```
Hello        Sop("Hello");
Hello        Sop("Hello");
Hello        Sop("Hello");
Hello        Sop("Hello");
```

Loop → to do an operation repeatedly.

While Loop

```
// initial value of some var
while (condition){
    // body
    // update var
}
```

```
int count = 0;
while (count < 4){
    System.out.println("Hello");
    count++;
}
```

// count = count + 1;

count	Loop	O/P
0	✓	Hello
1	✓	Hello
2	✓	Hello
3	✓	Hello
4	✗	

Q1) Given an integer n as input, print the nos. from 1 to n.

n = 4

1
2
3
4

```
int n = sc.nextInt();
int i = 1;
while(i<=n){
    System.out.println(i);
    i++;    // or ++i;
}
```

```
int n = sc.nextInt();
int i = 1;
while(i<=n){
    System.out.println(i++);
}
```

i++ → Post-increment operator

++i → Pre-increment operator

```
i = 3
y = 5
x = y + i++
```

```
j = 3
y = 5
x = y + ++j
```

j → 4

x = 5 + 4 = 9

i → 4

x → 5 + 3
= 8

Q2) Given an integer n as input, print the nos. from n to 1.

n = 4 4 3 2 1

```
int n = sc.nextInt();
while(n > 0){
    System.out.print(n+" ");
    n--;
}
```

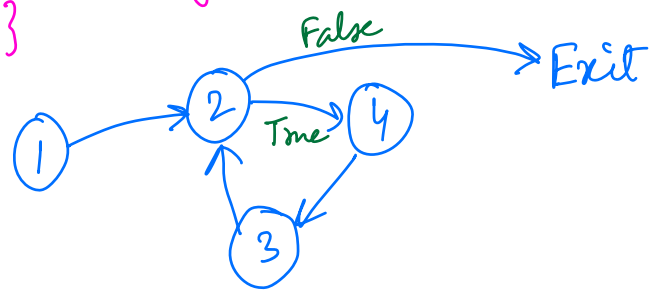
Q3) Given an integer n as input, print the odd nos. from 1 to n.

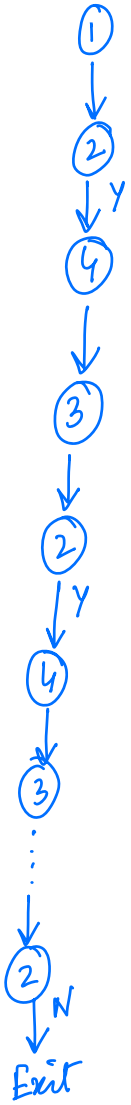
n = 6
1
3
5

```
int n = sc.nextInt();
int i = 1;
while(i <= n){
    System.out.println(i);
    i += 2;
}
```

For loop

```
for(initialization; condition; update){
    // Body
}
```





Q1) given n as input, print 1 to n.

```
for(int i=1; i<=n; i++){
    Sopln(i);
}
```

Q2) given n as input, print odd nos. from 1 to n.

```
for(int i=1; i<=n; i+=2){
    Sopln(i);
}
```

[Break till 10:08 PM]

Break statement

```
while(...){
    ...
    break;
    ...
}
```

exit the loop

Q) Check if a nr. n is prime or not.

→ a nr. with exactly 2 factors → 1 and itself.

Primes → 2, 3, 5, 7, 11, 13, 17, 19, ...

```
int n = sc.nextInt();
int cnt = 0;
for(int i=1; i<=n; i++){
    if(n%i==0){
        cnt++;
        if(cnt>2)
            break;
    }
}
if(cnt==2)
    Sop1n("Prime");
else
    Sop1n("Not Prime");
```

n = 15

i=1 → 15%1=0 ✓
cnt → 1
i=2 → 15%2≠0
i=3 → 15%3=0 ✓
cnt → 2
i=4 → 15%4≠0
i=5 → 15%5=0 ✓
cnt → 3
3>2 ✓

```
while(...){
    - -
    - .
    -
    continue;
    - -
    - -
}
```
} skip for this iteration

8) Print all even nos. from 1 to n.

```
int i=2;
while(i<=n){
    Sop1n(i);
    i+=2;
}
```

```
for(int i=1; i<=n; i++){
    if(i%2!=0)
        continue;
    System.out.println(i);
}
```

Questions with T test cases

Given T test cases, for each test case take a number as input and find if it is even or odd.

eg:

3 ⟶ Nr. of test cases

1. 6 ⟶ Even
2. 9 ⟶ Odd
3. 13 ⟶ Odd

```
int t = sc.nextInt();
for(int i=1; i<=t; i++){        <⟶>  while(t-->0)
    int n = sc.nextInt();
    if(n%2==0)
        Sop1n("Even");
    else
        Sop1n("Odd");
}
```

Scope of variables

Scope of a variable → part of a code where the variable is valid. (can be accessed)

```
int a=5;
if(x==3){
    int b=4;
    S.o.pln(a); ✓
    S.o.pln(b); ✓
}
S.o.pln(a); ✓
S.o.pln(b); ✗
```

```
int a=5;
int b=0;
if(x==3){
    b=4;
    S.o.pln(a);
    S.o.pln(b); ✓ → 4
}
S.o.pln(a);
S.o.pln(b); ✓ → 0 or 4.
```

```
int b=5;

{
   int a=4;

}

// b is valid, a doesn't exist.
```