

다이나믹 프로그래밍 문제

최백준 choi@startlink.io

1로 만들기

<https://www.acmicpc.net/problem/1463>

- 세준이는 어떤 정수 N 에 다음과 같은 연산중 하나를 할 수 있다.
 1. N 이 3으로 나누어 떨어지면, 3으로 나눈다.
 2. N 이 2로 나누어 떨어지면, 2로 나눈다.
 3. 1을 뺀다.
- 세준이는 어떤 정수 N 에 위와 같은 연산을 선택해서 1을 만드려고 한다. 연산을 사용하는 횟수의 최소값을 출력하시오.

1로 만들기

<https://www.acmicpc.net/problem/1463>

- $D[i] = i$ 를 1로 만드는데 필요한 최소 연산 횟수
- i 에게 가능한 경우를 생각해보자
 1. i 가 3으로 나누어 떨어졌을 때, 3으로 나누는 경우
 2. i 가 2로 나누어 떨어졌을 때, 2로 나누는 경우
 3. i 에서 1을 빼는 경우

1로 만들기

<https://www.acmicpc.net/problem/1463>

- $D[i]$ = i 를 1로 만드는데 필요한 최소 연산 횟수
- i 에게 가능한 경우를 생각해보자
 1. i 가 3으로 나누어 떨어졌을 때, 3으로 나누는 경우
 - $D[i/3] + 1$
 2. i 가 2로 나누어 떨어졌을 때, 2로 나누는 경우
 - $D[i/2] + 1$
 3. i 에서 1을 빼는 경우
 - $D[i-1] + 1$

1로 만들기

<https://www.acmicpc.net/problem/1463>

- $D[i]$ = i 를 1로 만드는데 필요한 최소 연산 횟수
- i 에게 가능한 경우를 생각해보자
 1. i 가 3으로 나누어 떨어졌을 때, 3으로 나누는 경우
 - $D[i/3] + 1$
 2. i 가 2로 나누어 떨어졌을 때, 2로 나누는 경우
 - $D[i/2] + 1$
 3. i 에서 1을 빼는 경우
 - $D[i-1] + 1$
- 세 값중의 최소값이 들어가게 된다.

1로 만들기

<https://www.acmicpc.net/problem/1463>

```
int go(int n) {  
    if (n == 1) return 0;  
    if (d[n] > 0) return d[n];  
    d[n] = go(n-1) + 1;  
    if (n%2 == 0) {  
        int temp = go(n/2) + 1;  
        if (d[n] > temp) d[n] = temp;  
    }  
    if (n%3 == 0) {  
        int temp = go(n/3) + 1;  
        if (d[n] > temp) d[n] = temp;  
    }  
    return d[n];  
}
```

1로 만들기

<https://www.acmicpc.net/problem/1463>

```
d[1] = 0;
for (int i=2; i<=n; i++) {
    d[i] = d[i-1] + 1;
    if (i%2 == 0 && d[i] > d[i/2] + 1) {
        d[i] = d[i/2] + 1;
    }
    if (i%3 == 0 && d[i] > d[i/3] + 1) {
        d[i] = d[i/3] + 1;
    }
}
```

1로 만들기

<https://www.acmicpc.net/problem/1463>

- Top-Down 방식
- C: <https://gist.github.com/Baekjoon/a53dc4861bd9d081682c>
- C++: <https://gist.github.com/Baekjoon/63b659f985beb8f64ca7>
- Java: <https://gist.github.com/Baekjoon/7b675fe68d3c2abfef40>

