

라이블리 프로그래밍 4 (연습)

소스 코드

최백준 [choi@startlink.io](mailto:choi@startlink.io)

---

C++14

```
1 #include <stdio>
2 #include <string>
3 int a[16][16];
4 char str[20];
5 int d[1<<16];
6 int main() {
7     int n;
8     scanf("%d",&n);
9     for (int i=0; i<n; i++) {
10         for (int j=0; j<n; j++) {
11             scanf("%d",&a[i][j]);
12         }
13     }
14     scanf("%s",str);
15     int start = 0;
16     memset(d,-1,sizeof(d));
17     for (int i=n-1; i>=0; i--) {
18         start = start * 2;
19         if (str[i] == 'Y') {
20             start += 1;
21         }
22     }
23     int p;
24     scanf("%d",&p);
25     d[start] = 0;
26     for (int i=0; i<(1<<n); i++) {
27         if (d[i] == -1) continue;
28         for (int j=0; j<n; j++) {
29             if (i&(1<<j)) { // j가 켜져 있음
30                 for (int k=0; k<n; k++) {
31                     if ((i&(1<<k))==0) { // k가 꺼져있음
32                         if (d[i|(1<<k)] == -1 || d[i|(1<<k)] > d[i] + a[j][k]) {
33                             d[i|(1<<k)] = d[i]+a[j][k];
34                         }
35                     }
36                 }
37             }
38         }
39     }
40     int ans = -1;
41     for (int i=0; i<(1<<n); i++) {
42         if (d[i] == -1) continue;
43         int cnt = 0;
44         for (int j=0; j<n; j++) {
45             if (i&(1<<j)) {
46                 cnt += 1;
47             }
48         }
49         if (cnt >= p) {
50             if (ans == -1 || ans > d[i]) {
51                 ans = d[i];
52             }
53         }
54     }
55     printf("%d\n",ans);
56     return 0;
57 }
58 }
```

YN

줄기 상태

=>

2<sup>N</sup>

7가 낮은 상태

N

N

O(2<sup>N</sup> · N<sup>2</sup>)

결과

메모리

시간

코드 길이

맞았습니다!!

1372 KB

16 ms

1359 B

C++

```
1 #include <iostream>
2 #include <iomanip>
3 using namespace std;
4 double a[20][20];
5 bool c[1<<20];
6 double d[1<<20];
7 int n;
8 double go(int i, int s) {
9     if (i == n) {
10         return 1.0;
11     }
12     if (c[s]) {
13         return d[s];
14     }
15     c[s] = true;
16     double &ans = d[s];
17     for (int k=0; k<n; k++) {
18         if ((s&(1<<k)) == 0) {
19             double temp = a[i][k]*go(i+1, s|(1<<k));
20             if (ans < temp) {
21                 ans = temp;
22             }
23         }
24     }
25     return ans;
26 }
27 int main() {
28     cin >> n;
29     for (int i=0; i<n; i++) {
30         for (int j=0; j<n; j++) {
31             cin >> a[i][j];
32             a[i][j] /= 100.0;
33         }
34     }
35     cout << fixed << setprecision(10) << go(0, 0)*100.0 << '\n';
36     return 0;
37 }
38
```

0~(i-1) 까지 환?  
T가 어떤 값?  
                    

OK  
T → K

$O(2^N \cdot N)$

결과

메모리

시간

코드 길이

맞았습니다!!

