Ques. Print natural numbers from 1 to 5.

$\boxed{\cancel{2}}\ \cancel{3}\ 4\ 5$

i

```
S.O.Pln (1);
SOPln (2);
SOPln (3);
S.O.Pln (4);
S.O.Pln (5);
```

```
int i = 1;
[ S.O.Pln (i);   →  1
[   i = i+1;     //  i = 2
  S.O.Pln (i);   →  2
    i = i+1;     //  i = 3
  S.O.Pln (i);   →  3
    i = i+1;     //  i = 4
  S.O.Pln (i);   →  4
    i = i+1;     //  i = 5
  S.O.Pln (i);   →  5
    i = i+1;     //  i=6
```

While loop

```
  int i =1 ;
while ( i<=5 ){

      S.O.P ln (i) ;
      i = i+1;

}
```

```
  int  i = 1 ;
while ( i<=5 ) {

      S.O.P ln (i) ;
      i = i+1;
}
```

i	i<=5	Output	i=i+1
1	true	1	2
2	true	2	3
3	true	3	4
4	true	4	5
5	true	5	6
6	false	⟶	Break [exit the loop]

Structure of while loop →

Step 1 → Initialisation of loop variable.

```
int i=1;
```

Step 2 → Write while with condition [boolean expression]

```
while (i<=5) {
```

Step 3 → write code for task

```
S.O.Pln (i);
```

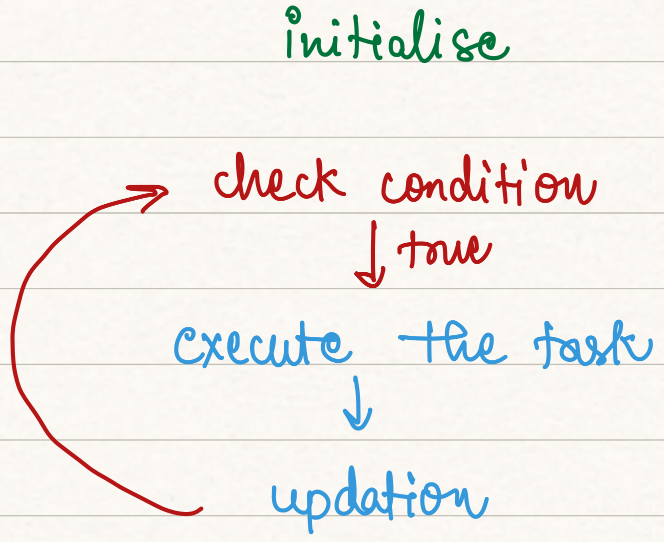
Step 4 → Updation of loop variable

```
i= i+1;
```

Syntax →

```
initialise
while (condition) {

    // task to be repeated
    // updation
}
```



Ques. Print even numbers from 0 to n

n = 4 → 0 2 4

```
✗ int   i = 0 ;
while  (i ≤ n  &&  i%2 == 0) {

      S.O.Pln( i );
      i++;

}
```

n = 4

i	i<=n && i%2==0		Output	i++
0	T && T ⇒	T	0	1
1	T && F ⇒	$\boxed{\text{false}}$	⟶ Break	

```
int i = 0;

while (i <= n) {
    if (i % 2 == 0) {
        S.O.Pln(i);
    }
    i++;
}
```

% → modulus

/ → divide

n = 7 → 0 2 4 6

i	i <= 7	i % 2 == 0	Output	i++
0	true	true	0	1
1	true	false	-	2
2	true	true	2	3
3	true	false	-	4
4	true	true	4	5
5	true	false	-	6
6	true	true	6	7
7	true	false	-	8
8	false	→	Break	

i → 0 1 2 3 4 5 6 7

```
int i = 0;
while (i <= n) {

      S.O.P ln (i);
      i = i+2;
}
```

n = 7

i	i <= 7	Output	i = i + 2
0	true	0	2
2	true	2	4
4	true	4	6
6	true	6	8
8	false	→	Break

i → 0 2 4

Doubts

Compilation Error

when Syntax is wrong

→ missed ;

→ missed { }

→ int = 5;

Runtime Error

when logical mistake is there

5/0

I am a chair.

[0-50]	first 50 units	0·50
[51-150)	next 100	0·75
[151-250]	next 100	1·2
251 & above	above 250	1·5

```
int units = scn.nextInt();
double amount = 0

if (units <= 50) {

        amount = 0·50 * units;
}
else if (units ≤ 150){

        amount = 25+ (units-50)*0·75;

}
```

20%

50*0·5 = 25

```
else if (units ≤ 250){

    amount = 100 + (units- 150) * 1·2;
}
 else {

    amount = 220 + (units-250)* 1·5;
}
```

$25 + 100 * 0.75$

$25 + 75 = 100$

```
// 20% surcharge
amount = 1·2*amount;
S.O.P     ((int) amount);
```

$x + 20\% x$

↳ $x + \left(\frac{20}{100}\right) x$

↳ $x + 0.2x$

↳ $x(1+0.2)$

↳ $\boxed{1.2x}$

```
int x = 10
                   → true
if (x ≥ 10) {
      → S.O.P ("Yes");                  Yes
} else if (x > 5) {
         S.O.P ("No");
}
```

```
        n1              n2              n3

if ( n1≥ n2 && n1≥ n3 ){
        // n1 is largest
        S.O.P (n1);

} else {

        if (n2> n3) {
                S.O.P (n2);
        } else {
                S.O.P (n3);
        }

}
```

```
int a=5 , b = 6                    5 == 5

if (a++ == --b) {
    S.O.P ("5 = 5");

} else {
    S.O.P ("NONE");
}
```

Post Increment → First use the current value and then update

Pre Incr / Decr → First update then use

$t = 4.5$

$$\frac{4.5 \times 0.2}{0.9}$$

$$t = \Big(\underbrace{(\text{int})t}_{4} + \underbrace{(t * 0.20)}_{0.9}\Big);$$

$$t = 4.9$$

$$t = (\text{int})\Big(\underbrace{t}_{4.5} + \underbrace{(t * 0.20)}_{0.9}\Big)$$

$$(4.5 + 0.9)$$

$$\underline{(\text{int})(5.4)}$$

$$t = 5$$

$$\underset{t}{\boxed{5.0}}$$

T = 150
N = 100

$\left(\frac{100*100}{150}\right) \rightarrow 66$ 75

```
double T = scn.nextInt();
 double N = Scn.nextInt();

double attend = (N*100)/T;
  if (attend >= 75){
        S.O.P(Yes)

 } else {
        S.O.P(No)
   }
```