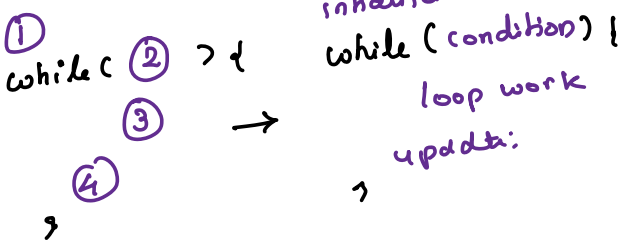
1 → Initialisation

2 → condition

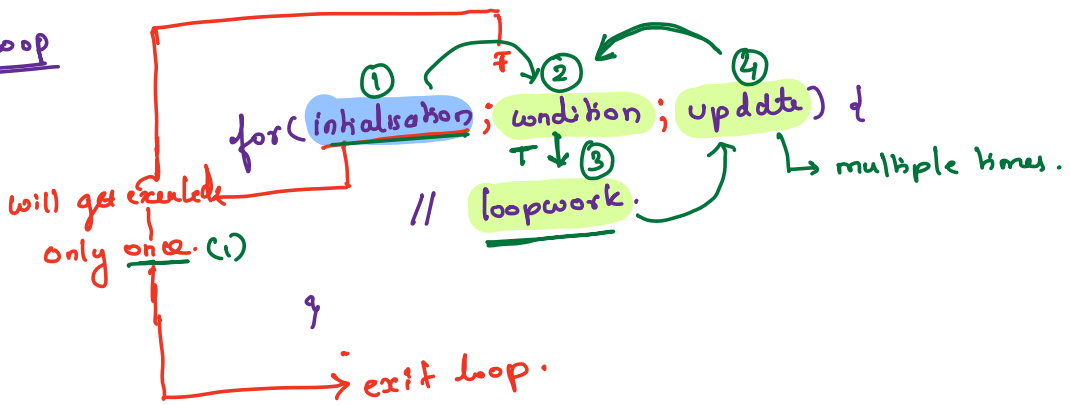
3 → loop work

4 → update

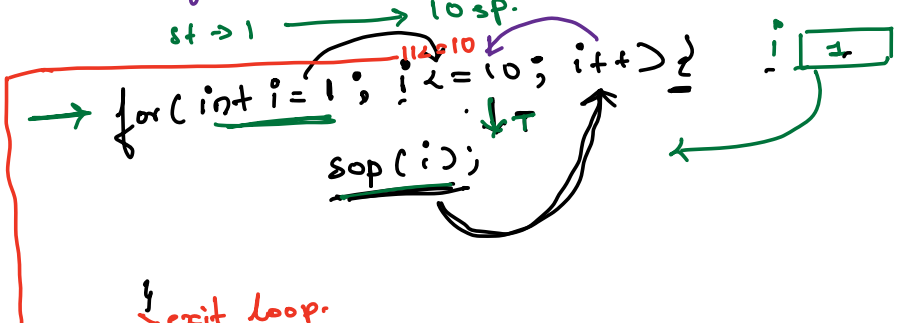
2 → condition → will repeat until it fails.



For loop



Q) Print number from 1 to 10;



i = 1 → 2 → ... 10 → 11 → exit loop.

o/p => 1 2 10

Q → write a for loop to print odd Number from 10, 20, 30

1 to N; (int N = sc.nextInt();)

1 ⟶ 10

i <= 10

i <= 20

⋮

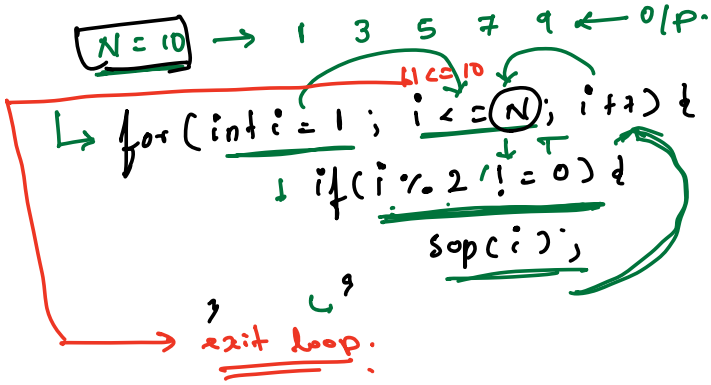
i <= N

1 ⟶ 10

10 ⟶ 1

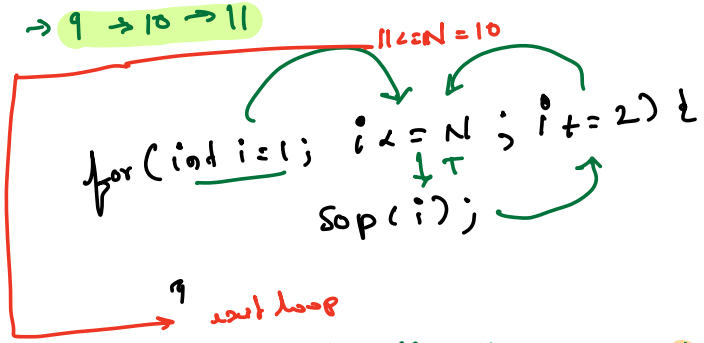
(i=10; i>=1; i--)

N = 10 → 1 3 5 7 9 ← o/p.



i=1→2→3→4→5→6→7→8 o/p :→ 1_3_5_7_9_

→9→10→11



1 +2⇒ 3 +2⇒ 5 +2→ 7 +2→ 9 +2→ 11 o/p 1 3 5 7 9

↓ exit loop.

