

## Nurse

---

- The director of a hospital want to schedule a working plan for a nurse in a given period of  $N$  consecutive days  $1, \dots, N$ . Due to the policy of the hospital, each nurse cannot work all the days  $1, \dots, N$ . Instead, there must be days off in which the nurse need to take a rest. A working plan is a sequence of disjoint working periods. A working period of a nurse is defined to be a sequence of consecutive days on which the nurse must work and the length of the working period is the number of consecutive days of that working period. The hospital imposes two constraints:
  - Each nurse can take a rest only one day between two consecutive working periods. it means that if the nurse takes a rest today, then she has to work tomorrow (1)
  - The length of each working period must be greater or equal to  $K1$  and less than or equal to  $K2$  (2)
  - The director of the hospital want to know how many possible working plans satisfying above constraint?
- **Input**
  - The input consists of one line which contains 3 positive integers  $N, K1, K2$  ( $2 \leq N \leq 1000, K1 < K2 \leq 400$ )
- **Output**
  - The output consists of only one single integer  $M$  modulo  $10^9+7$  where  $M$  is the total working plans satisfying the above constraints.

### Đề bài:

Một giám đốc của một bệnh viện muốn lập lịch làm việc cho y tá trong  $N$  ngày  $1 \dots N$ .

Y tá chỉ có thể làm việc liên tục trong  $x$  ngày ( $K1 \leq x \leq K2$ ), sau đó phải nghỉ 1 ngày.

- Y tá chỉ được nghỉ 1 ngày giữa 2 đợt làm việc liên tiếp. Có nghĩa rằng nếu y tá nghỉ hôm nay, ngày mai y tá đó sẽ phải làm việc. ngày

### Yêu cầu:

Tính số cách có thể lập lịch làm việc cho các y tá.

# Nurse

---

**Example:**

**Input:**

6 2 3

**Output:**

4

## Nurse – Dynamic Programming Algorithm

---

- Xác định bài toán con:
  - Gọi  $S0[i]$  là số cách lập lịch đến ngày thứ  $i$  và ngày thứ  $i$  là ngày nghỉ.
  - Gọi  $S1[i]$  là số cách lập lịch đến ngày thứ  $i$  và ngày thứ  $i$  là ngày làm việc.
- Khởi tạo:
  - $S0[i] = S1[i] = 0$ , với mọi  $i = 1, \dots, n$
  - $S0[1] = 1, S1[k1] = 1, S0[0] = 1$
- Công thức quy hoạch động:
  - $S0[i] = S1[i-1];$
  - $S1[i] = \sum_{j=k1}^{k2} S0[i-j];$
- Kết quả:
  - $S0[n] + S1[n];$
- Độ phức tạp:
  - $O(n^2).$

## Implementation – main process

```
#include <bits/stdc++.h>
#define MAX 2000
using namespace std;
int N, K1, K2;
void solve(){
    int S0[MAX], S1[MAX];
    for(int i = 1; i <= N; i++){        S0[i] = 0; S1[i] = 0;    }
    S0[1] = 1;    S1[K1] = 1;    S0[0] = 1;

    for(int i = K1+1; i <= N; i++){
        S0[i] = S1[i-1];        S1[i] = 0;
        for(int j = K1; j <= K2; j++){
            if(i-j >= 0)
                S1[i] = S1[i] + S0[i-j];
        }
    }
    int rs = S0[N] + S1[N];
    cout << rs;
}
```

## Implementation – main process

```
void input(){
    cin >> N >> K1 >> K2;
}

int main(){
    input();
    solve();
}
```