Print 1 to N using Recursion

If we take a look at this problem carefully, we can see that the idea of "loop" is to track some counter value e.g. "i=0" till "i <= 100". So if we aren't allowed to use loop, how else can be track something in C language!

Well, one possibility is the use of 'recursion' provided we use the terminating condition carefully. Here is a solution that prints numbers using recursion.

Method-1:

```
// C++ program to How will you print
//  numbers from 1 to 100 without using loop?
#include <iostream>
using namespace std;

class gfg{

// Prints numbers from 1 to npublic:
void printNos(unsigned int n)
{
    if(n > 0)
    {
        printNos(n - 1);
        cout << n << " ";
    }
    return;
}
};
// Driver code
int main()
{
    gfg g;
    g.printNos(100);
    return 0;
}
```

Output :

```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
16 17 18 19 20 21 22 23 24 25 26 27
28 29 30 31 32 33 34 35 36 37 38 39
40 41 42 43 44 45 46 47 48 49 50 51
52 53 54 55 56 57 58 59 60 61 62 63
64 65 66 67 68 69 70 71 72 73 74 75
76 77 78 79 80 81 82 83 84 85 86 87
88 89 90 91 92 93 94 95 96 97 98 99
100
```

Time Complexity: *O(n)*

Auxiliary Space: *O(n)*

Method 2:

```
// C++ program
#include <bits/stdc++.h>
using namespace std;

void printNos(int initial, int last)
{
    if (initial <= last) {
        cout << initial << " ";
        printNos(initial + 1, last);
    }
}

int main()
{
    printNos(1, 100);
    return 0;
}
```

Output

```
 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33
34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63
64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93
94 95 96 97 98 99 100
```

Time Complexity : O(n)

Auxiliary Space: *O(n)*

Now try writing a program that does the same but without any "if" construct.

Hint -- use some operator which can be used instead of "if".

Please note that recursion technique is good but every call to the function creates one "stack-frame" in program stack. So if there's constraint to the limited memory and we need to print large set of numbers, "recursion" might not be a good idea. So what could be the other alternative?

Another alternative is "goto" statement. Though use of "goto" is not suggestible as a general programming practice as "goto" statement changes the normal program execution sequence yet in some cases, use of "goto" is the best working solution.