Class - Pattern Problems

Concept

- warm up Problem.

Part-I

- break & continue
- Ternary operator
- Multiple Inputs
- ++ , -- operators (Inc-Dec)

Part-II

- Pattern Problems
- HW & Doubts
- Common & Mistakes

(a) Given a Number N, suppose I want to find

- Last Digit of Number — (5)
- First Digit of Number — (2)
- Sum of Digits of the Number — 2+3+6+5 = 16

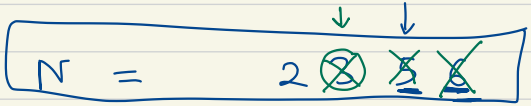
Input N = (2)3 6(5)

(I) N = 2356 N%10 → last Digit

(II) N = 2356
↑

• 2356

Extract all the digits from given NO



Last_Digit

Sum = 0

```
while(N>0){
    digit = N%10
    sum = sum + digit
    N = N/10
}
```

N	digit	
2356	6	+ 6
235	5	+ 5
23	3	+ 3
2	2	+ 2
0		Stop

N%10 = 0

$\frac{2356}{10} = 235$

$\frac{2}{10}$

→ print(sum) → 16

→ print(digit) → first digit

14 + 2 = 16

Float

123.25

(int) 123.25

123

Another Approach

N = 2356

```
while(N>10){
    N = N/10;
}
```

print(N)

N will reduce single digit no (less than 10)

2356 >10

235 >10

23 >10

N ⇒ 2 >10 No

Stop

Sum part

N 2 3 5

sum = 0 +6 +5 +3 +2

235

N

digit = N%10

235 → 6

23 → 5

2 → 3

0 → 2

Stop

Stop

while (N>0) {

① digit = N%10

② sum = sum + digit

③ N = N/10

3

↑ Modulo (Rem)

n%10 ⇒ 517%10 = 7

n/10 ⇒ 517/10 = 51

↓ Division

51.7

---

float ans = (float)517/10 = 51.7

BREAK

↳ used inside loop
↳ alter the flow of the loop
↳ Stop the loop as the statement is executed.
based upon certain condition

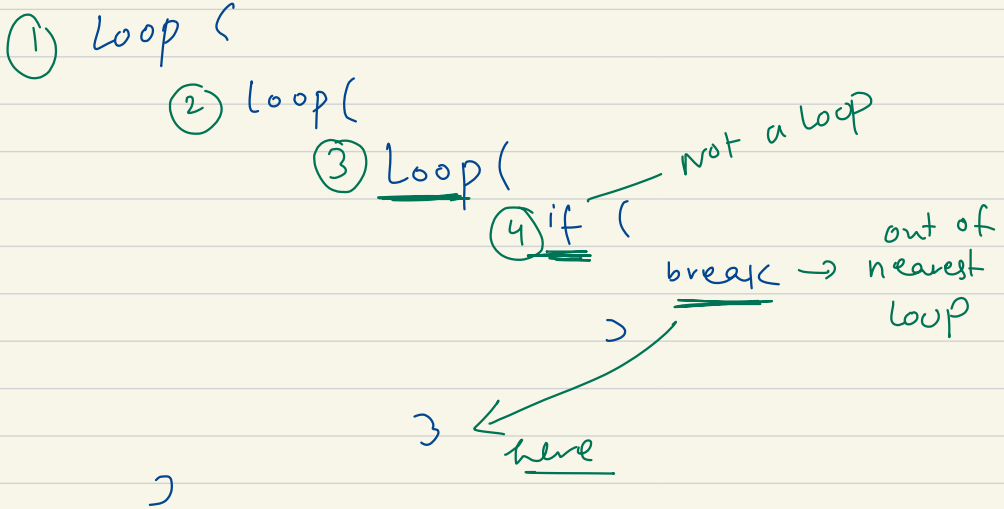
So Cal

```
for ( cal=0;  cal<=50,  cal = cal+1 ) {
    if ( gf_call_up ) {
        break;
```

