Uva 11420

Chest of Drawers

Time: 3 seconds

Problem Descriptions (1/3)

- A chest of drawers means a wardrobe which has many drawers aligned vertically as shown in the figure on the left.
- Although this is useful furniture, some problems arise when all the drawers need have provisions of locking - that is sometimes a drawer is not secured even if it is locked.
- For example, assume that the third drawer from the top is locked but the drawer immediately above it is not locked.

Problem Descriptions (2/3)

- Then the drawer that is locked is also not secured because one can access it by pulling out the drawer immediately above it.
- In a chest of *n* drawers, there are a number of ways to ensure that exactly *s* drawers are secure.
- For example for the chest of drawers shown on the left, exactly four drawers can be secured in six ways.
- These six ways are shown in Figure 2.
- Given the value of *n* and *s*, your job is to find out in how many ways they can be secured.

Problem Descriptions (3/3)

U	**L**	**L**	**L**	**L**	**L**
L	U	**L**	**L**	**L**	**L**
L	L	U	**L**	**L**	**L**
L	**L**	L	U	**L**	**L**
L	**L**	**L**	L	U	U
L	**L**	**L**	**L**	L	U

- **In this gure L means that the drawer is locked and U means that the corresponding drawer is unlocked.**
- **And here all six locking combinations are shown which ensures that exactly four drawers aresecured.**
- **Letters corresponding the secured drawers are boldfaced.**

Input (1/3)

- The input contains at most 5000 lines of inputs.
- Each line contains two integers *n* and *s* (1<=*n*<=65) and 0<=*s*<=65).
- Here *n* is the total number of drawers and *s* is the number of drawers that needs to be secured.
- Input is terminated by a line containing two negative numbers. This input should not be processed.

Output

- For each line of input produce one line of output.
- This line contains an integer which denotes in how many, *s* drawers out of the *n* drawers can be secured.

Sample I/O

n s

6 2
6 3
6 4
-1 -1

End of input

16
9
6

U	L	L	L	L	L
L	U	L	L	L	L
L	L	U	L	L	L
L	L	L	U	L	L
L	L	L	L	U	U
L	L	L	L	L	U

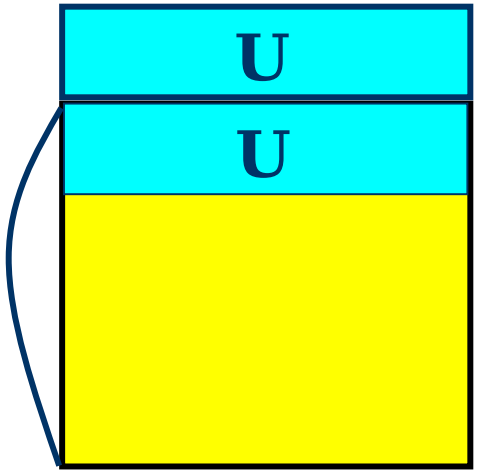
DP State Definition

- State:
  - dp[n][s][0]
    - ✓ n 個抽屜， s 個是安全的，且最上方的抽屜是 unLock
  - dp[s][s][1]
    - ✓ n 個抽屜， s 個是安全的，且最上方的抽屜是 Lock

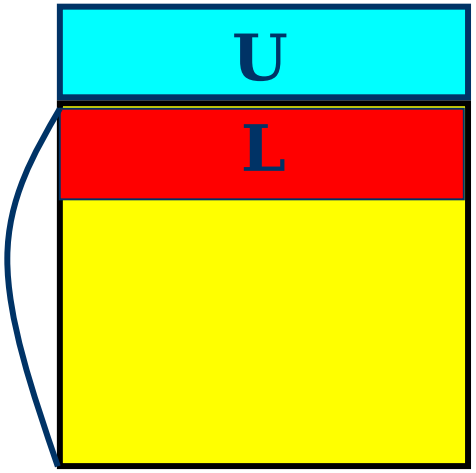
DP State Definition

- **State Transfer:**
  - **dp[n][s][0] = dp[n-1][s][0]+dp[n-1][s+1][1]**

dp[n-1][s][0]



dp[n-1][s+1][1]

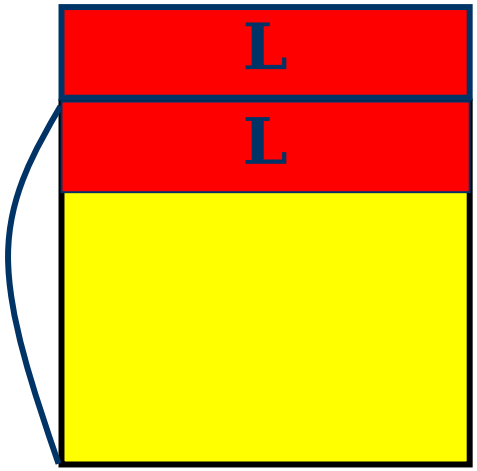


DP State Definition

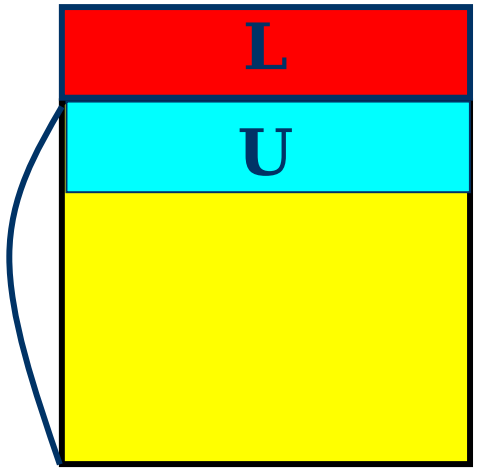
State Transfer:

- dp[n][s][1] = dp[n-1][s-1][1] + dp[n-1][s-1][0]

dp[n-1][s-1][1]



dp[n-1][s-1][0]



```cpp
#include <cstdlib>
#include <cstdio>
#define maxn 66+5
FILE *fPtr;

int main()
{
    freopen("d:\\11420_in.txt","r",stdin);
    fPtr=freopen("d:\\11420_out.txt","w",stdout);

    int N, S;
    long long dp[maxn][maxn][2]={};

    dp[1][0][0]=1; dp[1][1][1]=1;

    for (int n=2; n<maxn; n++)
    {
        dp[n][0][0]=dp[n-1][1][1]+dp[n-1][0][0]; //Initialization

        for (int s=1; s<=n; s++)
        {
            dp[n][s][0]=dp[n-1][s+1][1]+dp[n-1][s][0];
            dp[n][s][1]=dp[n-1][s-1][1]+dp[n-1][s-1][0];
        }
    }

    while (scanf("%d%d",&N,&S) && (N>=0 || S>=0))
        printf("%lld\n",dp[N][S][0]+dp[N][S][1]);

    fclose(fPtr);
    return 0;
}
```