Armstrong Number:

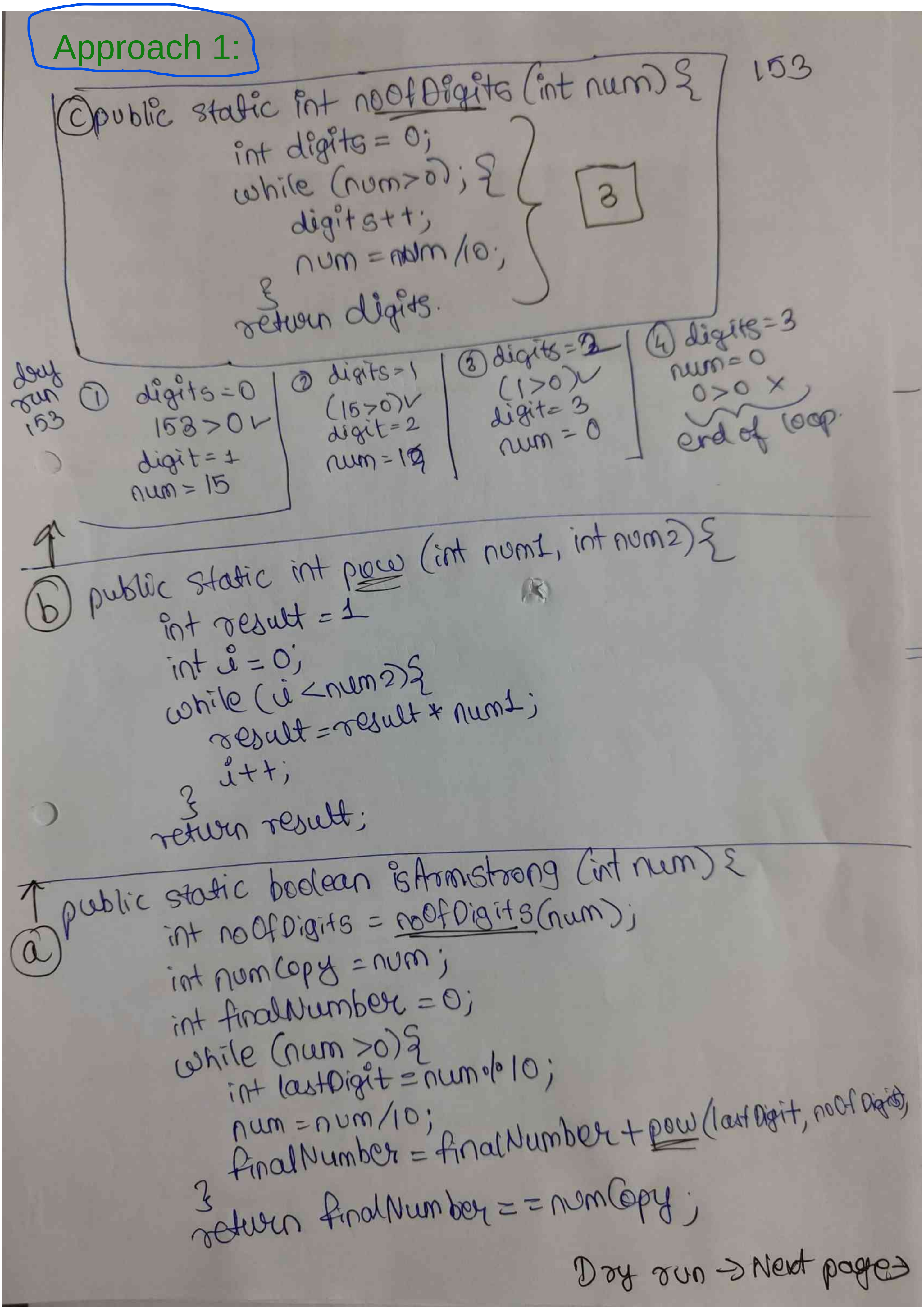
→ Eg: 153 ↓ 3 digits } → $1^3 + 5^3 + 3^3 = 1 + 125 + 27 = \underline{153}$

Approach 2: →

```
class ArmStrongNumber {
    public static void main (String[] args){
        int n = 153;
        int temp = n;
        int arm = 0;
        while (n>0) {
          int ld = n % 10;  // ld => LastDigit
          n = n/10
          arm = arm + (ld * ld * ld);
        }
        if (temp == arm) {
            sout ("Armstrong num");
        }
        else {
          sout ("Not Armstrong num");
        }
```

Dry run:
① n=153
153>0
ld = 3
n = 15
arm = 0 + (3×3×3)
arm = 27
	② n=15
	15>0
	ld = 5
	n = 1
	arm = 27 + (125)
	arm = 152
		③ n = 1
		1>0
		ld = 1
		n = 0
		arm = 152 + (1)
		arm = 153
			④ n=0
			0>0 ✗
			↓
			end of loop
			⑤ temp == arm
			↓
			Armstrong ✓

Approach 1:



153

```
Ⓒpublic static int noOfDigits (int num){
        int digits = 0;
        while (num>0); {
            digits++;
            num = num/10;
        }
        return digits.
```

