

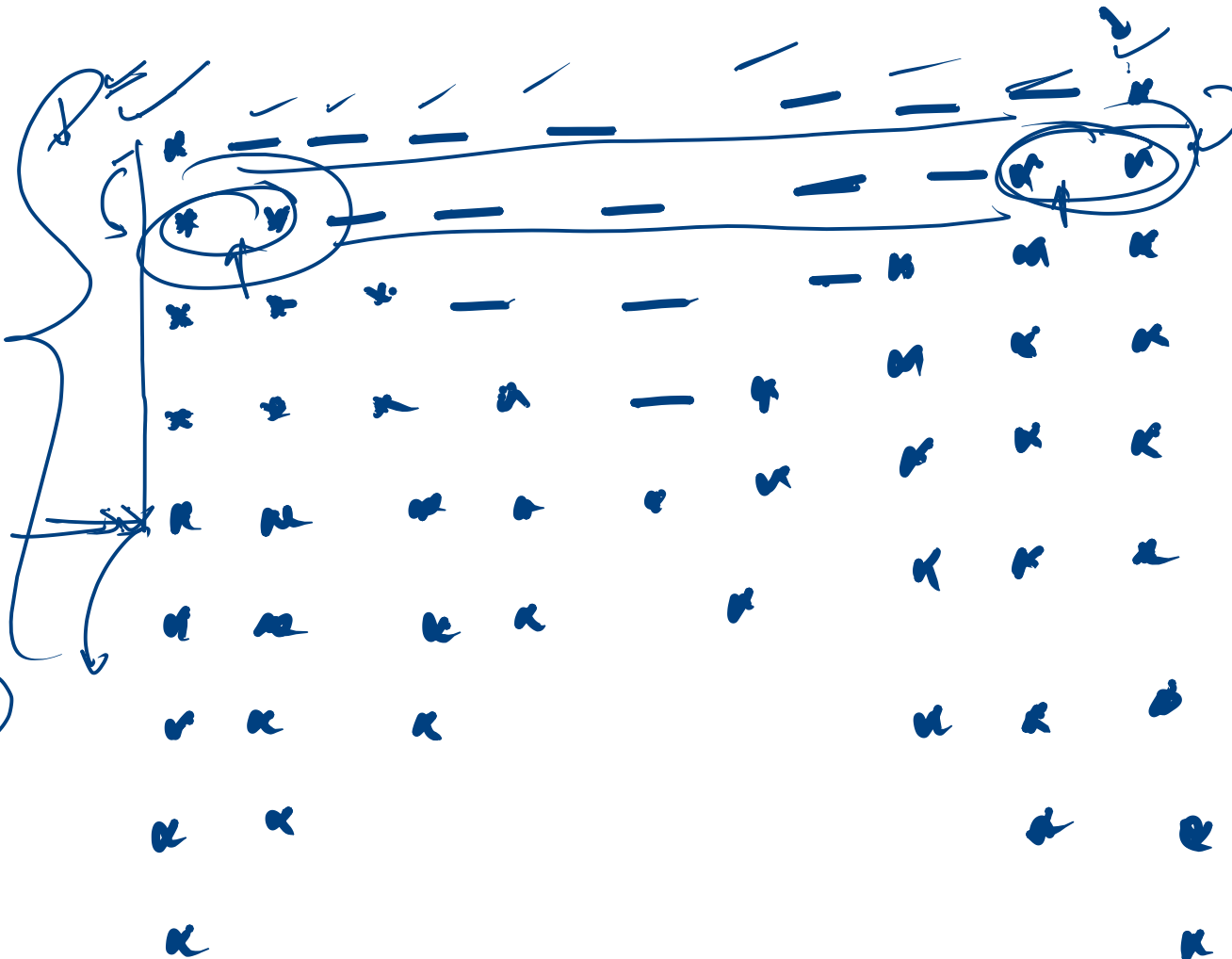
$$n_{\text{os}} = 2n - 1$$

$$n_{\text{st}} = 1$$

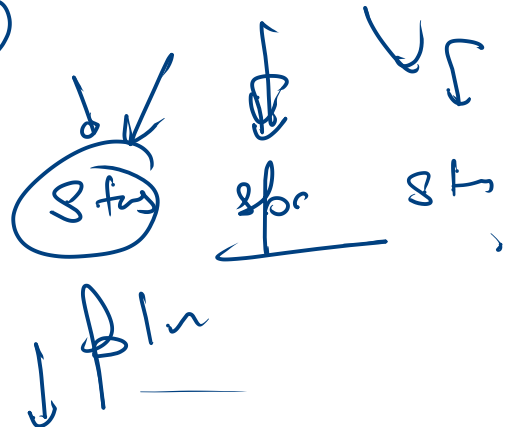
$$n_{\text{sp}} = 2n - 3$$

$$n_{\text{st}} + +$$

$$n_{\text{sp}} - = 2$$



$$2n - 1 - 2$$



$$n_{\text{st}} - -$$

$$n_{\text{sp}} + = 2$$

$$n \leq 5$$

$$n \leq n$$

```

public class Main {
    public static void main(String[] args) {
        int n = 5;
        int nsp = 2 * n - 3;
        int nst = 1;
        for(int r = 1; r <= (2 * n - 1); r++) {
            for(int st = 1; st <= nst; st++) {
                System.out.print("* ");
            }
            for(int sp = 1; sp <= nsp; sp++) {
                System.out.print(" ");
            }
            for(int st = 1; st <= nst; st++) {
                if(r == n && st == nst - 1) {
                    st++;
                }
                System.out.print("* ");
            }
            System.out.println();
            if(r < n) {
