**Print N to 1 Using Recursion**

Given a number **N**, the task is to print the numbers from **N to 1**.  
**Examples:**

***Input:****N = 10*  
***Output:****10 9 8 7 6 5 4 3 2 1*  
***Input:****N = 7*  
***Output:****7 6 5 4 3 2 1*

**Approach 1:** Run a loop from N to 1 and print the value of N for each iteration. Decrement the value of N by 1 after each iteration.  
Below is the implementation of the above approach.

Java

// Java program to print all numbers between 1

// to N in reverse order

import java.util.\*;

class GFG {

// Recursive function to print from N to 1

static void PrintReverseOrder(int N)

{

for (int i = N; i > 0; i--)

System.out.print( +i + " ");

}

// Driver code

public static void main(String[] args)

{

int N = 5;

PrintReverseOrder(N);

}

}

**Output**

5 4 3 2 1

***Time Complexity:****O(N)*

***Auxiliary Space:****O(1)*

**Approach 2:** We will use [recursion](https://www.geeksforgeeks.org/recursion/) to solve this problem.

1. Check for the base case. Here it is N<=0.
2. If base condition satisfied, return to the main function.
3. If base condition not satisfied, print N and call the function recursively with value (N – 1) until base condition satisfies.

Below is the implementation of the above approach.

Java

// Java program to print all numbers

// between 1 to N in reverse order

class GFG{

// Recursive function to print

// from N to 1

static void PrintReverseOrder(int N)

{

// If N is less than 1 then

// return static void function

if (N <= 0)

{

return;

}

else

{

System.out.print(N + " ");

// Recursive call of the function

PrintReverseOrder(N - 1);

}

}

// Driver code

public static void main(String[] args)

{

int N = 5;

PrintReverseOrder(N);

}

}

**Output**

5 4 3 2 1

**Time Complexity:**O(N)

**Auxiliary Space:**O(N)