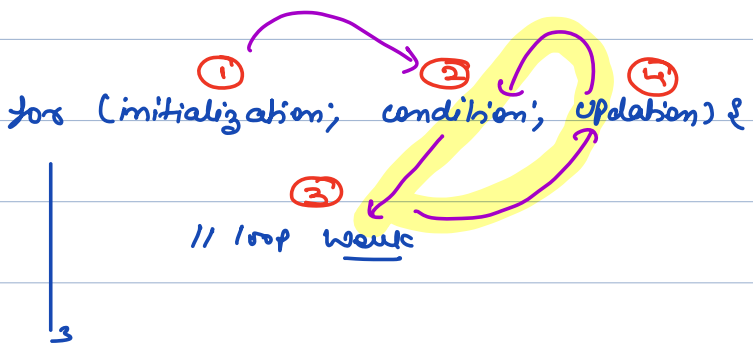
for loops

```
1.  initialization
2.  while (condition) {
3       // loop work
4       updation
    }
```



Ques) Given n as input, print from 1 to n.

i=1 2 3 4 5 n=5.

```
for (int i=1; i<=n; i++)
    print(i);
```

1 2 3 4 5

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

Ques) Given an input N, print odd numbers from 1 to N.

N = 5,   1 3 5

```
int i=1;
while (i<=N) {
      sop (i);
      i+=2;
}
```

```
for (int i=1; i<=N; i+=2) {
      sop(i);
}
```

-: **factors** :-

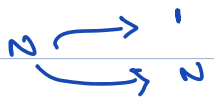
↳ x is a factor of N, if N is a multiple of x.

6 = 1, 2, 3, 6

10 = 1, 2, 5, 10

24 = 1, 2, 3, 4, 6, 8, 12, 24

Ques) Print all factors of n.



n → factors are in range [1 n]

```
for (i=1; i<=n; i++) {
    if (n%i == 0) {
        print(i);
    }
}
```

n = 6 → 1 2 3 6

Prime number

number divisible by 1 & itself. ✗

↳ n=1.

↷ n>0.

number with 2 factors. ✓

n=1 => 1 ✗

n=2 => 1, 2 ✓

n=4 => 1, 2, 4 ✗

Ques) Given a number n, check if it is a prime number.

```
int count = 0;
for (i=1; i<=n; i++) {
    if (n%i == 0) {
        count++;
    }
}
if (count == 2) {
    sop("prime");
} else {
    sop("not prime");
}
```

i	n%i	count	N=12
1	12%1==0	1	
2	12%2==0	2	
3	12%3==0	3	
4	12%4==0	4	
5	12%5≠0	4	
:			
12			

```
int count = 0;

for (i=1; i<=n; i++) {
        if (n%i == 0) {
                count++;
        }
        if (count>2) {break;}
}

if (count==2) {
        sop("prime");
} else {
        sop("not prime");
}
```

break
↓
stops the parent loop.
(just immediate parent loop only);

if n = 10^{10}.
without break => 10^{10}
with break => 4 times
1 2 3 4

Continue; keyword
↓
skips to the next iteration

```
//print odd no's from 1 to n

for (i=1; i<=n; i++) {
        if (i%2==0) {
                continue;
        }
        sop(i);
}
```

n = 10.

https://www.scaler.com/topics/java/online-java-compiler/?snippet_slug=79d72503cc08abbbafcc