1로 만들기

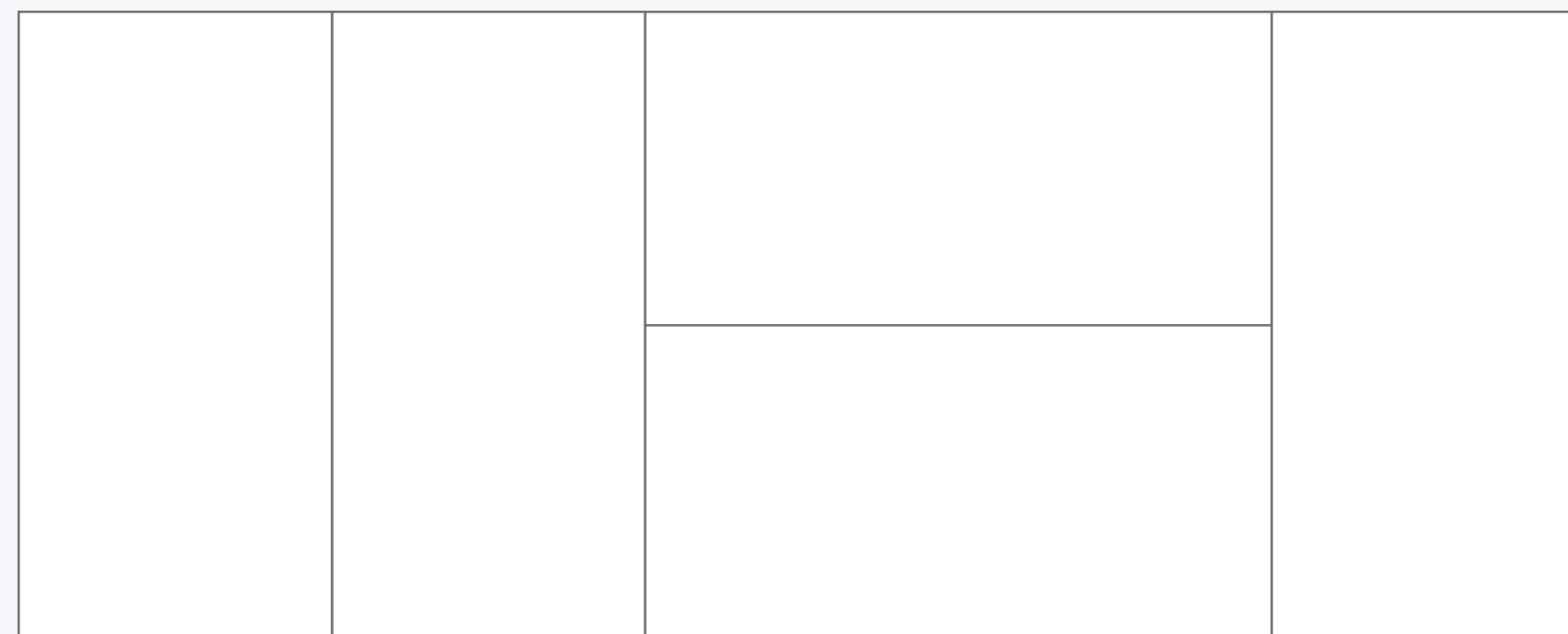
<https://www.acmicpc.net/problem/1463>

- Bottom-up 방식
- C: <https://gist.github.com/Baekjoon/30f4bb39cdc66f7f16c1>
- C++: <https://gist.github.com/Baekjoon/31e553ab3b371fe06384>
- Java: <https://gist.github.com/Baekjoon/0813d3bc5db11b9bb72d>

2×n 타일링

<https://www.acmicpc.net/problem/11726>

- 2×n 직사각형을 1×2, 2×1타일로 채우는 방법의 수
- 아래 그림은 2×5를 채우는 방법의 수
- $D[i] = 2 \times i$ 직사각형을 채우는 방법의 수