11208 KB

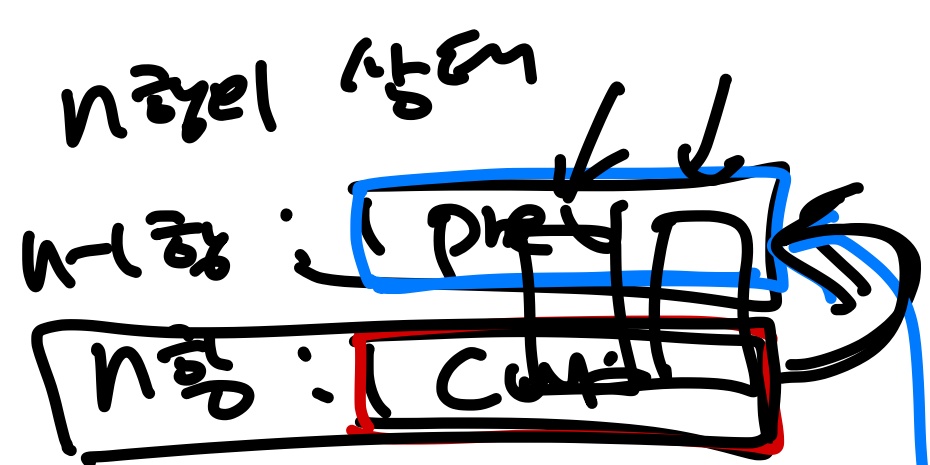
168 ms

737 B

C++14

```
1 #include <iostream>
2 #include <cstring>
3 #include <vector>
4 #include <algorithm>
5 using namespace std;
6 int a[1001][3];
7 long long d[1001][1001][1<<3];
8 const long long inf = 10000000000000000LL;
9 int states[11][3] = {
10     {0, 0, 0},
11     {1, 1, 1},
12     {2, 2, 1},
13     {4, 4, 1},
14     {0, 3, 1},
15     {0, 6, 1},
16     {1, 7, 2},
17     {4, 7, 2},
18     {3, 3, 2},
19     {6, 6, 2},
20     {5, 5, 2}
21 };
22 long long calc(int n, int cur) {
23     long long ans = 0;
24     if (n < 0 && cur != 0) return -inf;
25     if (cur & 1) ans += a[n][0];
26     if (cur & 2) ans += a[n][1];
27     if (cur & 4) ans += a[n][2];
28     return ans;
29 }
30 long long go(int n, int k, int state) {
31     if (n < 0) {
32         if (k == 0) return 0;
33         else return -inf;
34     }
35     if (k == 0) return 0;
36     if (k < 0) return -inf;
37     long long &ans = d[n][k][state];
38     if (ans != -inf) return ans;
39     for (int i=0; i<11; i++) {
40         int prev = states[i][0];
41         int cur = states[i][1];
42         int cost = states[i][2];
43         if (cur & state) continue;
44         ans = max(ans, calc(n, cur) + calc(n-1, prev) + go(n-1, k-cost, prev));
45     }
46     return ans;
47 }
48 int main() {
49     int n, m, k;
50     cin >> n >> k;
51     m = 3;
52     for (int i=0; i<n; i++) {
53         for (int j=0; j<m; j++) {
54             cin >> a[i][j];
55         }
56     }
57     fill(&d[0][0][0], &d[1001][0][0], -inf);
58     cout << go(n-1, k, 0) << '\n';
59     return 0;
60 }
61
```

행 → 도미노 → 상태  
(이전 행의 상태, 다음 행의 상태, 새로 놓은 도미노의 개수)



C++14

```
1 #include <iostream>
2 #include <vector>
3 #include <string>
4 using namespace std;
5 long long d[1<<15][100];
6 long long gcd(long long x, long long y) {
7     if (y == 0) return x;
8     else return gcd(y, x%y);
9 }
10 int main() {
11     int n;
12     cin >> n;
13     vector<string> num(n);
14     vector<int> a(n);  $\rightarrow num[i] \% k$ 
15     vector<int> len(n);  $\rightarrow num[i]$ 의 길이
16     for (int i=0; i<n; i++) {
17         cin >> num[i];
18         len[i] = num[i].size();
19     }
20     int k;
21     cin >> k;
22     for (int i=0; i<n; i++) {
23         for (int j=0; j<len[i]; j++) {
24             a[i] = a[i] * 10 + (num[i][j] - '0');
25             a[i] %= k;
26         }
27     }
28     vector<int> ten(51);
29     ten[0] = 1;
30     for (int i=1; i<=50; i++) {
31         ten[i] = ten[i-1] * 10;
32         ten[i] %= k;
33     }
34     d[0][0] = 1;
35     for (int i=0; i<(1<<n); i++) {
36         for (int j=0; j<k; j++) {
37             for (int l=0; l<n; l++) {
38                 if ((i&(1<<l)) == 0) {
39                     int next = j * ten[len[l]];
40                     next %= k;
41                     next += a[l];
42                     next %= k;
43                     d[i|(1<<l)][next] += d[i][j];
44                 }
45             }
46         }
47     }
48     long long t1 = d[(1<<n)-1][0];
49     long long t2 = 1;  $\rightarrow n!$ 
50     for (int i=2; i<=n; i++) {
51         t2 *= (long long)i;
52     }
53     long long g = gcd(t1, t2);
54     t1 /= g;
55     t2 /= g;
56     cout << t1 << '/' << t2 << '\n';
57     return 0;
58 }
```

