

# ⇒ Digit Traversal

0 1 2 3 4 5

int  $n = 123456$

idx = 12  
ans = 23

$$\Rightarrow n / 1000 \Rightarrow \underline{\underline{123}}$$

$$\Rightarrow n \% 100 \Rightarrow \underline{\underline{23}}$$

$n = 123$  ↙

$$n \% 10 = 3$$

$$n \% 100 = \underline{\underline{23}}$$

~~1234567~~

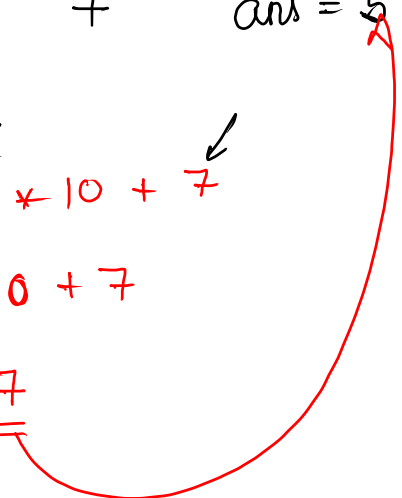
$$n = \underline{\underline{n / 10000}}$$

Given x and y, print xy

↳ x = 5, y = 7

o/p = 57

$$\begin{aligned} \text{ans} &= x * 10 + y \\ &= 50 + 7 \\ &= 57 \end{aligned}$$

$$\begin{array}{l} x \quad y \\ 5, \quad 7 \end{array} \quad \text{ans} = 57$$
  
$$\begin{aligned} &= 5 * 10 + 7 \\ &= 50 + 7 \\ &= \underline{\underline{57}} \end{aligned}$$


i/p)

t = 5

1) 9 ,

2) 3 ,

3) 5 ,

4) 0 ,

5) 7 ,

ans = 93507

ans = 0

$$\text{ans} = \text{ans} * 10 + 9 = 9$$

$$\text{ans} = \text{ans} * 10 + 3 = 93$$

$$\text{ans} = \text{ans} * 10 + 5 = 935$$

$$\text{ans} = \text{ans} * 10 + 0 = 9350$$

$$\text{ans} = \text{ans} * 10 + 7 = 93507$$

# Given x and y, print xy

```
public static void main(String[] args) {  
    Scanner scn = new Scanner(System.in);  
    int t = scn.nextInt();  
    for (int i = 0; i < t; i++) {  
        int x = scn.nextInt();  
        int y = scn.nextInt();  
  
        int ans = printxy(x, y);  
        System.out.println(ans);  
    }  
}  
  
public static int printxy(int x, int y) {  
    int ans = x * 10 + y;  
    return ans;  
}
```

Logic



# Print digit by digit of a three digit number

```
public static void main(String[] args) {  
    Scanner scn = new Scanner(System.in);  
    int n = scn.nextInt();  
  
    printDigitsFromLast(n);  
}  
public static void printDigitsFromLast(int n) {  
    while ( n > 0 ) {  
        int ans = n % 10;  
        System.out.println(ans);  
        n = n / 10;  
    }  
}
```

---

# Reverse a 3 digit number

↳ i/p :-  $n = 123$

↳ o/p :-  $ans = 321$

Note :- must create function and return int value

$n = 123$

,

$ans = 0$

step 1)

$int \text{ rem} = n \% 10$   
(3)

,  $ans = ans * 10 + rem$ ,  $n /= 10$   
(3) (12)

step 2)

$int \text{ rem} = n \% 10$   
(2)

,  $ans = ans * 10 + rem$ ,  $n /= 10$   
(32) (1)

step 3)

$int \text{ rem} = n \% 10$   
(1)

,  $ans = ans * 10 + rem$ ,  $n /= 10$   
(321) (0)

code

```
public static void main(String[] args) {  
    Scanner scn = new Scanner(System.in);  
    int t = scn.nextInt();  
    for (int i = 0; i < t; i++) {  
        int n = scn.nextInt();  
        System.out.println( reverse3DigitNumber(n) );  
    }  
}  
public static int reverse3DigitNumber(int n) {  
    int ans = 0;  
    while ( n > 0 ) {  
        int rem = n % 10;  
        ans = ans * 10 + rem;  
  
        n = n / 10;  
    }  
    return ans;  
}
```

# Reverse n-digit number

i/p    n = 5    , num = 0

<u>val</u> → 5 ,	num = num * 10 + val = 5
→ 2 ,	num = num * 10 + val = 52
→ 0 ,	num = num * 10 + val = 520
→ 7 ,	num = num * 10 + val = 5207
→ 0 ,	num = num * 10 + val = 52070



```
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt(); // no of digits
    int num = 0;
    for (int i = 0; i < n; i++) {
        int val = scn.nextInt();
        num = num * 10 + val;
    }
    System.out.println(num);

    int ans = reverseNumber(num);
    System.out.println(ans);
}

public static int reverseNumber(int n) {
    int ans = 0;
    while ( n > 0 ) {
        int rem = n % 10;
        ans = ans * 10 + rem;

        n = n / 10;
    }
    return ans;
}
```

Write a function to check if an Armstrong number or not

i/p

$$n = \underline{153}$$

count of digits = 3

$$\begin{aligned} \text{ans} &= \underline{(1)^3 + (5)^3 + (3)^3} \\ &= 1 + 125 + 27 \\ &= 153 \end{aligned}$$

$$n = 1634$$

count = 4

$$\begin{aligned} \text{ans} &= (1)^4 + (6)^4 + (3)^4 + (4)^4 \\ &= 1634 \end{aligned}$$

code

```
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int t = scn.nextInt();
    for (int i = 0; i < t; i++) {
        int n = scn.nextInt();
        boolean ans = armstrongNumber(n);
        System.out.println(ans);
    }
}

public static boolean armstrongNumber(int n) { // 153
    int temp = n;
    int ans = 0;
    while ( n > 0 ) {
        int rem = n % 10;
        ans = ans + ( rem * rem * rem ); // Math.pow(rem , count)
        n /= 10;
    }
    return (ans == temp);
}
```