3

dry run 153

① digits=0
153>0 ✓
digit=1
num=15

② digits=1
(15>0)✓
digit=2
num=1

③ digits=2
(1>0)✓
digit=3
num=0

④ digits=3
num=0
0>0 ✗
end of loop

```
ⓑ public static int pow (int num1, int num2){
        int result = 1
        int i = 0;
        while (i <num2){
            result = result * num1;
            i++;
        }
        return result;
```

```
ⓐ public static boolean isArmstrong (int num){
        int noOfDigits = noOfDigits(num);
        int numCopy = num;
        int finalNumber = 0;
        while (num >0){
            int lastDigit = num%10;
            num = num/10;
            finalNumber = finalNumber + pow(lastDigit, noOfDigits);
        }
        return finalNumber == numCopy;
```

Dry run → Next page →

Dry run → 153 → num

① noOfDigits = 3

numCopy = 153;

finalNumber = 0

153 > 0 ✓

lastDigit = 3

num = 15

finalNumber = $\underbrace{0 + pow(3, 3)}$

= 0 + 27

= 27

② noOfDigits = 3

numCopy = 153

finalNumber = 27

15 > 0 ✓

lastDigit = 5

num = 1

finalNumber = 27 + pow(5, 3)

= 27 + 125

= 152

③ noOfDigits = 3

numCopy = 153

finalNumber = 152

1 > 0 ✓

lastDigit = 1

num = 0

dry run → next page.

finalNumber = 152 + pow(1, 3)

= 152 + 1

= 153

num = 0 } → end of loop:

finalNumber == numCopy

153 == 153 Armstrong

pow (num1 = 3, num2 = 3)

① result = 1

i = 0

(0 < 3) ✓

result = 1 * 3 = 3

i = 1

② result = 3

i = 1

1 < 3

result = 3 * 3 = 9

i = 2

③ result = 9

i = 2

2 < 3

result = 9 * 3 = 27

i = 3

3 < 3 ✗ end of loop

∴ result = $\boxed{27}$

① pow(num1 = 5, num2 = 3)

result = 1

i = 0

(0 < 3) ✓

result = 1 * 5 = 5

i = 1

② result = 5

i = 1

(1 < 3)

result = 5 * 5 = 25

i = 2

③ result = 25

i = 2

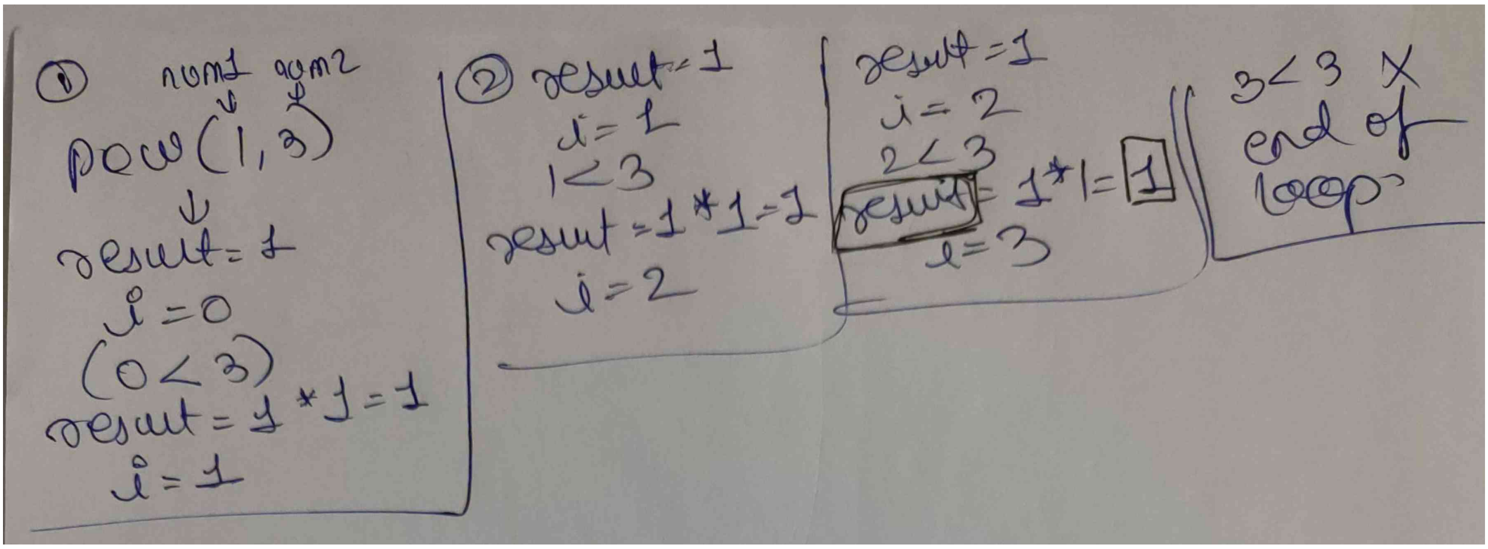
(2 < 3)

result = 25 * 5 = $\boxed{125}$

i = 3

(3 < 3) ✗

end of loop.



① $\overset{\text{num1}}{\downarrow}$ $\overset{\text{num2}}{\downarrow}$

pow(1, 3)

↓

result = 1

i = 0

(0 < 3)

result = 1 * 1 = 1

i = 1

② result = 1

i = 1

1 < 3

result = 1 * 1 = 1

i = 2

result = 1

i = 2

2 < 3

$\boxed{\text{result}}$ = 1 * 1 = $\boxed{1}$

i = 3

3 < 3 X

end of loop.