Ⓐ
```
{
        n=n+1;
}
```

Ⓑ
```
{
    n=n+2;
}
```

1
↓
3
↓
5
↓
7
↓
9

A)
```
count =1
while( count < 6)
{   // deliver
    count ++;
}
```
✗

1	✓
2	✓
3	✓
4	✓
5	✓

```
count =1
while ( cout <=6)
{
    // deliver
}
```
infinite loop

$345\cancel{2}/10 = \boxed{345}$

```
N = 3452
    count=0;
while( N>=0)
{
    N=N/10;  count++;
}
```

loop will never stop!

	N>=0	
N = 3452	True	co= 1
N = 345	True	2
N = 34	True	3
N = 3	True	4
N = 0	True	5
N=0	True	6
N=0 ↓		7
		⋮

A)

```
i=1
while(i<=N)
    print(i)
i = i+4
```

1 5 ...

B)

```
i=1
while(4*i<=N)
    print(4*i)
    i=i+1;
```

4

~~C)~~

```
i=0
i<=N
4*i.
i=i+1
```

0 4 8 12

D)

N=5

```
i =1
i <=N
print(4*i )
i=i+1
```

i=1	4
i=2	8
i=3	12
i=4	16
i=5	20
i=6	

first N multiples of 4

N → 5

4	8	12	16	20

Q Extract last digit from a given no. N.

N = 3456

% 10

N % 10

$3456 \% 10 = \boxed{6}$

a%b = remainder when a/b

% → remainder

a/b → division

16/3 = 5 — quotient

16%3 = 1 — remainder

Q print all digits of a given number N in a new line ⇓

$N = 21435$

5
3
4
1
2

$N = 2150$

0
5
1
2

$\boxed{N = 31546}$

✗

$N \% 10 = 31546 \% 10 = \boxed{6}$
$N \% 100 = 31546 \% 100 = \boxed{46}$
$N \% 1000 = 31546 \% 1000 = \boxed{546}$

3 1 5 4 6

to know what is last dig

$N \% 10$ → $\boxed{6}$

to remove last dig ← $N = N/10$ → 3154

3 1 5 4

$N \% 10$ → $\boxed{4}$
$N = N/10$ → 315

3 1 5

$N \% 10$ → 5
~~N~~ $N = N/10$ → 31

3 1

$N \% 10$ → 1
$N = N/10$ → 3

3

$N \% 10$ → 3
$N = N/10$ → 0

Do we initially know how many digits?
↓
No

Int

N=0

```
if (n<0) { n = n*-1; }
```

edge case

exceptional

```
~~else~~ if ( N==0)
{      sop( 0); }
else{ while ( N>0)
      {
          System.out.println(N%10);
          N=N/10;
      }
}
```

sum = sum+ N%10;

-ve numbers

-6437 %10 ~~X~~ 7

7
3
4
6

how mod works with negative!

Later

digits(-6437) *-1

= digits(6437)

Q sum of digits of N?

```
sum = sum+ N%10;
```

Q integer $\underline{N}$ ⇒ reverse the integer

34561 $\boxed{1\ 6\ 5\ 4\ 3}$

final ans should be an int only

$N = \boxed{3461} * 10 + 2$ $\boxed{\text{append} \Rightarrow 2}$

$\Downarrow$

$\boxed{34612} * 10 + 3$ append 3

$\downarrow$

$\underline{346123}$

$\boxed{r * 10 + d}$ $\leftarrow d$

$N = 3\ 4\ 5\ 6\ 1$ ✓

ans = 0

got ↓ append ↓ removed

$N \% 10 = 1$

ans $* 10 + 1 = 0 * 10 + 1$

$\boxed{ans = 1}$

$\underline{N = N/10}$

✓

3 4 5 6

$N \% 10 = 6$

ans $* 10 + 6 = 1 * 10 + 6$

$\boxed{ans = 16}$

$\underline{N = N/10}$

3 4 5

$N \% 10 = 5$

ans $* 10 + 5 = 16 * 10 + 5$

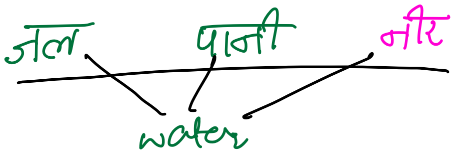
$\boxed{ans = 165}$

until $N > 0$

```
ans = 0;
while( N > 0 ) {

    int d = N%10;
      ans = ans * 10 + d;
     N = N/10;
}
  SOP(ans);
```