                nsp -= 2;
                nst++;
            } else {
                nsp += 2;
                nst--;
            }
        }
    }
}

```

9 = 5

9 rows

n = 5

9 = 1 2 3 4 5

nsp = 7

8 1 -1

nst = 1

2 3 4 5

\* \_ \_ \_ \_ \_ \*

\* \_ \_ \_ \_ \*

\* \_ \_ \_ \*

\* \_ \_ \*

\* \_ \*

9 = 5

1

sp <= -1

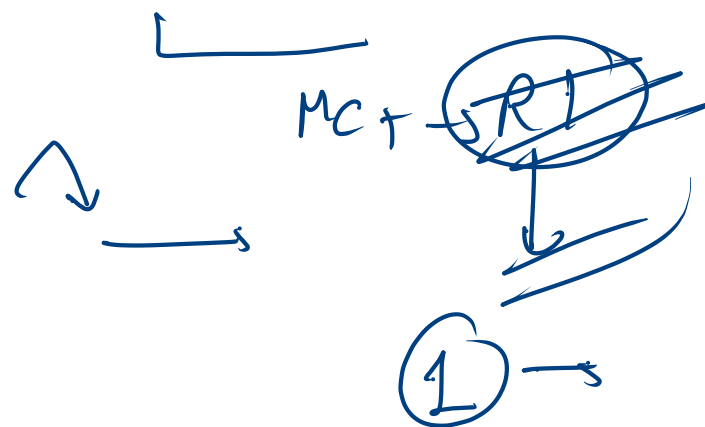
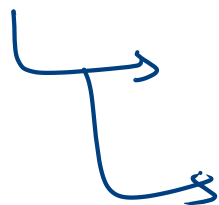
1 2 3 4 5

st = 1 2 3 4

6

60%

40%



$f = 0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4$   
 $s = 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$   
 $next = 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$

(n)

n=5

~~III~~  
✓

```

for (int i = 1; i <= n; i++) {
    → s = 0; P(f)
    → next = f + s;
    {
        f = s;
        s = next;
    }
}

```

$\left\{ \begin{array}{l} 0 \\ 1 \\ 1 \\ 2 \\ 5 \end{array} \right\}$  ✓

init

checking

update

for (init  $i=0$ ;  $i < n$ ;  $i++$ ) {

{

init  $i=0$ ;