결과	메모리	시간	코드 길이
맞았습니다!!	27592 KB	252 ms	1353 B

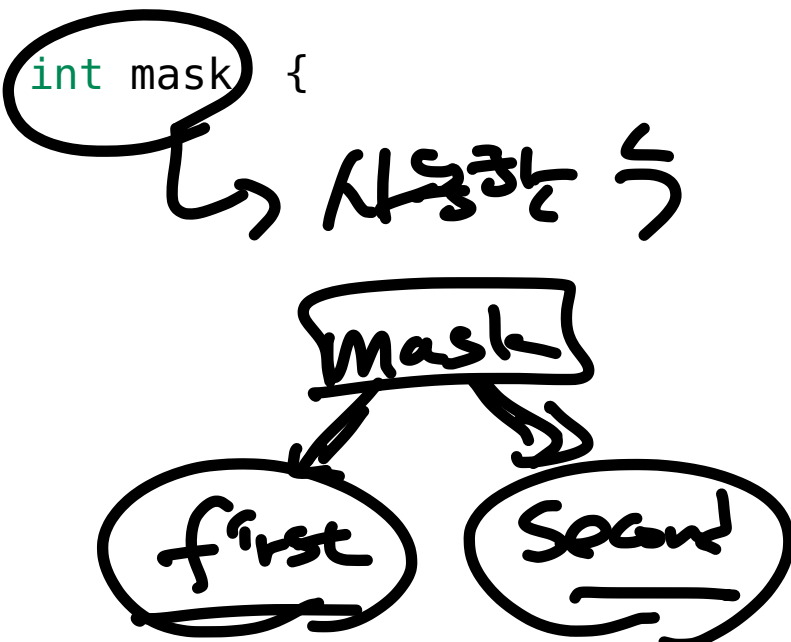


C++14

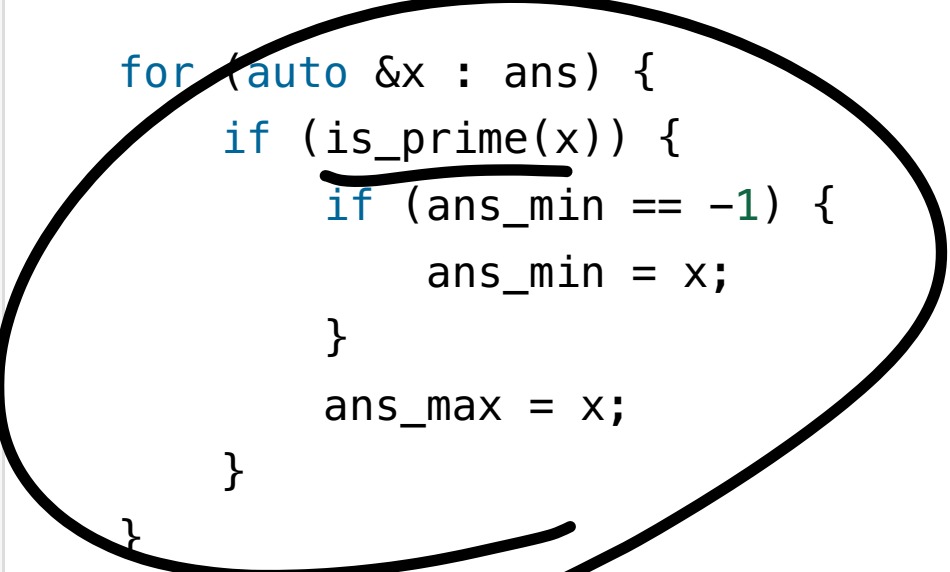
```
1 #include <iostream>
2 #include <vector>
3 #include <set>
4 using namespace std;
5 set<int> d[1<<6];
6 bool is_prime(int x) {
7     if (x < 2) {
8         return false;
9     }
10    for (int i=2; i*i <= x; i++) {
11        if (x % i == 0) {
12            return false;
13        }
14    }
15    return true;
16 }
17 set<int> solve(vector<int> &a, int mask) {
18     set<int> &ans = d[mask];
19     if (ans.size() > 0) {
20         return ans;
21     }
22     if (mask == 0) {
23         return ans;
24     }
25     int n = a.size();
26     for (int i=0; i<(1<<n); i++) {
27         int first = i & mask;
28         int second = ~first & mask;
29         if (first == 0 || second == 0) {
30             continue;
31         }
32         solve(a, first);
33         solve(a, second);
34
35         for (int x: d[first]) {
36             for (int y: d[second]) {
37                 if (x != 0) {
38                     ans.insert(y/x);
39                 }
40                 if (y != 0) {
41                     ans.insert(x/y);
42                 }
43                 ans.insert(x+y);
44                 ans.insert(x-y);
45                 ans.insert(y-x);
46                 ans.insert(x*y);
47             }
48         }
49     }
50     return ans;
51 }
52 int main() {
53     int n;
54     cin >> n;
55     vector<int> a(n);
56     for (int i=0; i<n; i++) {
57         cin >> a[i];
58         d[1<<i].insert(a[i]);
59     }
60
61     auto ans = solve(a, (1<<n)-1);
62
63     int ans_min=-1;
64     int ans_max=-1;
65
66     for (auto &x : ans) {
67         if (is_prime(x)) {
68             if (ans_min == -1) {
69                 ans_min = x;
70             }
71             ans_max = x;
72         }
73     }
74
75     if (ans_min == -1) {
76         cout << -1 << '\n';
77     } else {
78         cout << ans_min << '\n';
79         cout << ans_max << '\n';
80     }
81     return 0;
82 }
83
```