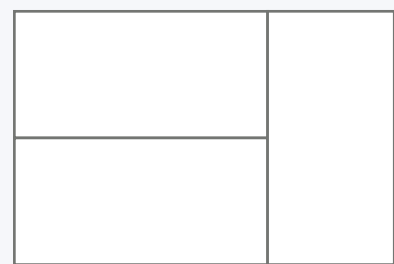
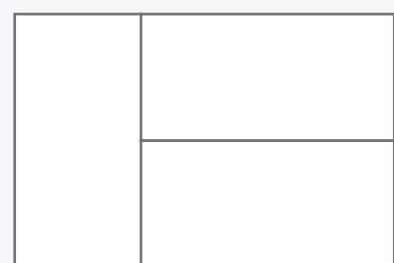
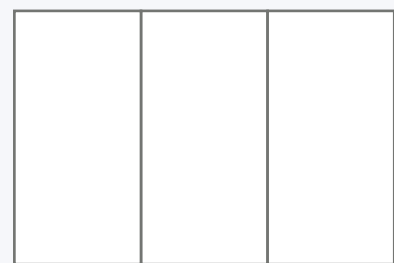


2×n 타일링

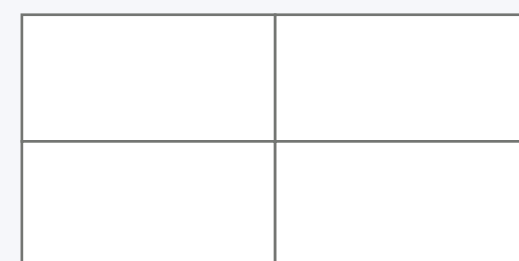
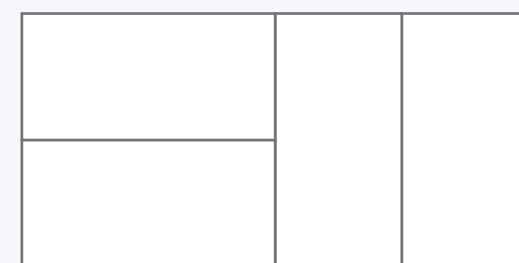
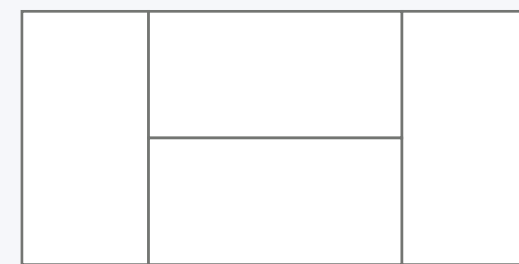
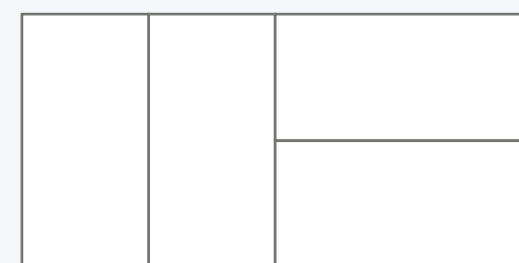
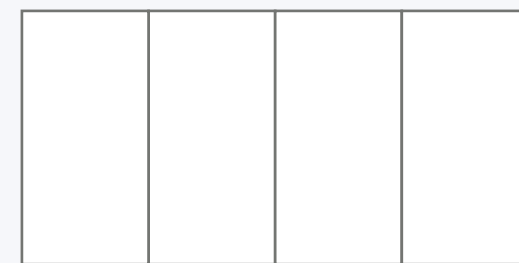
<https://www.acmicpc.net/problem/11726>

