

Logic-1: shareDigit

Given two ints, each in the range 10-99, return true if there is a digit that appears in both numbers, such as the 2 in 12 and 23.

Note: $(n / 10) \% 10$, gives the left digit (in the 10s column).

$n \% 10$ gives the right digit (in the 1s column).

```
public boolean shareDigit(int a, int b) {  
  
}
```

Step 1

Start by separating out all four digits from a and b.

The **ones** digit of integer **a**.

The **tens** digit of integer **a**.

The **ones** digit of integer **b**.

The **tens** digit of integer **b**.

```
public boolean shareDigit(int a, int b) {  
    int digit1A = a % 10;  
    int digit10A = (a / 10) % 10;  
    int digit1B = b % 10;  
    int digit10B = (b / 10) % 10;  
}
```

Example: a = 58 b = 83

int digit1A = 8

int digit10A = 5

int digit1B = 3

int digit10B = 8

Step 2

We now declare & initialize our return variable, and return it..

```
public boolean shareDigit(int a, int b) {  
    int digit1A = a % 10;  
    int digit10A = (a / 10) % 10;  
    int digit1B = b % 10;  
    int digit10B = (b / 10) % 10;  
    boolean share = false;  
    return share;  
}
```

Step 3

We now start to compare each digit in **a** with each digit in **b**.

First, we'll compare the **ONES** digit of **a** with the **ONES** digit of **b**.

```
public boolean shareDigit(int a, int b) {  
    int digit1A = a % 10;  
    int digit10A = (a / 10) % 10;  
    int digit1B = b % 10;  
    int digit10B = (b / 10) % 10;  
    boolean share = false;  
    if (digit1A == digit1B) {  
        share = true;  
    }  
    return share;  
}
```

a = 58 b = 83



Step 4

Second: Compare the **ONES** digit of **a** with the **TENS** digit of **b**.

```
public boolean shareDigit(int a, int b) {
    int digit1A = a % 10;
    int digit10A = (a / 10) % 10;
    int digit1B = b % 10;
    int digit10B = (b / 10) % 10;
    boolean share = false;
    if (digit1A == digit1B) {
        share = true;
    }
    if (digit1A == digit10B) {
        share = true;
    }
    return share;
}
```

a = 58 b = 83
↑ ↑

Step 5

Third: Compare the **TENS** digit of **a** with the **ONES** digit of **b**.

```
public boolean shareDigit(int a, int b) {
    int digit1A = a % 10;
    int digit10A = (a / 10) % 10;
    int digit1B = b % 10;
    int digit10B = (b / 10) % 10;
    boolean share = false;
    if (digit1A == digit1B) {
        share = true;
    }
    if (digit1A == digit10B) {
        share = true;
    }
    if (digit10A == digit1B) {
        share = true;
    }
    return share;
}
```

a = 58 b = 83
↑ ↑

Step 6

Third: Compare the **TENS** digit of **a** with the **TENS** digit of **b**.
DONE!

```
public boolean shareDigit(int a, int b) {
    int digit1A = a % 10;
    int digit10A = (a / 10) % 10;
    int digit1B = b % 10;
    int digit10B = (b / 10) % 10;
    boolean share = false;
    if (digit1A == digit1B) {
        share = true;
    }
    if (digit1A == digit10B) {
        share = true;
    }
    if (digit10A == digit1B) {
        share = true;
    }
    if (digit10A == digit10B) {
        share = true;
    }
    return share;
}
```

a = 58 b = 83
↑ ↑