다진것: 사용가능한 수의  
상태가 어떻게  
만들수 있는지  
모든 수

입력



first & second == 0  
first | second == mask



C++14

```
1 #include <iostream>
2 #include <algorithm>
3 #include <string>
4 #include <vector>
5 #include <cstring>
6 using namespace std;
7 pair<int,int> d[2501][51][1<<4];
8 string a[4];
9 string str;
10 pair<int,int> go(int index, int covered, int mask) {
11     if (mask == 0) {
12         return make_pair(0, 0);
13     }
14     auto &ans = d[index][covered][mask];
15     if (ans.first != -1) {
16         return ans;
17     }
18     ans = make_pair(1000000, -1000000);
19     if (index == str.size()) {
20         return ans;
21     }
22     ans = go(index+1, max(0, covered-1), mask);
23     for (int i=0; i<4; i++) {
24         if (mask & (1<<i)) {
25             if (str.substr(index, a[i].size()) == a[i]) {
26                 int nextc = 0;
27                 if (a[i].size() > covered) {
28                     nextc = (int)a[i].size() - covered;
29                 }
30                 auto temp = go(index, max(covered, (int)a[i].size()), mask - (1<<i));
31                 ans = make_pair(min(temp.first+nextc, ans.first), max(temp.second+nextc, ans.second));
32             }
33         }
34     }
35     return ans;
36 }
37 int main() {
38     cin >> str;
39     for (int i=0; i<4; i++) {
40         cin >> a[i];
41     }
42     sort(a,a+4);
43     memset(d,-1,sizeof(d));
44     auto ans = go(0, 0, 15);
45     cout << ans.first << ' ' << ans.second << '\n';
46     return 0;
47 }
48
```

3중, 2중

3중의 경우

사용

사용 X

start [index]  
[i]

결과

메모리

시간

코드 길이

맞았습니다!!

18208 KB

244 ms

1258 B

C++14

```
1 #include <iostream>
2 #include <cstring>
3 using namespace std;
4 char a[20][20];
5 int d[14*14][1<<14];
6 int n,m;
7 int score[6][6] = {
8     {10,8,7,5,0,1},
9     {8,6,4,3,0,1},
10    {7,4,3,2,0,1},
11    {5,3,2,2,0,1},
12    {0,0,0,0,0,0},
13    {1,1,1,1,0,0}
14 };
15 int calc(char x, char y) {
16     return score[x-'A'][y-'A'];
17 }
18 int go(int num, int state) {
19     if (num >= n*m) {
20         return 0;
21     }
22     if (d[num][state] >= 0) {
23         return d[num][state];
24     }
25     int &ans = d[num][state];
26     ans = 0;
27     int x = num/m;
28     int y = num%m;
29     ans = go(num+1, (state >> 1));
30     if ((state & 1) == 0) {
31         if (x != n-1) {
32             int temp = go(num+1, (state >> 1) | (1 << (m-1)));
33             temp += calc(a[x][y], a[x+1][y]);
34             ans = max(ans, temp);
35         }
36         if (y != m-1 && (state & 2) == 0) {
37             int temp = go(num+2, (state >> 2));
38             temp += calc(a[x][y], a[x][y+1]);
39             ans = max(ans, temp);
40         }
41     }
42     return ans;
43 }
44 int main() {
45     cin >> n >> m;
46     for (int i=0; i<n; i++) {
47         cin >> a[i];
48     }
49     memset(d,-1,sizeof(d));
50     cout << go(0, 0) << '\n';
51     return 0;
52 }
```

$N \times M \times 2^M$

73 (state & 1)

결과

메모리

시간

코드 길이

맞았습니다!!