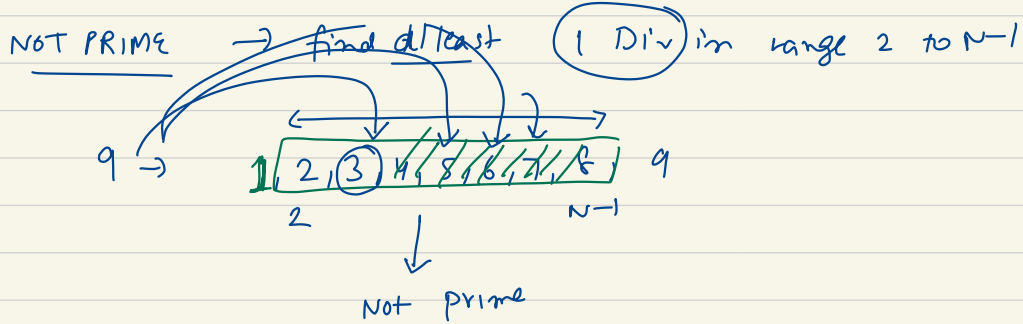
3

2

out of loop


1 Loop {
2 loop {
3 Loop {
not a loop
4 if (
break → out of nearest loop
}
here
}

Prime NO — No divisors except 1 and N.




```
i=2
while (i<= N-1){
    if (N%i ==0){
        → NOT PRIME
        → break;
```

Stop the loop early

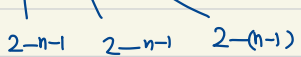
3 3

PRIME NO'S IN RANGE

A B

1 — 20

- 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, ... 20



Print first N Primes

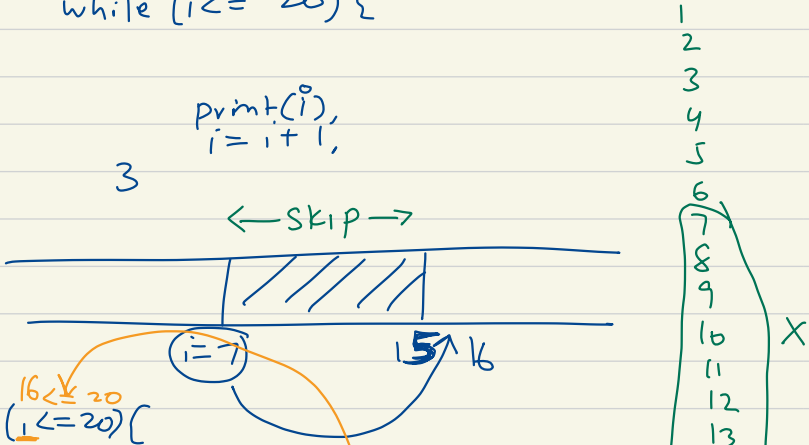
10.35
'continue'