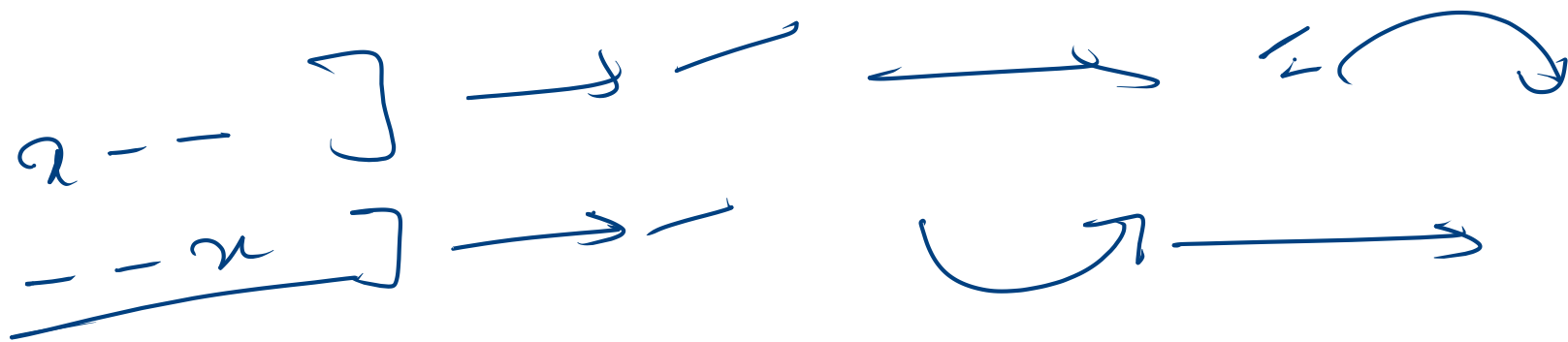
while ( $i < n$ ) {

$i++$

}

$x++ \rightarrow$  Post Increment  $\rightarrow$  use then update  
 $++x \rightarrow$  Pre Increment  $\rightarrow$  update then use

---




int n = ~~5~~ 7

System.out.println(n++);  
System.out.println(++n);

{ 5  
7 }



fast 

perfect

3      12      60      360      2160



1 7 0 2 5 6 9 10 12 14 16 18



```

public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt(); // 5 4 3 2
    → int prod = 1;
    while (n > 0) {
        int num = scn.nextInt(); ←
        prod = prod * num; ←
        System.out.print(prod + " ");
    }
}

```

prod = 1 2 8 64

use → decrease  
↑      ↑

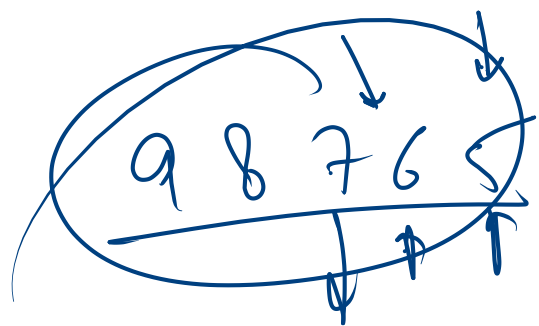
n > 0

{ not clear }  
clear

2      4      8

Compare → figure out whether to → update  
our first loop or no

2      8      64



→ 5

5

123  $\frac{1}{10}$  10

12  $\frac{1}{10}$   
10

12/10 → 1

2 -

→ 123  
→ 10

1  $\frac{1}{10}$

1/10

①

1

8  
6  
7  
8  
9

{  $\frac{1}{10}$  }  
{ 10 }

10

1 2  
123  
101  
23

20  
3

①  $\rightarrow$  1237148

⑦

1 2 3 4 5  
5 4 3 2 1

- i) Count the digits in a number.
- ii) Sum a number.