14532 KB

44 ms

1134 B

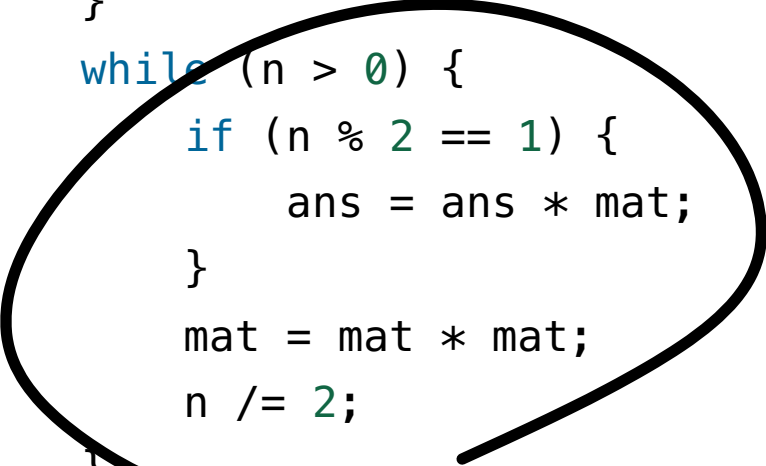
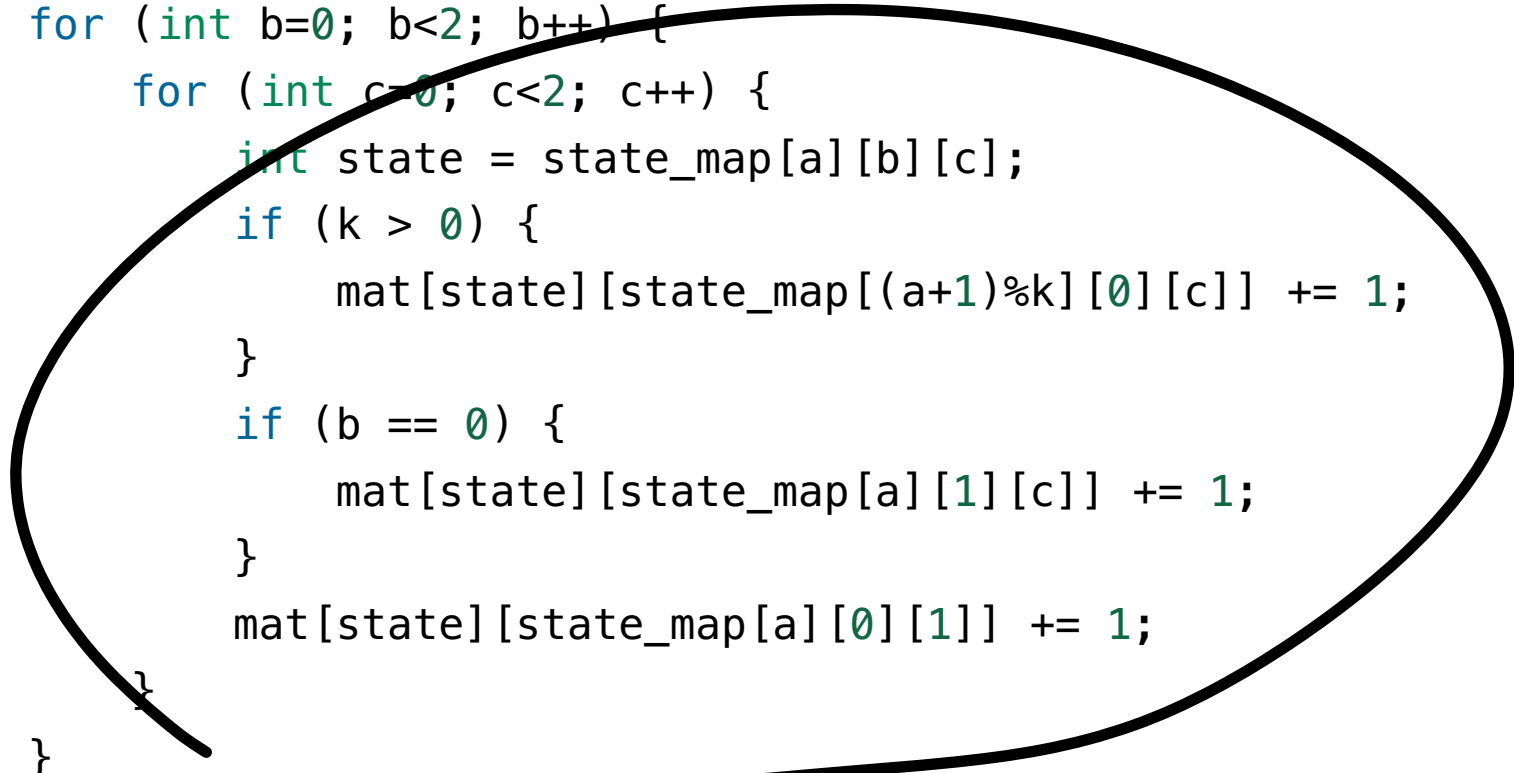
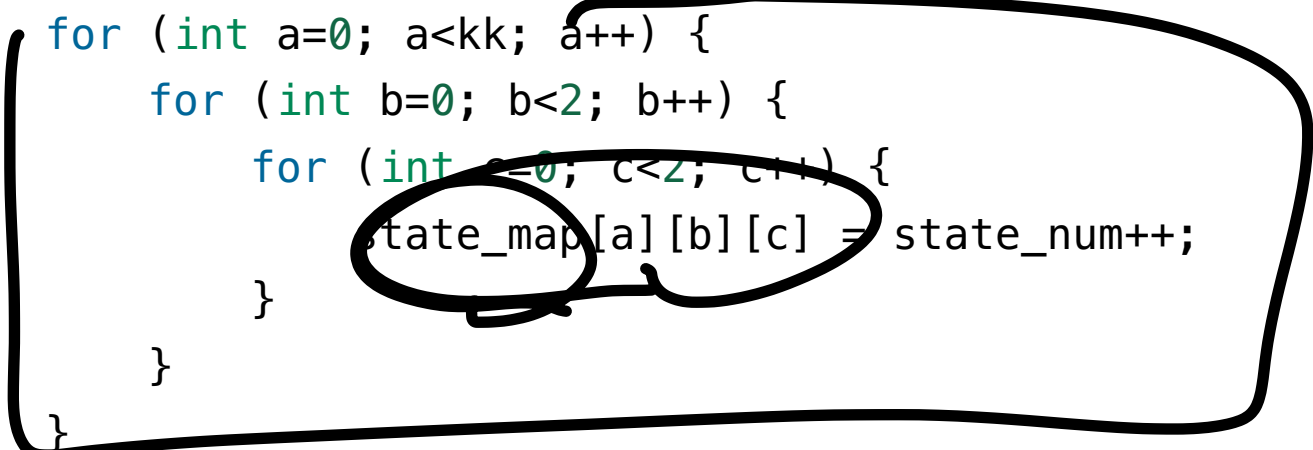
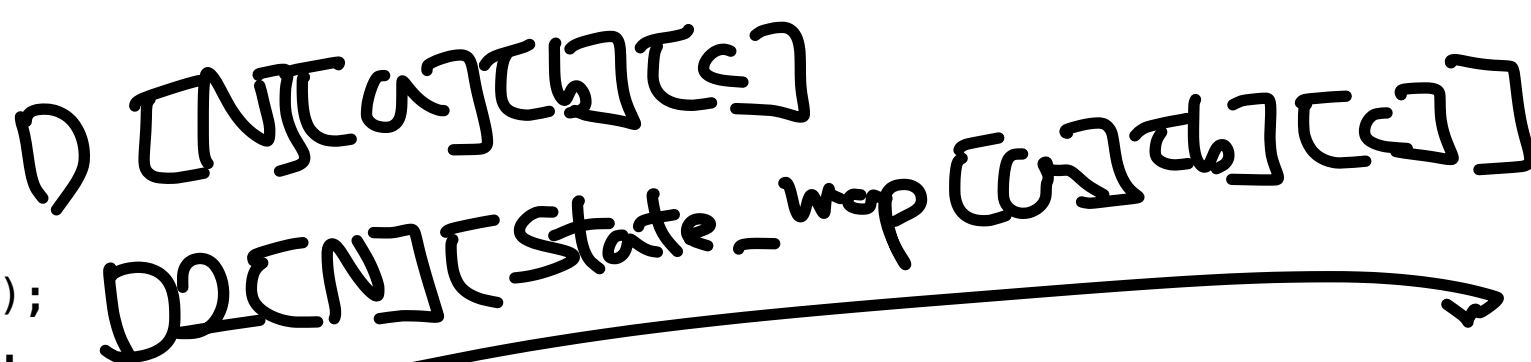
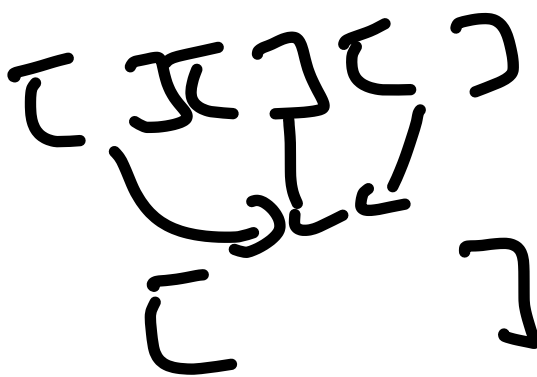


