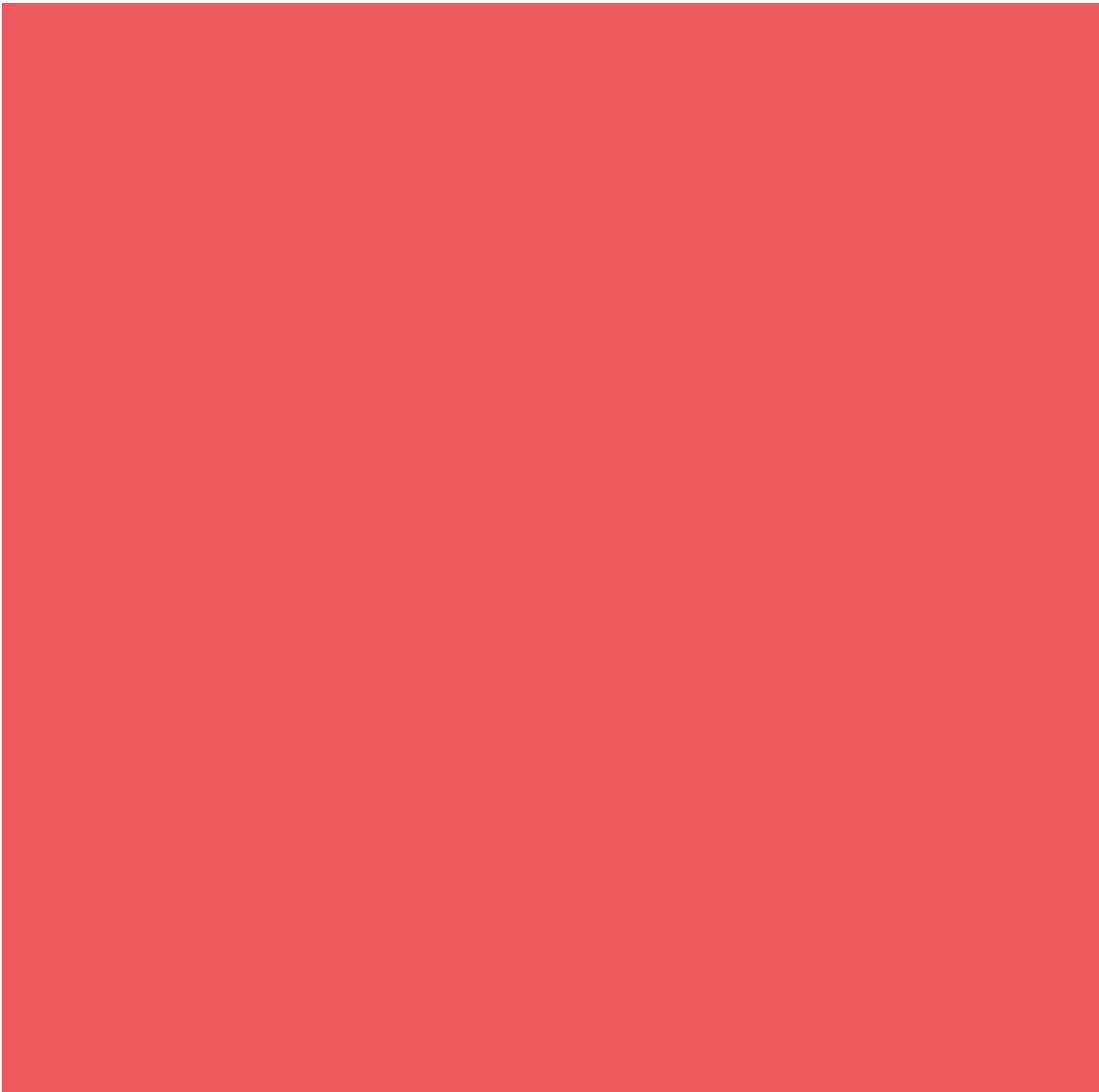
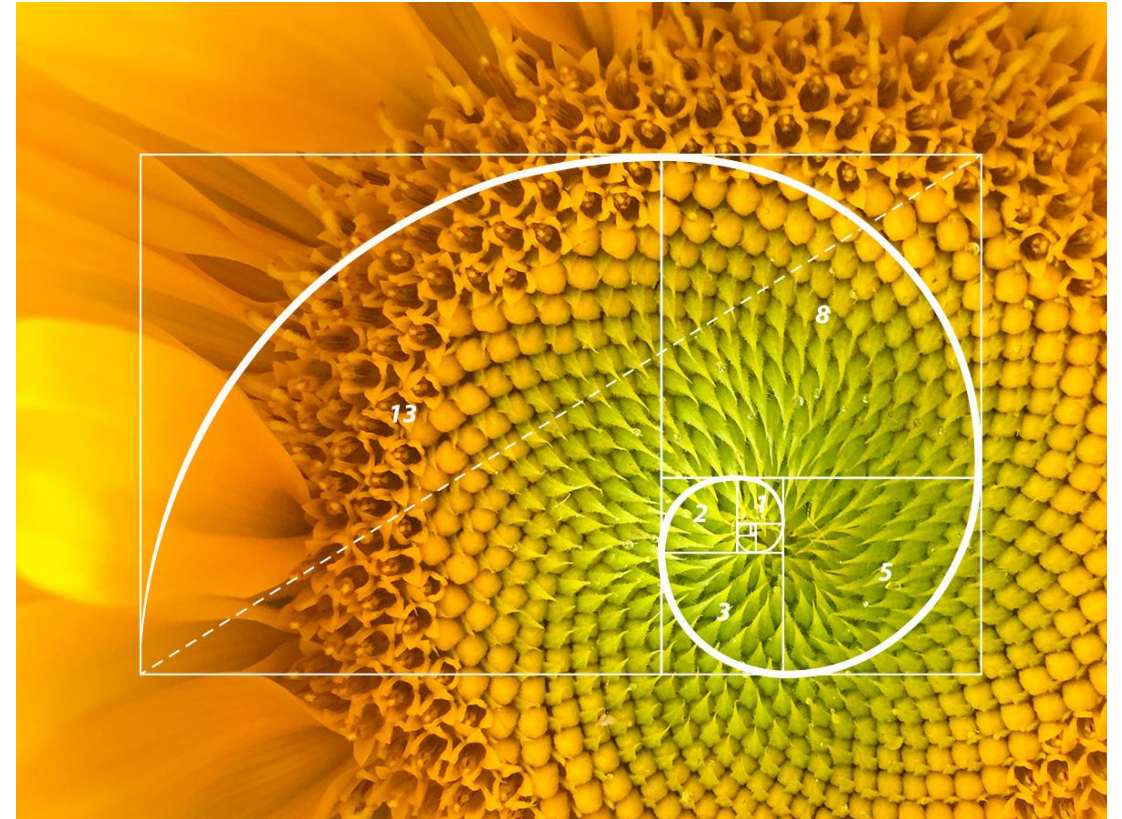