11

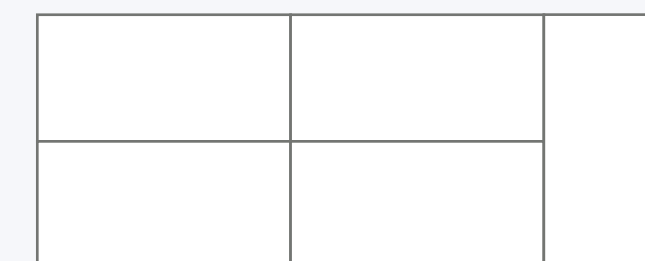
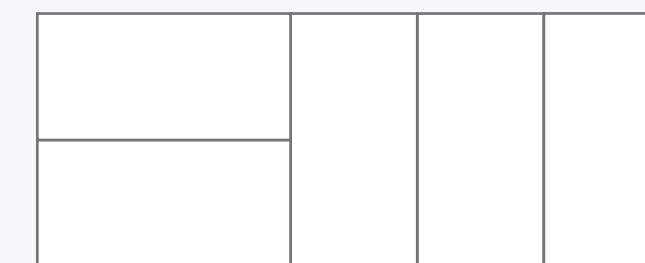
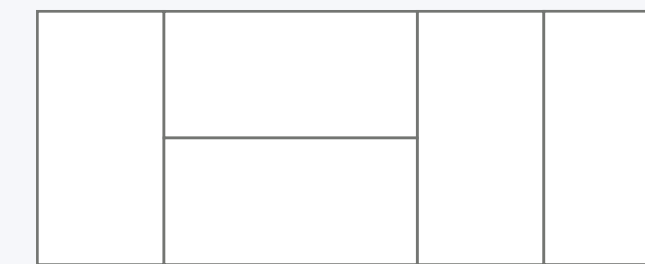
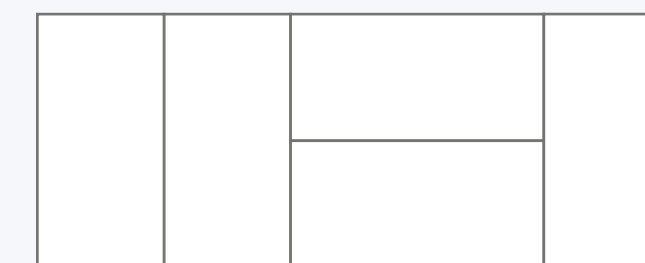
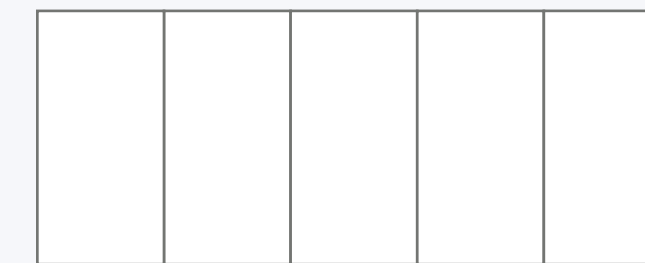
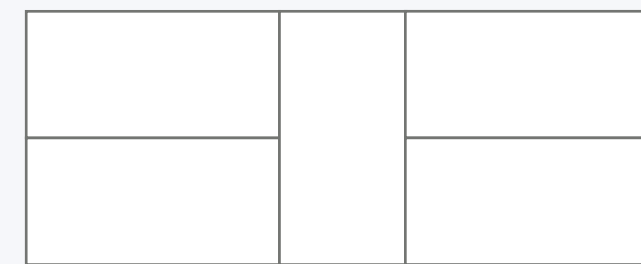
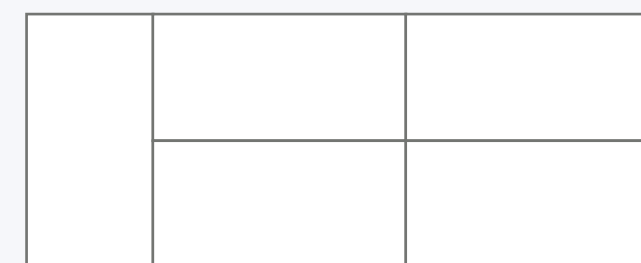
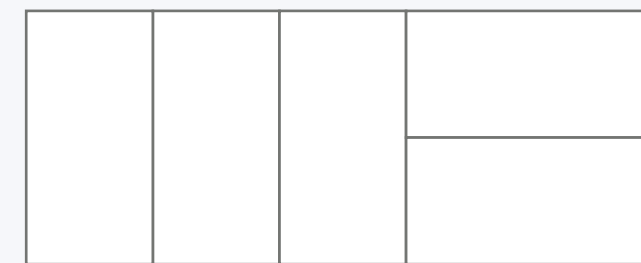
2×3