C++14

```
1 #include <iostream>
2 #include <vector>
3 using namespace std;
4 const long long mod = 1000000007;
5 using matrix = vector<vector<long long>>;
6 matrix operator*(const matrix &a, const matrix &b) {
7     int n = a.size();
8     matrix c(n, vector<long long>(n));
9     for (int i=0; i<n; i++) {
10         for (int j=0; j<n; j++) {
11             for (int k=0; k<n; k++) {
12                 c[i][j] += a[i][k] * b[k][j];
13                 c[i][j] %= mod;
14             }
15         }
16     }
17     return c;
18 }
19 int main() {
20     int n, m;
21     cin >> n >> m;
22     matrix ans(m, vector<long long>(m));
23     matrix even(m, vector<long long>(m));
24     matrix odd(m, vector<long long>(m));
25     matrix a(m, vector<long long>(m));
26     for (int i=0; i<m; i++) {
27         ans[i][i] = 1;
28         odd[i][i] = 1;
29         if (i-1 >= 0) {
30             odd[i][i-1] = 1;
31             even[i][i-1] = 1;
32         }
33         if (i+1 < m) {
34             odd[i][i+1] = 1;
35             even[i][i+1] = 1;
36         }
37     }
38     n -= 1;
39     a = even * odd;
40     int p = n/2;
41     while (p > 0) {
42         if (p % 2 == 1) {
43             ans = ans * a;
44         }
45         a = a * a;
46         p /= 2;
47     }
48     if (n % 2 == 1) {
49         ans = ans * even;
50     }
51     long long sum = 0;
52     for (int i=0; i<m; i++) {
53         for (int j=0; j<m; j++) {
54             sum += ans[i][j];
55             sum %= mod;
56         }
57     }
58     cout << sum << '\n';
59     return 0;
60 }
61
```

