

Print 4,13,22,31.....n

Series:- 4, 13, 22, 31, 40, 49,

↳ from 4 to n by +9

```
// from 4 to n by +9
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();

    int i = 4;
    while (i <= n) {
        System.out.println(i);
        i += 9;
    }
}
```

Print n, n-k, n-2k, n-3k.... till l

i/p :- $n = 50;$

$k = 5;$

$l = 4;$

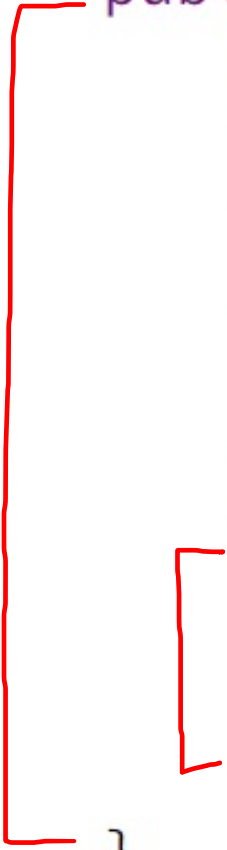
Series:- 50, 45, 40, 35, 30, 25, 20, 15, 10, 5

↳ from n to l by -k

pseudo
code

```
int i = n;  
while ( i >= l ) {  
    Syso(i);  
    i -= k;  
}
```

```
public static void main(String[] args) {  
    Scanner scn = new Scanner(System.in);  
    int n = scn.nextInt();  
    int k = scn.nextInt();  
    int l = scn.nextInt();  
  
    int i = n;  
    while (i >= l) {  
        System.out.println(i);  
        i -= k;  
    }  
}
```

A red bracket on the left side of the code block spans from the opening curly brace of the main method to its closing curly brace, indicating the scope of the entire method. A second red bracket is positioned to the left of the while loop, spanning from its opening curly brace to its closing curly brace, indicating the scope of the loop.

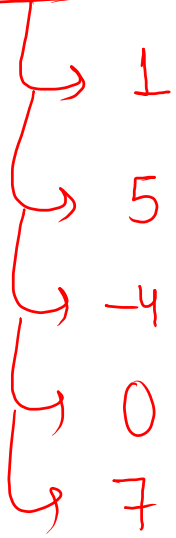
Running Sum for loop

Series :-	2	5	-4	-7	0	12
	↓	↓	↓	↓	↓	↓
running sum :-	2	7	3	-4	-4	8

(sum of all the previous numbers including itself)

Running Sum for loop

$n = 5$



running sum

1

6

2

2

9

dry run

```
public static void main(String[] args) {  
    → Scanner scn = new Scanner(System.in);  
    → int n = scn.nextInt();  
  
    → int sum = 0;  
    for (int i = 0; i < n; i++) {  
        → int num = scn.nextInt();  
        → sum += num;  
        → System.out.print(sum + " ");  
    }  
}
```

n = 5 ;

sum = ~~0~~ ~~4~~ ~~8~~ ~~-2~~ ~~-2~~ 1

i = 0, (0 < 5) ✓ num = 4

i = 1, (1 < 5) ✓ num = 2

i = 2, (2 < 5) ✓ num = -8

i = 3, (3 < 5) ✓ num = 0

i = 4, (4 < 5) ✓ num = 3

i = 5, (5 < 5) X

qp

4	6	-2	-2	1
---	---	----	----	---

\Rightarrow Fibonacci Series

\hookrightarrow every no. is sum of previous 2 numbers

first no. = 0
second no. = 1 (original)

series - 0, 1, 1, 2, 3, 5, 8, 13, 21, ∞

Nth Fibonacci Number 7

$$\underline{\underline{f(1) = 1, f(2) = 1}}$$

series:-

1	1	2	3	5	8	13	21	34	...
1	2	3	4	5	6	7	8	9	

i/p:- $n = 9$,

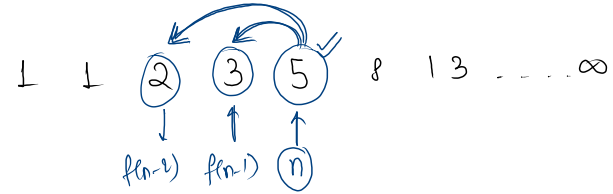
logic:-

use 2 variables to store previous 2 no.'s

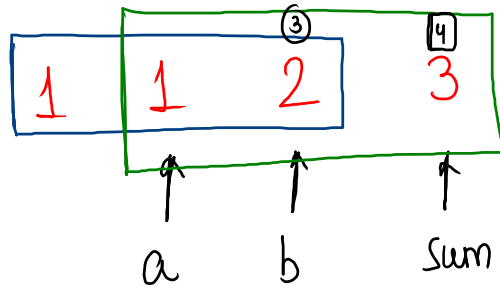
and find next sum using those

and keep the series moving until you find n^{th} term.

$$f(n) = \underline{\underline{f(n-1)}} + \underline{\underline{f(n-2)}}$$



Series:-



5 8 13 21 34

$$\left\{ \begin{array}{l} \text{sum} = a + b ; \\ a = b ; \\ b = \text{sum} ; \end{array} \right.$$

times :-	term
1	3
2	4
3	5
⋮	⋮
<u>(n-2)</u>	n th

code

```
public static void main(String[] args) {  
    Scanner scn = new Scanner(System.in);  
    int n = scn.nextInt();  
  
    if (n == 1) {  
        System.out.println(1);  
    } else if (n == 2) {  
        System.out.println(1);  
    } else {  
        int a = 1;  
        int b = 1;  
        int sum = 0;  
        for (int i = 3; i <= n; i++) {  
            sum = a + b;  
            a = b;  
            b = sum;  
        }  
        System.out.println(sum);  
    }  
}
```