2×4



2×5



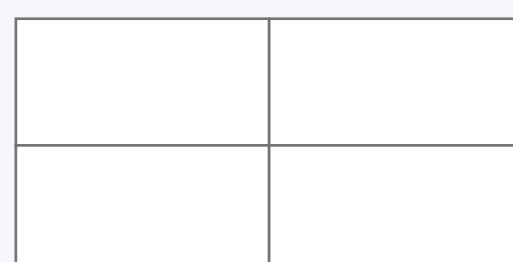
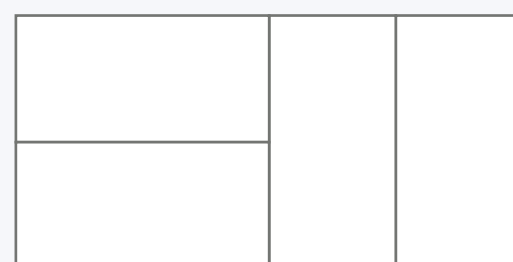
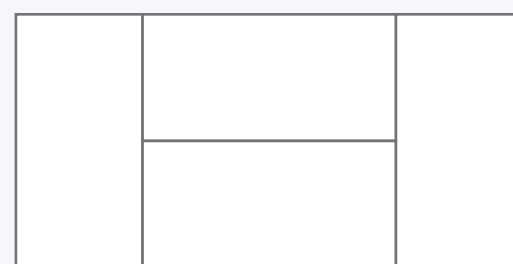
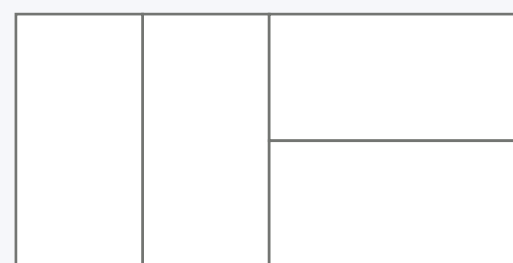
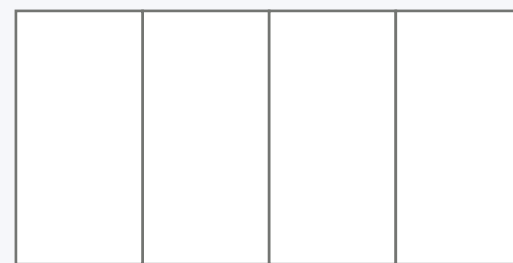
2×n 타일링