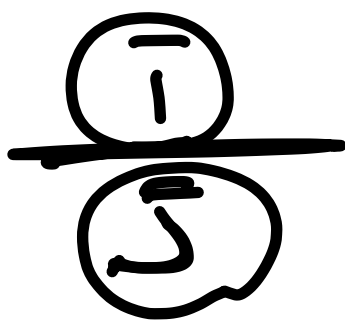
C++14

```
1 #include <iostream>
2 #include <vector>
3 using namespace std;
4 const long long mod = 1000000007;
5 using matrix = vector<vector<long long>>;
6 matrix operator * (const matrix &a, const matrix &b) {
7     int n = a.size();
8     matrix c(n, vector<long long>(n));
9     for (int i=0; i<n; i++) {
10         for (int j=0; j<n; j++) {
11             for (int k=0; k<n; k++) {
12                 c[i][j] += a[i][k] * b[k][j];
13                 c[i][j] %= mod;
14             }
15         }
16     }
17     return c;
18 }
19 int state_map[10][2][2];
20 int main() {
21     long long n;
22     int k;
23     cin >> n >> k;
24     int kk = max(k, 1);
25     int state_num = 0;
26     for (int a=0; a<kk; a++) {
27         for (int b=0; b<2; b++) {
28             for (int c=0; c<2; c++) {
29                 state_map[a][b][c] = state_num++;
30             }
31         }
32     }
33     int size = kk * 2 * 2;
34     matrix ans(size, vector<long long>(size));
35     matrix mat(size, vector<long long>(size));
36     for (int i=0; i<size; i++) {
37         ans[i][i] = 1;
38     }
39     for (int a=0; a<kk; a++) {
40         for (int b=0; b<2; b++) {
41             for (int c=0; c<2; c++) {
42                 int state = state_map[a][b][c];
43                 if (k > 0) {
44                     mat[state][state_map[(a+1)%kk][0][c]] += 1;
45                 }
46                 if (b == 0) {
47                     mat[state][state_map[a][1][c]] += 1;
48                 }
49                 mat[state][state_map[a][0][1]] += 1;
50             }
51         }
52     }
53     while (n > 0) {
54         if (n % 2 == 1) {
55             ans = ans * mat;
56         }
57         mat = mat * mat;
58         n /= 2;
59     }
60     long long sum = ans[0][state_map[0][0][1]] + ans[0][state_map[0][1][1]];
61     sum %= mod;
62     cout << sum << '\n';
63     return 0;
64 }
```



C++14

```
1 #include <iostream>
2 #include <vector>
3 using namespace std;
4 const long long mod = 1000000007;
5 using matrix = vector<vector<long long>>;
6 matrix operator*(const matrix &a, const matrix &b) {
7     int n = a.size();
8     matrix c(n, vector<long long>(n));
9     for (int i=0; i<n; i++) {
10         for (int j=0; j<n; j++) {
11             for (int k=0; k<n; k++) {
12                 c[i][j] += a[i][k] * b[k][j];
13                 c[i][j] %= mod;
14             }
15         }
16     }
17     return c;
18 }
19 int isset(int state, int index) {
20     if (state & (1 << index)) {
21         return 1;
22     } else {
23         return 0;
24     }
25 }
26 bool ok(int pstate, int state, int m) {
27     for (int i=0; i<m-1; i++) {
28         int color11 = isset(pstate, i);
29         int color12 = isset(pstate, i+1);
30         int color21 = isset(state, i);
31         int color22 = isset(state, i+1);
32         if (color11 == color12 && color12 == color21 && color21 == color22) {
33             return false;
34         }
35     }
36     return true;
37 }
38 int main() {
39     long long n;
40     int m;
41     cin >> n >> m;
42     int size = (1<<m);
43     matrix ans(size, vector<long long>(size));
44     matrix a(size, vector<long long>(size));
45     for (int i=0; i<size; i++) {
46         ans[i][i] = 1;
47         for (int j=0; j<size; j++) {
48             if (ok(i, j, m)) {
49                 a[i][j] = 1;
50             }
51         }
52     }
53     n -= 1;
54     while (n > 0) {
55         if (n % 2 == 1) {
56             ans = ans * a;
57         }
58         a = a * a;
59         n /= 2;
60     }
61     long long sum = 0;
62     for (int i=0; i<size; i++) {
63         for (int j=0; j<size; j++) {
64             sum += ans[i][j];
65             sum %= mod;
66         }
67     }
68     cout << sum << '\n';
69     return 0;
70 }
71
```



