

1 Apr 2023

### Combination Formula

$${}^nC_r = \frac{n!}{(n-r)!r!}$$

${}^nC_r$ :

$${}^nC_r: {}^5C_2 = \frac{5!}{(5-2)! \times 2!}$$

$$= \frac{5!}{3! \times 2!}$$

$$= \frac{5 \times 4 \times \cancel{3!}}{\cancel{3!} \times 2!}$$

$$= \frac{5 \times \cancel{4}^2}{\cancel{2} \times 1}$$

$$= 10$$

$$\therefore {}^5C_2 = 10$$

Pgm: Enter n } Scanner  
Enter r }

int nCr;  
 $nCr = (factorial(n) / (factorial(n-r) * factorial(r)))$

```
J Main.java 1 x
J Main.java > Main > main(String[])
23 //
24
25 // WAP to calculate nCr
26 import java.util.*;
27 public class Main
28 {
29     static int factorial(int n)
30     {
31         if (n==0)
32             return 1;
33         else
34             return n*factorial(n-1);
35     }
36
37     public static void main(String[] args)
38     {
39         int n, r, nCr;
40         Scanner sc = new Scanner(System.in);
41         n = sc.nextInt(); 5
42         r = sc.nextInt(); 2
43         nCr = (factorial(5)) / (factorial(5-2)*factorial(2));
44         System.out.println("nCr is "+nCr);
45     }
46
47 }
48 }
```

Run | Debug

① factorial(5) = 5  
factorial(3) \* factorial(2)  
② 3 ③ 2  
= 120 / (6 \* 2)  
= 10

## Fibonacci Series:

0 1 1 2 3 5 8 13 21

```
class Main
{
    psum(—|—)
    {
        int n1=0, n2=1, n3, i;
        Sop(n1+" "+n2);

        for(i=2; i<10; i++)
        {
            n3 = n1+n2, Sop(" "+n3);
            n1 = n2;
            n2 = n3;
        }
    }
}
```

o/p: <sup>n1</sup> <sup>n2</sup>  
~~n1~~ ~~n2~~ ~~n3~~ n3  
0 1 1 2 3 5 13 21 34

i = ~~2~~ ~~3~~ ~~4~~ ~~5~~ ~~6~~ ~~7~~ ~~8~~ ~~9~~ 10  
n1 = ~~0~~ ~~1~~ ~~1~~ ~~2~~ ~~3~~ ~~5~~ ~~13~~ 21  
n2 = ~~1~~ ~~1~~ ~~2~~ ~~3~~ ~~5~~ ~~13~~ 21 34  
n3 = ~~1~~ ~~2~~ ~~3~~ ~~5~~ ~~13~~ 21 34

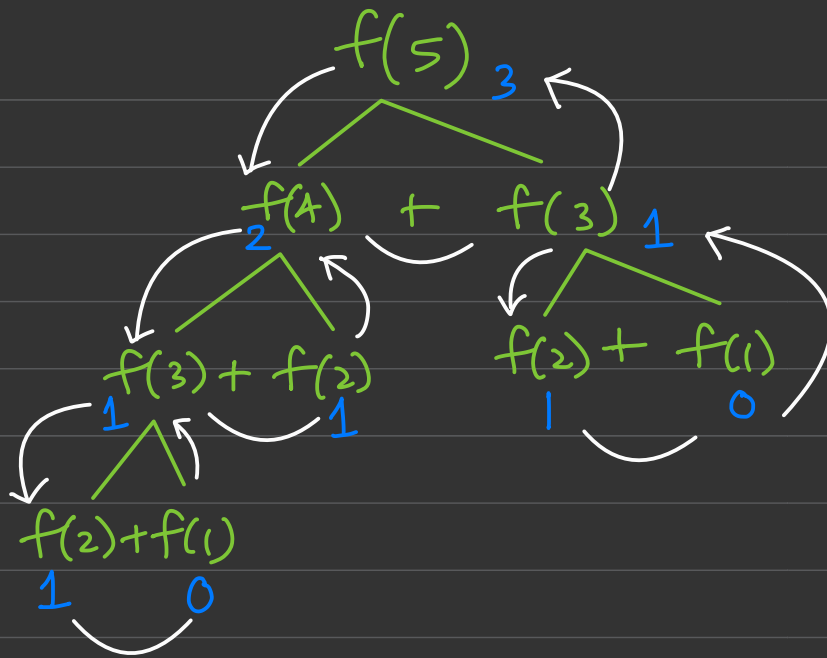
## Fibonacci Series using Recursion:

0 1 1 2 3 5 8 13 21  
f(1) f(2) f(3) f(4) f(5) f(6) f(7) f(8)

$$\begin{aligned} \textcircled{1} \quad f(n) &= f(n-1) + f(n-2) \\ f(3) &= f(2) + f(1) \\ f(4) &= f(3) + f(2) \end{aligned}$$

Repeat  $\textcircled{1}$  till  $n \leq 3$

If  $n < 3$ , then  $f(n) = (n-1)$   
eg: If  $n = 2$ , then  $f(2) = 1$   
If  $n = 1$ , then  $f(1) = 0$



```

J Main.java 1 •
J Main.java > Main > main(String[])
75
76 static int fibo(int n)
77 {
78     if (n<3)
79         return (n-1);
80     else
81         return fibo(n-1)+fibo(n-2);
82 }
83
Run | Debug
84 public static void main(String[] args)
85 {
86     int i, no_of_terms;
87     Scanner sc = new Scanner(System.in);
88     no_of_terms = sc.nextInt(); // 5
89     int res;
90     for(i=1;i<=no_of_terms;i++)
91     {
92         res = fibo(i);
93         System.out.print(res + " ");
94     }
95 }

```

Handwritten notes on the code:

- Yellow arrow from `no_of_terms = 5` to `for(i=1; i<=no_of_terms; i++)`.
- Yellow arrow from `res = fibo(i);` to `for(i=1; i<=no_of_terms; i++)`.
- Yellow arrow from `System.out.print(res + " ");` to `for(i=1; i<=no_of_terms; i++)`.
- Yellow arrow from `no_of_terms = 5` to `for(i=1; i<=no_of_terms; i++)`.
- Yellow arrow from `res = fibo(i);` to `for(i=1; i<=no_of_terms; i++)`.
- Yellow arrow from `System.out.print(res + " ");` to `for(i=1; i<=no_of_terms; i++)`.
- Yellow arrow from `no_of_terms = 5` to `for(i=1; i<=no_of_terms; i++)`.
- Yellow arrow from `res = fibo(i);` to `for(i=1; i<=no_of_terms; i++)`.
- Yellow arrow from `System.out.print(res + " ");` to `for(i=1; i<=no_of_terms; i++)`.

Handwritten notes on the right side of the code editor:

- `i = 1 2 3 4 5 6`

Terminal output:

```

5
0 1 1 2 3 %
(base) ingledarshan@192 NS 01 Apr 2023 %

```

NS 01 Apr 2023

```
J Main.java 1 •
J Main.java > Main > main(String[])
75
76 static int fibo(int n)
77 {
78     if (n<3)
79         return n-1;
80     else
81         return fibo(n-1)+fibo(n-2);
82 }
83
Run | Debug
84 public static void main(String[] args)
85 {
86     int i, no_of_terms;
87     Scanner sc = new Scanner(System.in);
88     no_of_terms = sc.nextInt(); // 5
89     int res;
90     for(i=1;i<=no_of_terms;i++)
91     {
92         res = fibo(i);
93         System.out.print(res + " ");
94     }
95 }
```

Handwritten notes and diagrams:

- Yellow arrows and numbers (1, 2, 3, 4, 5) indicating recursive calls to `fibo` from the `main` method.
- Handwritten Fibonacci sequence: 0, 1, 1, 2, 3, 5.
- Handwritten tree diagram for  $f(5)$ :  
 $f(5) = f(4) + f(3)$   
 $f(4) = f(3) + f(2)$   
 $f(3) = f(2) + f(1)$   
 $f(2) = f(1) + f(0)$

TERMINAL

```
5
0 1 1 2 3 %
(base) ingledarshan@192 NS 01 Apr 2023 %
```

Ln 88, Col 40 Tab Size: 4 UTF-8 LF {} Java

H/w: Java Program to Find the Sum of Natural Numbers using Recursion [easy]

<https://www.programiz.com/java-programming/examples/sum-natural-numbers-recursion>

[https://www.youtube.com/watch?v=Cv\\_IgVhYbFs&ab\\_channel=ProgrammingForBeginners](https://www.youtube.com/watch?v=Cv_IgVhYbFs&ab_channel=ProgrammingForBeginners)

\_\_\_\_\_ X \_\_\_\_\_