ~~1~~ ~~0~~ HashMap<Character,Integer>  
s: 1  
a: 1  
d: 1  
o: 1 ~~0~~  
w: 1  
n: 1

ransomNote = "hello"  
                  ✓✓✓✓✓

All characters exist, and never went below zero! → return true

# LeetCode #12 Integer to Roman

Given an integer, convert it to a Roman numeral.

HashMap<Integer, String>

M	CM	D	CD	C	XC	L	XL	X	IX	V	IV	I
1000	900	500	400	100	90	50	40	10	9	5	4	1

N = 48

40 XL  
5 V  
3 I  
8 I

Final Answer = XLVIII

# LeetCode #12 Integer to Roman

Given an integer, convert it to a Roman numeral.

HashMap<Integer, String>

	M	CM	D	CD	C	XC	L	XL	X	IX	V	IV	I
numberList:	1000	900	500	400	100	90	50	40	10	9	5	4	1

↑  
i

HashMaps are not ordered, so we have to put numbers into a List and sort it!

```
List<Integer> numberList = new LinkedList<>(romanMap.keySet());
Collections.sort(numberList, Collections.reverseOrder());
```




# LeetCode #12 Integer to Roman

Given an integer, convert it to a Roman numeral.

HashMap<Integer, String>

	M	CM	D	CD	C	XC	L	XL	X	IX	V	IV	I
numberList:	1000	900	500	400	100	90	50	40	10	9	5	4	1

  
i

**N = 59**

```
String answer = "";
int i = 0;
while(n > 0) {

    if(numberList.get(i) > n) {
        i++;
    }

    if(n - numberList.get(i) >= 0) {
        String numeral = romanMap.get(numberList.get(i));
        answer += numeral;
        n -= numberList.get(i);
    }
}
return answer;
```


Answer =

# LeetCode #12 Integer to Roman

Given an integer, convert it to a Roman numeral.

numberList:

HashMap<Integer, String>												
M	CM	D	CD	C	XC	L	XL	X	IX	V	IV	I
1000	900	500	400	100	90	50	40	10	9	5	4	1

  
i

**N = 59**

```
String answer = "";
int i = 0;
while(n > 0) {
    if(numberList.get(i) > n) {
        i++;
    }
    if(n - numberList.get(i) >= 0) {
        String numeral = romanMap.get(numberList.get(i));
        answer += numeral;
        n -= numberList.get(i);
    }
}
return answer;
```


Answer =

# LeetCode #12 Integer to Roman

Given an integer, convert it to a Roman numeral.

numberList:

HashMap<Integer, String>												
M	CM	D	CD	C	XC	L	XL	X	IX	V	IV	I
1000	900	500	400	100	90	50	40	10	9	5	4	1

  
i

**N = 59**

```
String answer = "";
int i = 0;
while(n > 0) {
    if(numberList.get(i) > n) {
        i++;
    }
    if(n - numberList.get(i) >= 0) {
        String numeral = romanMap.get(numberList.get(i));
        answer += numeral;
        n -= numberList.get(i);
    }
}
return answer;
```


Answer =

# LeetCode #12 Integer to Roman

Given an integer, convert it to a Roman numeral.

numberList:

HashMap<Integer, String>												
M	CM	D	CD	C	XC	L	XL	X	IX	V	IV	I
1000	900	500	400	100	90	50	40	10	9	5	4	1

  
i

**N = 59**

```
String answer = "";
int i = 0;
while(n > 0) {
    if(numberList.get(i) > n) {
        i++;
    }
    if(n - numberList.get(i) >= 0) {
        String numeral = romanMap.get(numberList.get(i));
        answer += numeral;
        n -= numberList.get(i);
    }
}
return answer;
```


Answer =

# LeetCode #12 Integer to Roman

Given an integer, convert it to a Roman numeral.

numberList:

HashMap<Integer, String>												
M	CM	D	CD	C	XC	L	XL	X	IX	V	IV	I
1000	900	500	400	100	90	50	40	10	9	5	4	1

  
i

N = 59

```
String answer = "";
int i = 0;
while(n > 0) {
    if(numberList.get(i) > n) {
        i++;
    }

    if(n - numberList.get(i) >= 0) {
        String numeral = romanMap.get(numberList.get(i));
        answer += numeral;
        n -= numberList.get(i);
    }
}
return answer;
```


Answer =

# LeetCode #12 Integer to Roman

Given an integer, convert it to a Roman numeral.

numberList:

HashMap<Integer, String>												
M	CM	D	CD	C	XC	L	XL	X	IX	V	IV	I
1000	900	500	400	100	90	50	40	10	9	5	4	1

  
i

N = 59

```
String answer = "";
int i = 0;
while(n > 0) {
    if(numberList.get(i) > n) {
        i++;
    }
    if(n - numberList.get(i) >= 0) {
        String numeral = romanMap.get(numberList.get(i));
        answer += numeral;
        n -= numberList.get(i);
    }
}
return answer;
```


Answer =

# LeetCode #12 Integer to Roman

Given an integer, convert it to a Roman numeral.

numberList:

HashMap<Integer, String>												
M	CM	D	CD	C	XC	L	XL	X	IX	V	IV	I
1000	900	500	400	100	90	50	40	10	9	5	4	1

  
i

**N = 59**

```
String answer = "";
int i = 0;
while(n > 0) {
    if(numberList.get(i) > n) {
        i++;
    }
    if(n - numberList.get(i) >= 0) {
        String numeral = romanMap.get(numberList.get(i));
        answer += numeral;
        n -= numberList.get(i);
    }
}
return answer;
```


Answer =

# LeetCode #12 Integer to Roman

Given an integer, convert it to a Roman numeral.

numberList:

HashMap<Integer, String>												
M	CM	D	CD	C	XC	L	XL	X	IX	V	IV	I
1000	900	500	400	100	90	50	40	10	9	5	4	1

  
i

N = 59

```
String answer = "";
int i = 0;
while(n > 0) {
    if(numberList.get(i) > n) {
        i++;
    }
    if(n - numberList.get(i) >= 0) {
        String numeral = romanMap.get(numberList.get(i));
        answer += numeral;
        n -= numberList.get(i);
    }
}
return answer;
```

Answer =




# LeetCode #12 Integer to Roman

Given an integer, convert it to a Roman numeral.

numberList:

HashMap<Integer, String>												
M	CM	D	CD	C	XC	L	XL	X	IX	V	IV	I
1000	900	500	400	100	90	50	40	10	9	5	4	1

  
i

**N = ~~59~~  
9**

```
String answer = "";
int i = 0;
while(n > 0) {

    if(numberList.get(i) > n) {
        i++;
    }

    if(n - numberList.get(i) >= 0) {
        String numeral = romanMap.get(numberList.get(i));
        answer += numeral;
        n -= numberList.get(i);
    }
}
return answer;
```


Answer = L

# LeetCode #12 Integer to Roman

Given an integer, convert it to a Roman numeral.

numberList:

HashMap<Integer, String>												
M	CM	D	CD	C	XC	L	XL	X	IX	V	IV	I
1000	900	500	400	100	90	50	40	10	9	5	4	1

  
i

N = ~~59~~  
9

```
String answer = "";
int i = 0;
while(n > 0) {

    if(numberList.get(i) > n) {
        i++;
    }

    if(n - numberList.get(i) >= 0) {
        String numeral = romanMap.get(numberList.get(i));
        answer += numeral;
        n -= numberList.get(i);
    }
}
return answer;
```


Answer = L

# LeetCode #12 Integer to Roman

Given an integer, convert it to a Roman numeral.

numberList:

HashMap<Integer, String>												
M	CM	D	CD	C	XC	L	XL	X	IX	V	IV	I
1000	900	500	400	100	90	50	40	10	9	5	4	1

  
i

N = ~~59~~  
9

```
String answer = "";
int i = 0;
while(n > 0) {

    if(numberList.get(i) > n) {
        i++;
    }

    if(n - numberList.get(i) >= 0) {
        String numeral = romanMap.get(numberList.get(i));
        answer += numeral;
        n -= numberList.get(i);
    }
}
return answer;
```

Answer = L

# LeetCode #12 Integer to Roman

Given an integer, convert it to a Roman numeral.

numberList:

HashMap<Integer, String>												
M	CM	D	CD	C	XC	L	XL	X	IX	V	IV	I
1000	900	500	400	100	90	50	40	10	9	5	4	1



i

**N = 59**  
**9**

```
String answer = "";
int i = 0;
while(n > 0) {

    if(numberList.get(i) > n) {
        i++;
    }

    if(n - numberList.get(i) >= 0) {
        String numeral = romanMap.get(numberList.get(i));
        answer += numeral;
        n -= numberList.get(i);
    }
}
return answer;
```

Answer = L

# LeetCode #12 Integer to Roman

Given an integer, convert it to a Roman numeral.

numberList:

HashMap<Integer, String>												
M	CM	D	CD	C	XC	L	XL	X	IX	V	IV	I
1000	900	500	400	100	90	50	40	10	9	5	4	1



**N = 59**  
**9**  
**0**

```
String answer = "";
int i = 0;
while(n > 0) {

    if(numberList.get(i) > n) {
        i++;
    }

    if(n - numberList.get(i) >= 0) {
        String numeral = romanMap.get(numberList.get(i));
        answer += numeral;
        n -= numberList.get(i);
    }
}
return answer;
```

Answer = LIX

# LeetCode #12 Integer to Roman

Given an integer, convert it to a Roman numeral.

numberList:

HashMap<Integer, String>												
M	CM	D	CD	C	XC	L	XL	X	IX	V	IV	I
1000	900	500	400	100	90	50	40	10	9	5	4	1



**N = 59**  
**9**  
**0**

```
String answer = "";
int i = 0;
while(n > 0) {

    if(numberList.get(i) > n) {
        i++;
    }

    if(n - numberList.get(i) >= 0) {
        String numeral = romanMap.get(numberList.get(i));
        answer += numeral;
        n -= numberList.get(i);
    }
}
return answer;
```

**Answer = LIX**

# LeetCode #12 Integer to Roman

Given an integer, convert it to a Roman numeral.

HashMap<Integer, String>

	M	CM	D	CD	C	XC	L	XL	X	IX	V	IV	I
numberList:	1000	900	500	400	100	90	50	40	10	9	5	4	1

HashMaps **do not** maintain order

TreeMaps **do** maintain order