C++14

```
1 #include <iostream>
2 #include <vector>
3 #include <algorithm>
4 using namespace std;
5 typedef vector<vector<long long>> matrix;
6 const long long mod = 1000000009LL;
7 matrix operator * (const matrix &a, const matrix &b) {
8     int n = a.size();
9     matrix c(n, vector<long long>(n));
10    for (int i=0; i<n; i++) {
11        for (int j=0; j<n; j++) {
12            for (int k=0; k<n; k++) {
13                c[i][j] += a[i][k] * b[k][j];
14                c[i][j] %= mod;
15            }
16        }
17    }
18    return c;
19 }
20 int dx[] = {-2,-1,1,2};
21 int dy[] = {1,0,0,1};
22 matrix make(int n, int m) {
23     auto on = [&](int state, int col, int row) {
24         return state & (1 << (col*n+row));
25     };
26     auto possible = [&](int state, int next) {
27         for (int i=0; i<n; i++) {
28             if (!on(next, 0, i)) continue;
29             for (int k=0; k<4; k++) {
30                 if (i+dx[k] < 0 || i+dx[k] >= n) continue;
31                 if (on(state, dy[k], i+dx[k])) return false;
32             }
33         }
34         return true;
35     };
36     int states = (1 << 2*n);
37     matrix adj = vector<vector<long long>>(states, vector<long long>(states));
38     for (int i=0; i<states; i++) {
39         for (int j=0; j<(1<<n); j++) {
40             if (possible(i, j)) {
41                 int k = (i >> n) | (j << n);
42                 adj[i][k] = 1;
43             }
44         }
45     }
46     return adj;
47 }
48 matrix pow(matrix a, int m) {
49     int n = a.size();
50     matrix ans = vector<vector<long long>>(n, vector<long long>(n));
51     for (int i=0; i<n; i++) {
52         ans[i][i] = 1;
53     }
54     while (m > 0) {
55         if (m % 2 == 1) {
56             ans = ans * a;
57         }
58         a = a * a;
59         m /= 2;
60     }
61     return ans;
62 }
63 int main() {
64     int t;
65     cin >> t;
66     while (t--) {
67         int n, m;
68         cin >> n >> m;
69         matrix adj = make(n, m);
70         matrix ans = pow(adj, m);
71         long long sum = 0;
72         for (int i=0; i<(1<<(2*n)); i++) {
73             sum = (sum + ans[0][i]) % mod;
74         }
75         cout << sum << '\n';
76     }
77     return 0;
78 }
79
```

결과

메모리

시간

코드 길이

맞았습니다!!

3980 KB

6052 ms

2007 B



끝

---

# 코드 플러스

<https://code.plus>

- 슬라이드에 포함된 소스 코드를 보려면 "정보 수정 > 백준 온라인 저지 연동"을 통해 연동한 다음, "백준 온라인 저지"에 로그인해야 합니다.
- 강의 내용에 대한 질문은 코드 플러스의 "질문 게시판"에서 할 수 있습니다.
- 문제와 소스 코드는 슬라이드에 첨부된 링크를 통해서 볼 수 있으며, "백준 온라인 저지"에서 서비스됩니다.
- 슬라이드와 동영상 강의는 코드 플러스 사이트를 통해서만 볼 수 있으며, 동영상 강의의 녹화와 다운로드, 배포와 유통은 저작권법에 의해서 금지되어 있습니다.
- 다른 경로로 이 슬라이드나 동영상 강의를 본 경우에는 [codeplus@startlink.io](mailto:codeplus@startlink.io) 로 이메일 보내주세요.
- 강의 내용, 동영상 강의, 슬라이드, 첨부되어 있는 소스 코드의 저작권은 스타트링크와 최백준에게 있습니다.