```
1   //catalan path
2   int toPath(point from,point to){
3       int total=mul(fac[to.x-from.x+to.y-from.y],invFac[to.x-from.x]);
4       total=mul(total,invFac[to.y-from.y]);
5       if(min(from.x-c,b)<to.y){
6           int mns;
7           int hor=to.x-from.x;
8           int ver=to.y-from.y;
9           point des;
10          des.x=ver-min(from.x-c,b)+from.y-1+from.x;
11          des.y=min(from.x-c,b)+1+hor;
12          mns=mul(fac[des.x-from.x+des.y-from.y],invFac[des.x-from.x]);
13          mns=mul(mns,invFac[des.y-from.y]);
14          total=add(total,(-1)*mns);
15      }
16      return total;
17  }
18  //n^2logn solution for chaining
19  int add(int _a, int _b){
20      _a = (_a + MOD) % MOD;
21      _b = (_b + MOD) % MOD;
22      return (_a + _b) % MOD;
23  }
24  int mul(int _a, int _b){
25      _a = (_a + MOD) % MOD;
26      _b = (_b + MOD) % MOD;
27      return ((ll)((ll)_a * (ll)_b)) % MOD;
28  }
29  int big_mod(int v, int p){
30      if(p == 0){ return 1; }
31      int ret = big_mod(v, p / 2);
32      if(p % 2 == 0){ return mul(ret, ret); }
33      else{ return mul(ret, mul(ret, v)); }
34  }
35  int n, m, dp[2010][2010], fac[2010], inv_fac[2010], inv[2010];
36  void input(){
37      int i, j;
38      sii(n, m);
39  }
40  void solve(){
41      int i, j, x, k, y;
42      for(i = 1; i <= n; i++){
43          for(j = i, dp[0][i + 1] = 1; j >= 1; j--){
44              for(k = 0, x = 1; k * j <= i; k++){
45                  y = mul(x, inv_fac[k]);
46                  dp[i][j] = add(dp[i][j], mul(y, dp[i - k * j][j + 1]));
47                  x = mul(x, mul(inv[j], j % 2 == 0 ? mul(m, m) : m));
48              }
49          }
50      }
51      pi(mul(dp[n][1], fac[n])); nl;
52  }
53  void pre_process(){
54      int i, j;
55      for(i = 1, fac[i - 1] = 1; i <= 2005; i++){
56          fac[i] = mul(fac[i - 1], i);
57      }
58      for(i = 2004, inv_fac[i + 1] = big_mod(fac[i + 1], MOD - 2); i >= 0;
    i--){
59          inv_fac[i] = mul(inv_fac[i + 1], i + 1);
60      }
61      for(i = 2005; i >= 1; i--){
62          inv[i] = mul(inv_fac[i], fac[i - 1]);
63      }
64  }
65  int main(){
66  //    freopen("input.txt","r",stdin);
```

```
67  //     freopen("output.txt", "w", stdout);
68      pre_process();
69      input();
70      solve();
71  }
72
73
```