Steps till n greater than 0

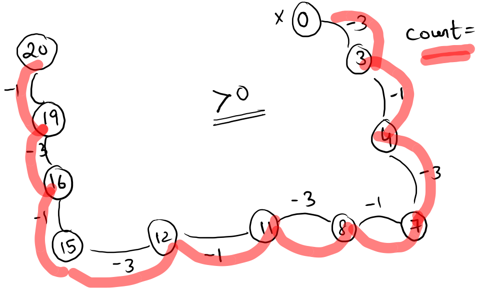
eg. n = 20

even → n-1 ✓

odd → n-3 ✓

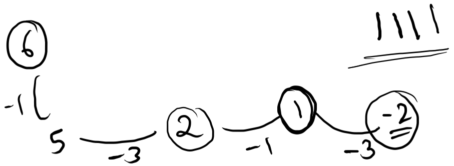
⇒ total no. of steps

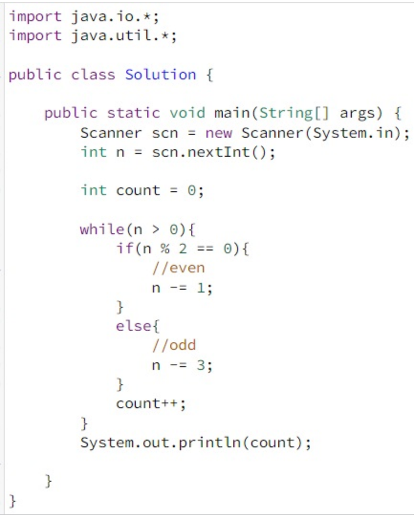
while.



count = ~~0~~ ~~1~~ ~~2~~ ~~3~~ ~~4~~ ~~5~~ ~~6~~ ~~7~~ ~~8~~ ~~9~~ 10

eg.





```
import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();

        int count = 0;

        while(n > 0){
            if(n % 2 == 0){
                //even
                n -= 1;
            }
            else{
                //odd
                n -= 3;
            }
            count++;
        }
        System.out.println(count);

    }
}
```

nth power of 10 using while loop

while

? { n=2 ↳ 100 10×10

n=3 ↳ 1000 10×10×10

n=4 ↳ 10000 10×10×10×10

n=6 1 000000

n=0 $\boxed{1}$ }

? { 10^n

n=2 ↳ $10^2 = 100$

n=4 $10^4 = 10,000$

n=0 $10^0 = 1$ }

$x^y \Rightarrow 2^3 = 8$

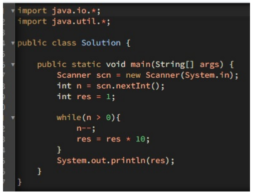
p = { Math.pow(x,y) = x^y

Math.pow(2,3) = 8 }

n = ~~3~~ ~~2~~ ~~1~~ 0

res = ~~1~~ ~~10~~ ~~100~~ $\boxed{1000}$ (result) → 1000

```
while (n > 0)          2>0
{
    res *= 10;         1>0
    n--;               0>0
}
```



```
import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();
        int res = 1;

        while(n > 0){
            n--;
            res = res * 10;
        }
        System.out.println(res);
    }
}
```

n = ~~3~~ ~~2~~ ~~1~~ 0

res = ~~1~~ ~~10~~ ~~100~~ $\boxed{1000}$

eg. i n=3

o 1000 ↑

3>0

2>0

1>0

~~$\boxed{0>0}$~~

$\boxed{0>0}$

Print nth Tribonacci number

T_0 a = 0

T_1 b = 1

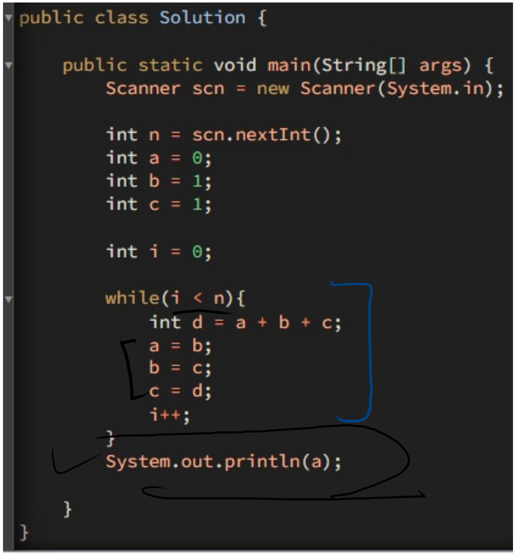
T_2 c = 1

$$\boxed{T_n = T_{n-1} + T_{n-2} + T_{n-3}}$$

for (i=0 ______ <n)

0 1 1 2 4 7 13 24....

0 1 2 3 4 5 6 7



```java
public class Solution {

    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);

        int n = scn.nextInt();
        int a = 0;
        int b = 1;
        int c = 1;

        int i = 0;

        while(i < n){
            int d = a + b + c;
            a = b;
            b = c;
            c = d;
            i++;
        }
        System.out.println(a);

    }
}
```

i/p → 6

n=6

$a = \cancel{0}\ \cancel{1}\ \cancel{1}\ \cancel{2}\ \cancel{4}\ \cancel{7}\ 13$

$b = \cancel{1}\ \cancel{1}\ \cancel{2}\ \cancel{4}\ \cancel{7}\ \cancel{13}\ 24$

$c = \cancel{1}\ \cancel{2}\ \cancel{4}\ \cancel{7}\ \cancel{13}\ \cancel{24}\ 44$

$i = \cancel{0}\ \cancel{1}\ \cancel{2}\ \cancel{3}\ \cancel{4}\ \cancel{5}\ 6$

0<6

1<6

2<6

3<6

4<6

5<6

$\boxed{6<6}$

d = 44

0 1 1 2 4 7 13 24 44

Print all digits from end

Q?

1 2 3 4 5

↳ 5
4
3
2
1

123

3
2
1

987

7
8
9

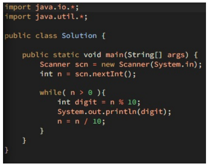
while
revise.
/ %.

int 1 2 3 4 5

while
loop {
Syso
}

1 2 3 4 5 %10 → 5
digit = 5
/10
1 2 3 4 %10 → 4
/10
1 2 3 %10 → 3
/10
12 % 10 → 2
/10
1 % 10 → 1
/10
0 stop.

Code. ?



```java
import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();

        while( n > 0 ){
            int digit = n % 10;
            System.out.println(digit);
            n = n / 10;
        }
    }
}
```

n = 1234

1234 > 0
digit = 4

123 > 0
digit = 3

12 > 0
digit = 2

1/10 = 0

1 > 0
digit = 1

4
3
2
1

0 > 0

10) 1234 (123
123
4

1234 % 10 = 4.

1234 / 10 = 123

String. sol.

int n = 1234;

String s = "" + n;

s = "1234";
0 1 2 3

len = 4

i = ~~3~~ ~~2~~ ~~1~~ 0

for (int i = s.length()-1 ; i ≥ 0 ; i--)
{
Syso (s. charAt(i));
}

3 ≥ 0
2 ≥ 0
1 ≥ 0
0 ≥ 0

4
3
2
1

doubt

int n = 4236

4

String s = "" + n;

s.length()

.length()