Quiz

1 -> Initialisation (only once);

2 -> condition

3 -> loop work

4 -> update

① 1 → 2 → 3 → 4 then stop ✗

② 1 → 2 → 3 → 4 → 1 and stop. ✗

③ 1 → 2 → 3 → 4 → start from 3 ✗

④ 1 → 2 → 3 → 4 → then again start from 2 ✓

Que-2

1) 0 → N ✗
   sop(i);

2) i=1 → i<=N (i+2) ✗
   sop(i) → odd no

3) int i
   i=2 i<=N ++ ✗
   3, 4, 5, 6

4) i=2 → i<=N = i=i+2; ✓
   2 4 6 8 10...

Q23 → (Pm)

A=>

```
(int i;) → Declaration
for(i=N ; i>=1 ; i--) {
    sop(i); →
}
sop(i) → ? (Pm)
```

→ scope of i

initializing variable

Q4)

```
psvm(String args[]){
→ for(int i=1; i<=3; i++){
      sopln(i);.
  }
→ sopln(i);  → error → C.E
}                 i not defined.
```

Q5)

```
psvm(String args[]){
→ int i=1;
→ for(int i=1; i<=3; i++){
      sopln(i);
  }
  sopln(i)
}
```

int i=1

compilation error

i already declared main method.

Q6)

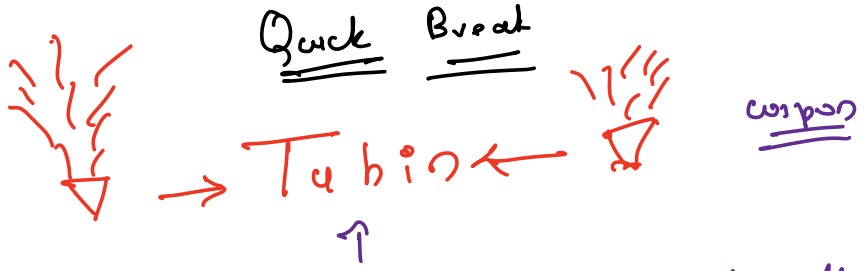
```
psvm(String args[]){
  int i;
→ for(i=1; i<=3; i++){
      sopln(i);
  }
exit loop → sopln(i);
}
```

i = 1

i = 1 → 2 → 3 → 4

o/p

1
2
3
4



Sponsors → snack, tea, coffee

10:27 => 10:40 pm

Q => Print first and last digit of a number

N = 6175

→ 5 → last digit → N%10;

??? = first digit ←

o/p 6 → fd
5 → ld.

int fd;

1) declare fd;
2) store d → digit inside fd.
3) outside loop print fd;

fd → first digit

getting int ← of last digit

6175 ←

d = n%10; → 5
sop(d); fd = d;

n = n/10. => 617

o/p 5 7 1 6 → fd

sop(fd);
sop(ld);

```
int n = sc.nextInt();
int ld = n%10;
int fd=0;
for(int i=n; i>0; i=i/10){
    int d = i%10;
    fd = d;
}
sop(fd);
sop(ld);
```

```
int i=n;
while(i>0){
  int d = i%10;
  i = i/10;
}
```

n = 1432 → ld = 2 fd = i = 1432

i	i>0	d (i%10)	fd	i = i/10
1432	T	2 → 2		143
143	T	3 → 3		14
14	T	4 → 4		1 (carry)
1	T	1 → 1		0
0 → F → exit loop				

O/P fd = 1 ld = 2

Q=> Given a number N, reverse the number.

N = 6123 ⇒ 3216

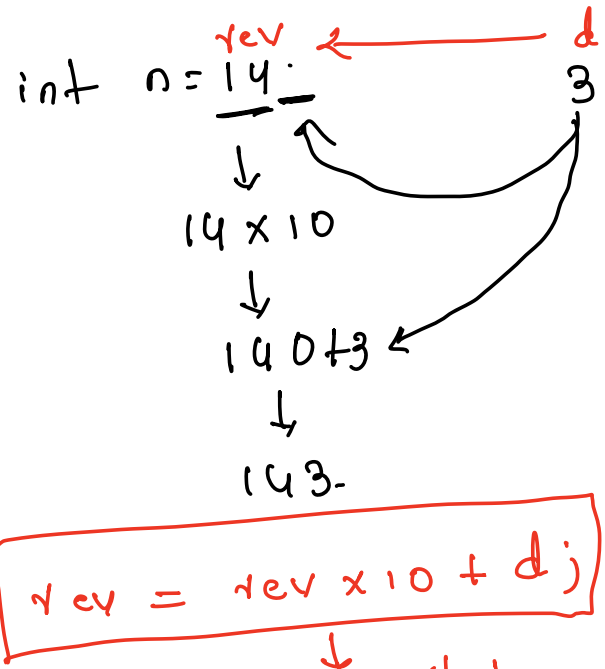
715 ⇒ 517.

