Print the final number xyzw...

n = 5

9 → 9

7 → 97

6 → 976

1 → 9761

0 → 97610

```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();

    int ans = 0;
    while (n-- > 0) {
        int num = scn.nextInt();
        ans = ans * 10 + num;
    }

    System.out.println(ans);
}
```

5

ans = 0

int n = 5;

9 → ans = 0*10+9
ans = 9

6 → ans = 9*10+6
= 90+6
= 96

1 → ans = 96*10+1
= 960+1
= 961

0 → ans = 961*10+0
= 9610+0
= 9610

5 → ans = 9610*10+5
= 96100+5
= 96105

```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();

    int ans = 0;
    while (n-- > 0) {
        int num = scn.nextInt();
        ans = ans * 10 + num;
    }

    System.out.println(ans);
}
```

n = 5

ans = 0 ✓

num = 5 , ans = 0 * 10 + 5
= 5

num = 7 , ans = 5 * 10 + 7
= 57

num = 1 , ans = 57 * 10 + 1
= 571

num = 3 , ans = 571 * 10 + 3
= 5713

num = 0 , ans = 5713 * 10 + 0
= 57130

Reverse n-digit number

↳ n = 5

int ans = 0

num = 12345 , rem = 12345 % 10 = 5 , num = 12345 / 10 = 1234
ans = ans * 10 + 5 = 5

num = 1234 , rem = 1234 % 10 = 4 , num = 1234 / 10 = 123
ans = ans * 10 + 4 = 54

num = 123 , rem = 123 % 10 = 3 , num = 123 / 10 = 12
ans = ans * 10 + 3 = 543

num = 12 , rem = 12 % 10 = 2 , num = 12 / 10 = 1
ans = ans * 10 + 2 = 5432

num = 1 , rem = 1 % 10 = 1 , num = 1 / 10 = 0
ans = ans * 10 + 1 = 54321

Code

```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();

    int num = 0;
    for (int i = 0; i < n; i++) {
        int val = scn.nextInt();
        num = num * 10 + val;
    }
    System.out.println(num);
    int result = reverse(num, n);
    System.out.println(result);
}

public static int reverse(int num, int n) {
    int ans = 0;
    while (num > 0) {
        int rem = num % 10;
        num = num / 10;

        ans = ans * 10 + rem;
    }
    return ans;
}
```

Divide n by 2 3 5 and tell steps

n = 3300

2	3300
2	1650
3	825
5	275
5	55
	11

ans = 5

```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int count = scn.nextInt();

    while (n % 2 == 0) {
        n = n / 2;
        count += 2;
    }

    while (n % 3 == 0) {
        n = n / 3;
        count += 3;
    }

    while (n % 5 == 0) {
        n = n / 5;
        count += 5;
    }

    System.out.println(count);
    System.out.println(n);
}
```

⇒ Armstrong number

num = 371

count of digit = 3

$$ans = 3^3 + 7^3 + 1^3$$

$$= (3\times3\times3) + (7\times7\times7) + (1\times1\times1)$$

$$= 27 + 343 + 1$$

$$= 371$$

num = 8

count = 1

$ans = 8^1 = 8$

num = 1634

Count = 4

ans = $1^4 + 6^4 + 3^4 + 4^4$

= 1 + 1296 + 81 + 256

ans = 1634

123

↳ count = 3

ans = $1^3 + 2^3 + 3^3$

= 1 + 8 + 27

= 36

Write a function to check if an Armstrong number or not

t = 2

num → 123

→ 371

temp = num;
num = 371

used to count the digits

```
while (num > 0){
    num = num / 10;
    count++;
}
```

a^b

used to calculate the answer

```
while (temp > 0) {
    int rem = temp % 10;
    temp = temp / 10;
    ans = ans + (int)(Math.pow(rem, count));
}
```

```
if (ans == temp)  print (D)
else              print (NA)
```