

```

#include
using namespace std;

int ack(int m, int n)
{
    if (m == 0){
        return n + 1;
    }
    else if((m > 0) && (n == 0)){
        return ack(m - 1, 1);
    }
    else if((m > 0) && (n > 0)){
        return ack(m - 1, ack(m, n - 1));
    }
    return 0;
}

int main()
{
    int A;
    A = ack(1, 2);
    cout << A << endl;
    return 0;
}

```

Output:

4

In this case, to solve the query of $\text{ack}(1,2)$ it takes a large number of recursive steps, where the time complexity is actually **$O(\text{m} \cdot \text{ack}(\text{m}, \text{n}))$** to compute $\text{ack}(\text{m}, \text{n})$. Whereas the space complexity is **$O(\text{m})$** to compute $\text{ack}(\text{m}, \text{n})$.