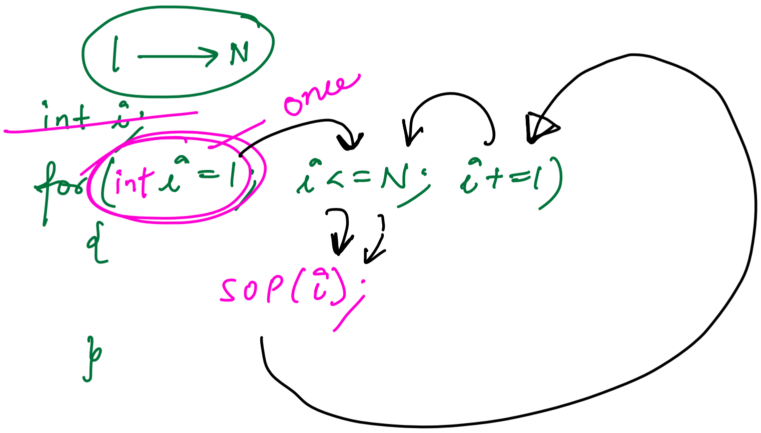
```
initialisation

while ( condition ) {
    statements
    update
}
```

while   for



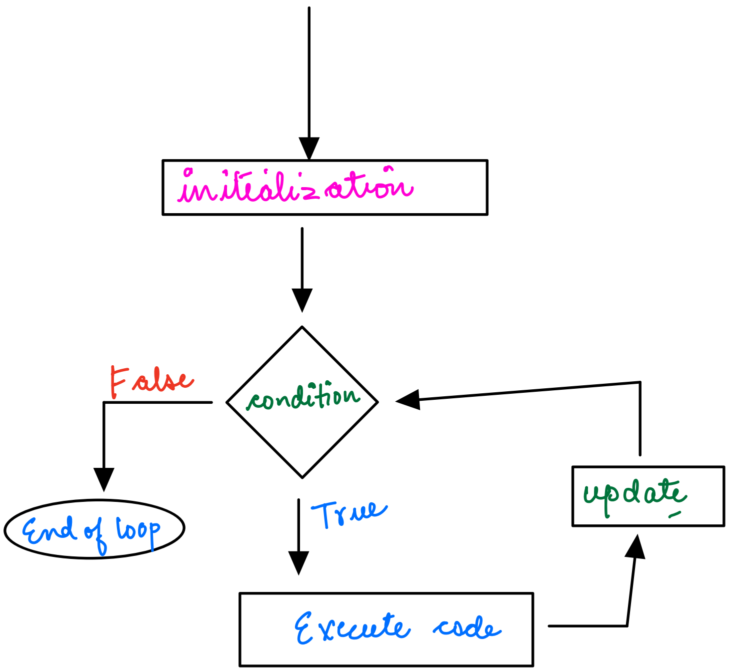
for every employee
add there salary into sum

```
for (intialisation; condition; updation) {
    statements
}
```

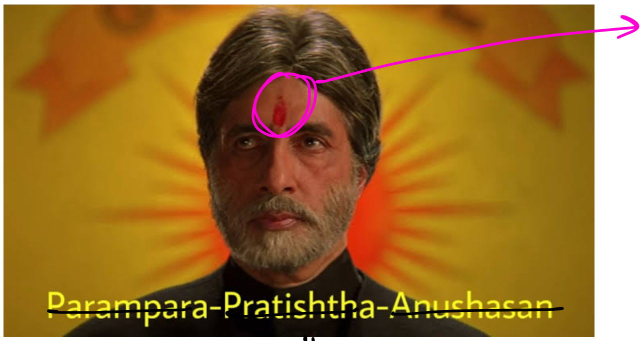


~~int i;~~

```
for (int i = 1; i <= N; i += 1)
{
        SOP(i);
}
```



#



Trust → me

Patience ⇓ consistency discipline

factors of a number N

⇓

when it completely divides N

what are factors?

24

1 2 3 4 6 8 12 24

8

$24/4 \Rightarrow$ remainder = 0

8

1 2 3 4 5 6 7 8

$1 \longrightarrow N$

go from $1 \longrightarrow n$ i

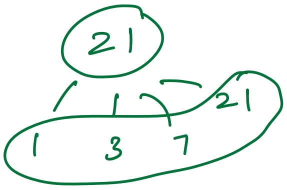
check if i divides n

```
count = 0;
for( int i = 1; i<=n; i++)
{
      // if i divides n
      if( n % i == 0)
      {      // i is a factor.
             count ++;
      }
}
SOP(count);
```

→ counting the no of factors

prime numbers

exactly 2 factors
1 & itself



1 → 1

prime composite

{2 → smallest prime no}

HW: checking if a no is prime or not

↓

check if(count==2)

Do every quest from previous class

↓

for loops