<https://www.acmicpc.net/problem/11726>

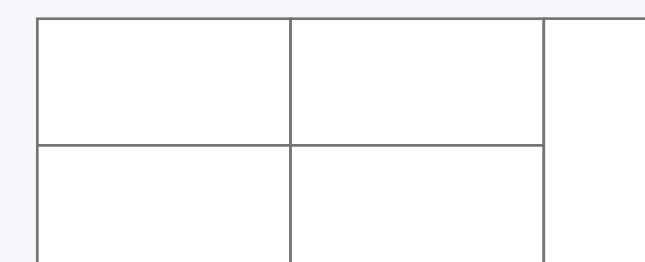
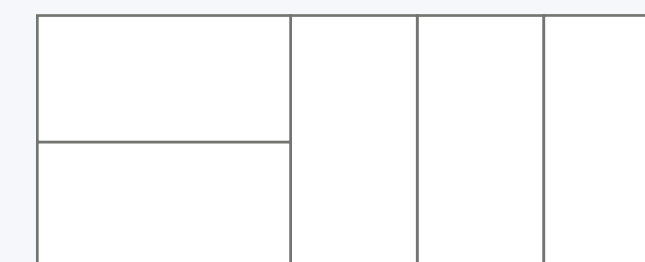
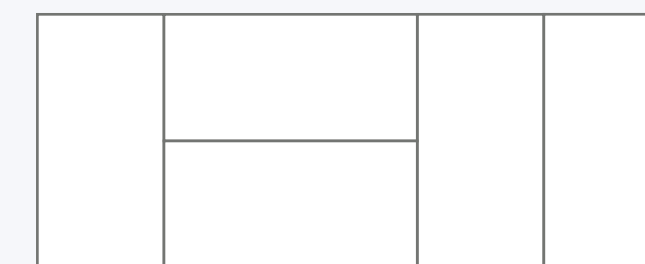
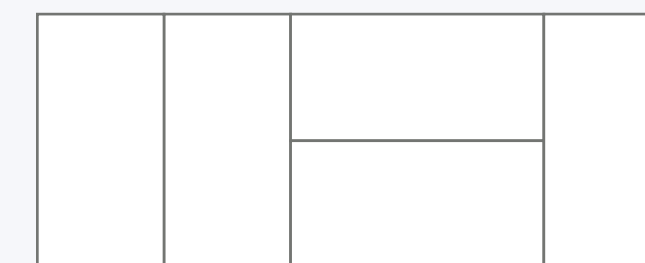
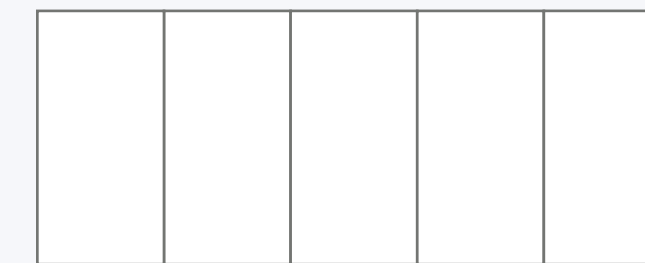
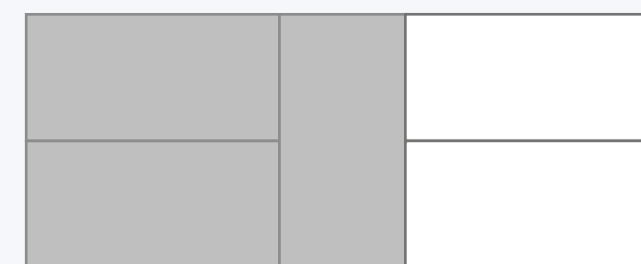
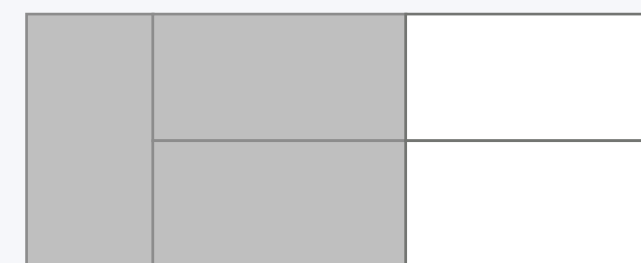
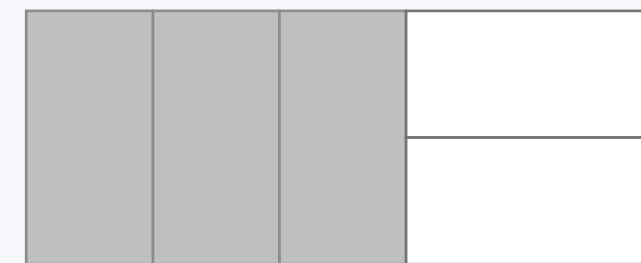
2×3