Continue

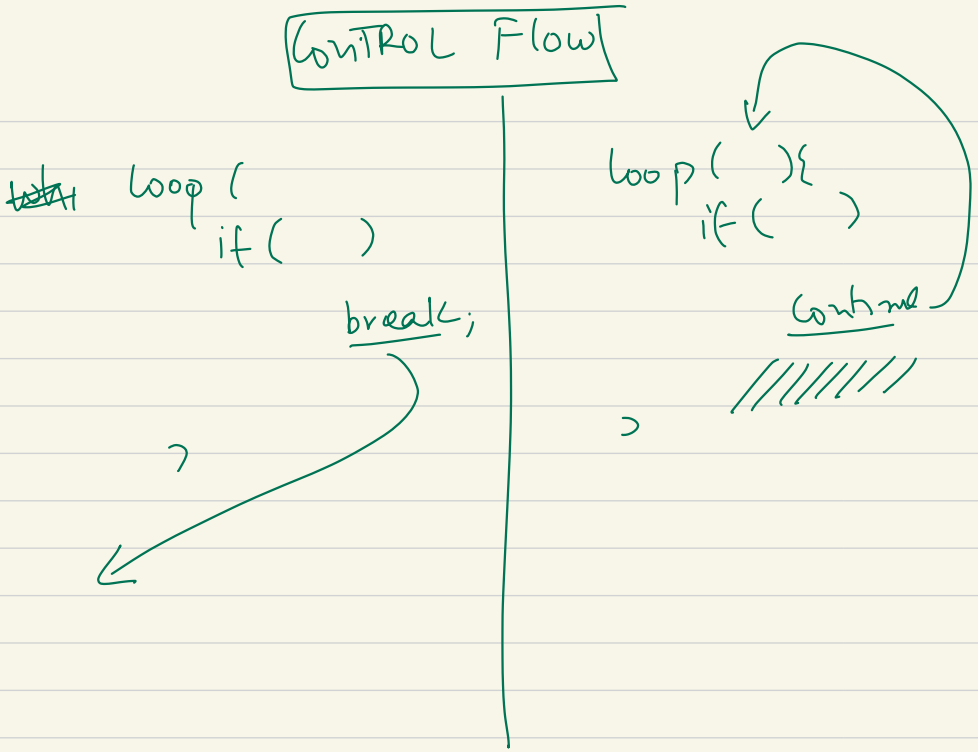
↳ used inside a loop

↳ control (exec flow) to the starting of the loop.

```
i = 1
while (i <= 20) {
    print(i),
    i = i + 1,
}
```



```
i=1
while (condition Y) {        // i<=10
    if (condition X) {       // i==7
        continue;
    }
    work.
    i = i+1.
}
```


CONTROL FLOW
loop (
if ( )
break;
}
loop ( ){
if ( )
continue
}

Next Class

- Ternary op
- ++, --
- Switch case

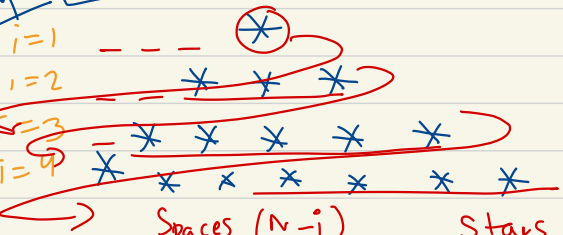
Pattern

```
1
1 2
1 2 3
1 2 3 4
```

```
1
2 3
4 5 6
7 8 9 10
```

```
*
* *
* * *
* * * *
```

Warm up — N=4



Code

1. N Rows
2. N-i spaces, 2i-1 stars

	Spaces (N-i)	Stars 2i-1
i=1	4-1 = 3	1, 2(1)-1 = 1
i=2	4-2 = 2	3, 2(2)-1 = 3
i=3	4-3 = 1	5, 2(3)-1 = 5
i=4	4-4 = 0	7, 2(4)-1 = 7

Stars = 1

Stars +2 = 3

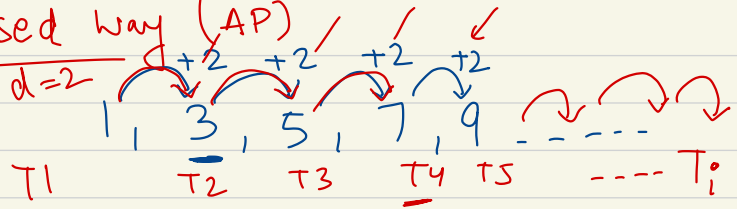
⋮

$T_i = 1 + (i-1)2$

$= 1 + 2i - 2$

$= \boxed{2i-1}$

Generalised way (AP)



$$T_4 = T_1 + 3d$$
$$= 1 + 3(2)$$
$$= 7$$

$$T_i = T_1 + (i-1)d$$
$$= 1 + (i-1)2$$
$$= 1 + 2i - 2$$
$$= 2i - 1$$

AP $a, a+d, a+2d, a+3d \ldots\ldots, a+(i-1)d$

i^{th}, $5, 9, 13, 17, \ldots\ldots$ (i^{th})