Recursive Tracing





Iterative Factorial

```
public static int iterativeFactorial(int n) {  
    int result = 1;  
    for (int i = 2; i <= n; i++) {  
        result *= i;  
    }  
    return result;  
}
```

$\text{result} = (1 * 2) = (2 * 3) = (6 * 4) = 24$

$i = 4$

Recursive Factorial

```
public static int recursiveFactorial(int 4 {  
    public static int recursiveFactorial(int 3 {  
        public static int recursiveFactorial(int 2 {  
            public static int recursiveFactorial(int 1 {  
                if (n == 1) {  
                    return 1;  
                } else {  
                    return n * recursiveFactorial(n - 1);  
                }  
            }  
        }  
    }  
}
```

Recursive Factorial

```
public static int recursiveFactorial(int 4 {  
    public static int recursiveFactorial(int 3 {  
        public static int recursiveFactorial(int 2 {  
            public static int recursiveFactorial(int 1 {  
                if (n == 1) {  
                    return 1; ← Base Case!  
                } else {  
                    return n * recursiveFactorial(n - 1);  
                }  
            }  
        }  
    }  
}
```



Recursive Factorial

```
public static int recursiveFactorial(int 4 {  
    public static int recursiveFactorial(int 3 {  
        recursiveFactorial(1) {  
            public static int recursiveFactorial(int 2 {  
                public static int recursiveFactorial(int 1 {  
                    if (n == 1) {  
                        return 1;  
                    } else {  
                        return n * recursiveFactorial(n - 1);  
                    }  
                }  
            }  
        }  
    }  
}
```

= 24