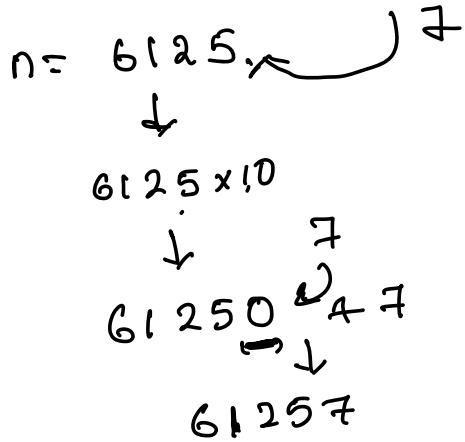
String

Print

int n = 421

int rev = 124;





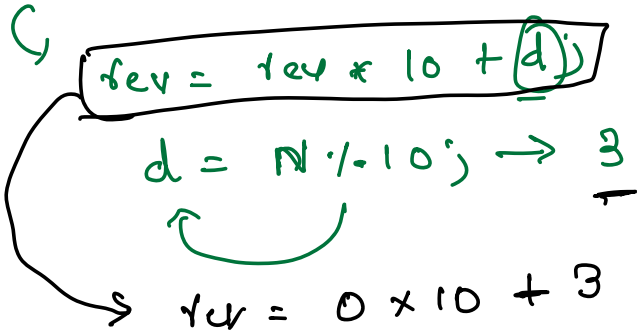
rev = rev x 10 + dj

↓

Adding digit to a number at the last

3

N = 6143 =>

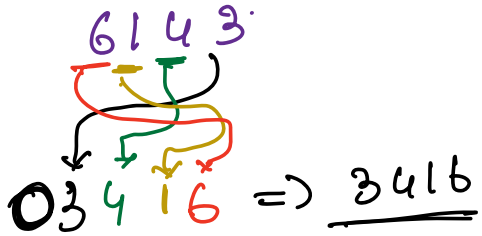


rev = 0 x 10 + 3

= 0 + 3

= 3.

=) approach =)



1) rev = 0;

2) d = n % 10

3) rev = rev x 10 + d

4) n = n / 10;

Condition

```
int n = sc.nextInt();
int rev = 0;
for(int i=n; i>0; i=i/10){
        int d = n % 10;
        rev = rev * 10 + d;
}
sop(rev);
```

Olp ⇒ 2413

I/p 3142

Doubt → Amstrong number

↳ 153 => $1^3 + 5^3 + 3^3$

=> 1 + 125 + 27

=> 153

(num == sum) (limited 3 digits)

↓

amstrong number.

1 2 3 4 5 ... N

(1) (2)

12

3

N

```
int N = sc.nextInt();
int num = 1; 2 3 ... N
while (num <= N) {
    int sum = 0;
    while (num > 0) {
        int d = num % 10;
        sum = sum + (d*d*d);
        num = num / 10;
    }
    if (num == sum) {
        sop(num);
    }
    num++;
}
```

1) take a loop from 1 → N

2) take a loop inside the above loop.

3) use the inner loop to get all digit

4) int sum = 0;

5) sum = sum + (d*d*d).

Cube

loop → 1 ⟶ N

inner ↳ (loop2 to check amst sum)

→ if (num == sum) {

print num

5) Compare sum with num

↓

if true → Print if false → ignore

```
long bal = sc.nextInt();
int T = sc.nextInt();
while (T > 0) {
    int op_type = sc.nextInt();
    long amt = sc.nextInt();
    if (op_type == 1) {
        bal = bal + amt;
        sop(bal);
    } else {
        if (bal >= amt) {
            bal -= amt;
        } else {
            sop("Transaction failed");
        }
    }
    T--;
}
```

```
int A = 5;
int B = 10;
while (A++ < B) {
    sop(A + " ");
}
sop(B);
```

5 _ 6 _ 7 _ 8 _ 9 _ 10

N = 35 ⟵ 2 ⟶ 1

35 / 2 → 17 / 2 → 8 → 4 → 2 → 1

① – ② – ③ ④ ⑤

steps 0; n = 35

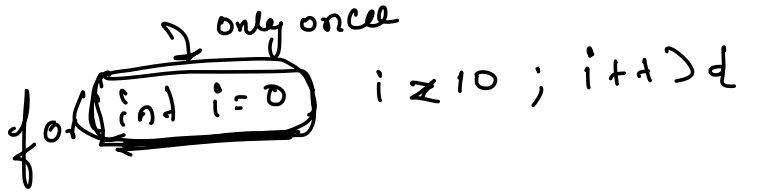
while (n != 1) { 35-17-8-4-2-1

n = n / 2;

steps++;

}

sop(steps);



→ int i=0 ; only once

{ while (i<=10){

sop(i);

i++

}

https://www.interviewbit.com/snippet/fd1f1bf0b196055aa38c/