$$T_i = T_1 + (i-1)d \qquad i = 7$$

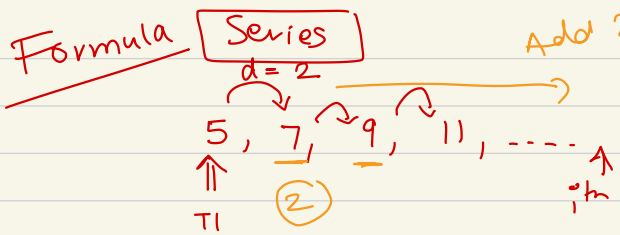
$$= 5 + (i-1)4$$

$$= 5 + 4i - 4$$

$$= 4i + 1$$

$$= 4(7) + 1 = 29$$

Formula | Series



5 → +2 → 7 → 2 → 9

$$T_i = T_1 + (i-1)\,d$$

$$= 5 + (i-1)\,2$$

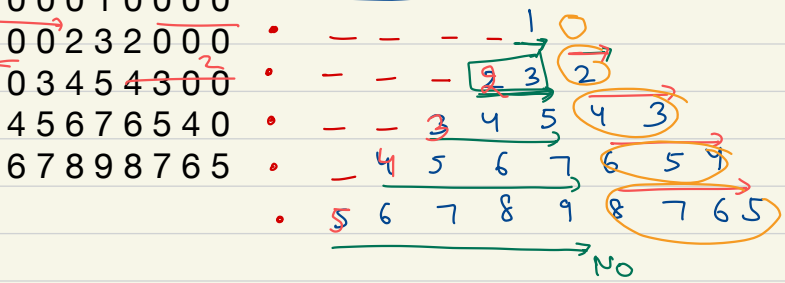
$$= 5 + 2i - 2$$

$$= 2i + 3$$

Assignment

```
    0 0 0 0 1 0 0 0 0
    0 0 0 2 3 2 0 0 0
    0 0 3 4 5 4 3 0 0
    0 4 5 6 7 6 5 4 0
    5 6 7 8 9 8 7 6 5
```

N = 5



Obs

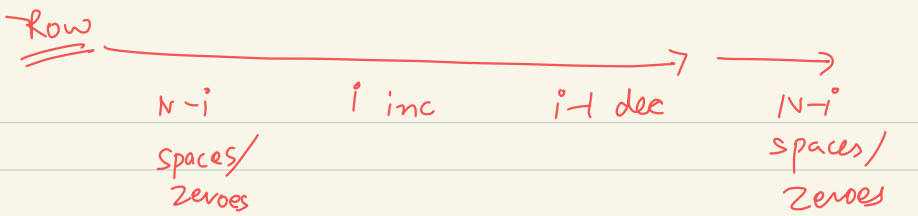
1. N Rows

2. Each ~~N-i spaces~~ zeroes

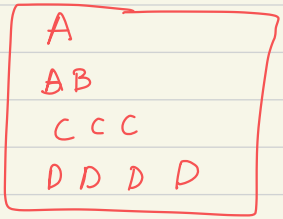
i Nos in inc Order

Starting from val that is i

	Spaces (N-i)	inc	dec
i=1	4	1	0
i=2	3	2	1
i=3	2	3	2
i=4	1	4	3
i=5	0	5	4
		i times	i-1 No



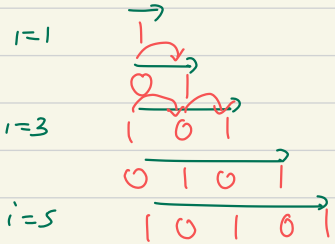
i=1
i=2
i=3
i=4



val = A
↳ B
↳ C
↳ D α i times

Optional

N = 5

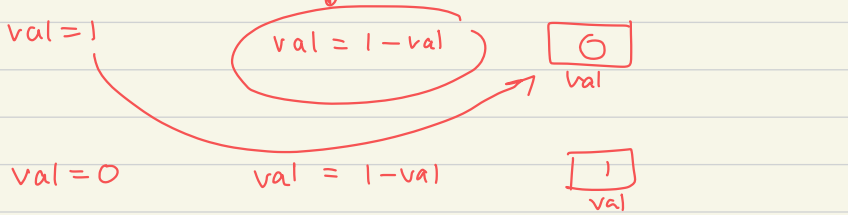


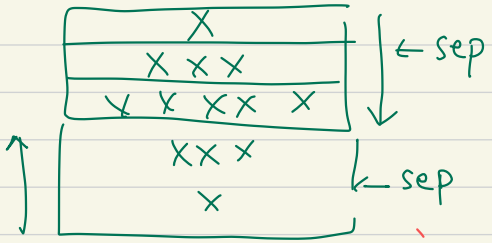
Obs

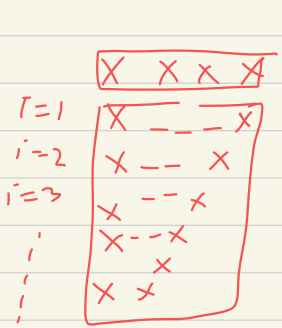
1. N Rows
2. i digits in i^{th} Row
3. Odd Row
   ↳ val = 1

   even
   ↳ val = 0
4. Toggle 0 → 1 → 0 → ...

Toggle







star
star

for (i= _____) (

spaces 2 loop

3