Ans 1 - Given a number n, print the following pattern without using any loop.

n, n-5, n-10, ..., 0, 5, 10, ..., n-5, n

There should be 0 or at most one occurrence of negative number in the series.

.vscode > recursion8.cpp

```cpp
#include<iostream>
using namespace std;
void decrement_function(int n,int decrement){
    if(n<0) return;
    cout<<n<<" ";
    decrement_function(n-decrement,decrement);
    cout<<n<<" ";
}
int main(){
    decrement_function(20,5);
    return 0;
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMI

```
PS D:\cpprograme> cd "d:\cpprograme\.vs
($?) { .\recursion8 }
20 15 10 5 0 0 5 10 15 20
PS D:\cpprograme\.vscode>
```

Q2 - Find m-th summation of first n natural numbers where m-th summation of first n natural numbers is defined as following:

If m > 1: SUM(n, m) = SUM(SUM(n, m - 1), 1)

Else :SUM(n, 1) = Sum of first n natural numbers.

.vscode > recursion8.cpp

```cpp
#include<iostream>
using namespace std;
int sum(int n, int m) {
    if (m == 1) {
        return (n * (n + 1)) / 2;
    } else {
        return sum(sum(n, m-1), 1);
    }
}

int main(){
    int result=sum(5,3);
    cout<<result;
    return 0;
}
```

Q3 - Given a number n which denotes the number of variables in the equation and a val which denotes the sum of these variables, count the number of such non-negative integral solutions that are possible.

Sample Input: n=5 val=1

Sample Output: 5

.vscode > recursion9.cpp

```cpp
#include<iostream>
using namespace std;
int countSolutions(int n, int val) {
    if (n == 0) {
        return (val == 0) ? 1 : 0;
    }
    int count = 0;
    for (int x = 0; x <= val; x++) {
        count += countSolutions(n-1, val-x);
    }
    return count;
}

int main(){
    cout<<countSolutions(5,1);
    return 0;
}
```

```
PS D:\cpprograme> cd "d:\cpprograme\.vscode
; if ($?) { .\recursion9 }
5
PS D:\cpprograme\.vscode>
```