2×4



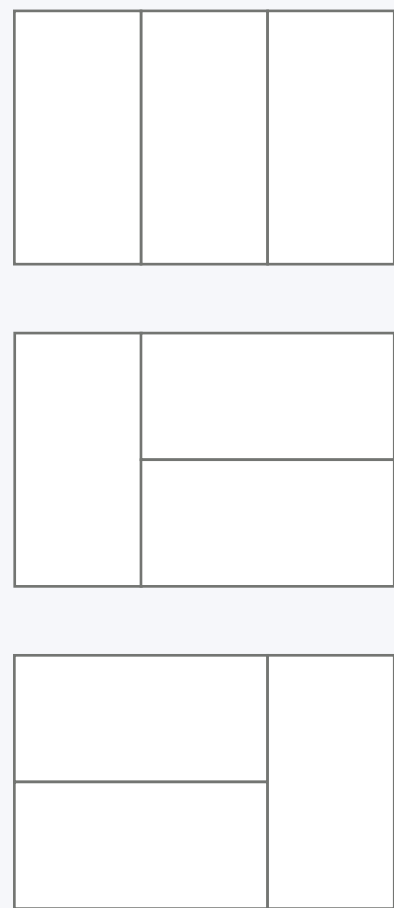
2×5



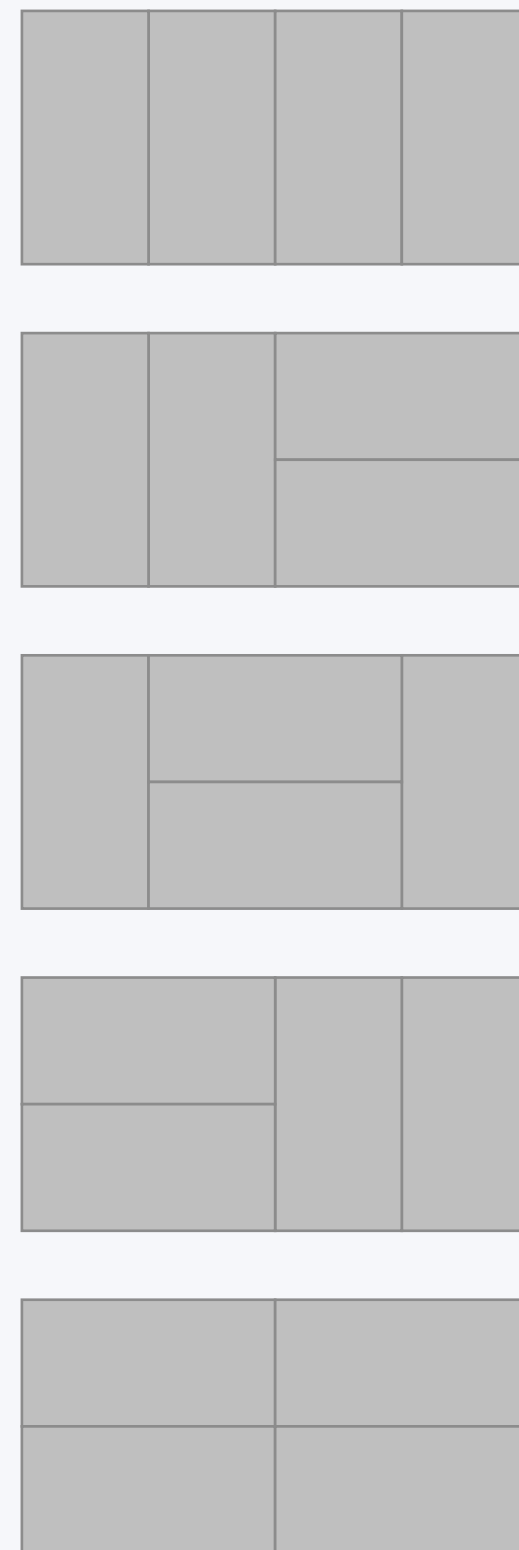
2×n 타일링

<https://www.acmicpc.net/problem/11726>

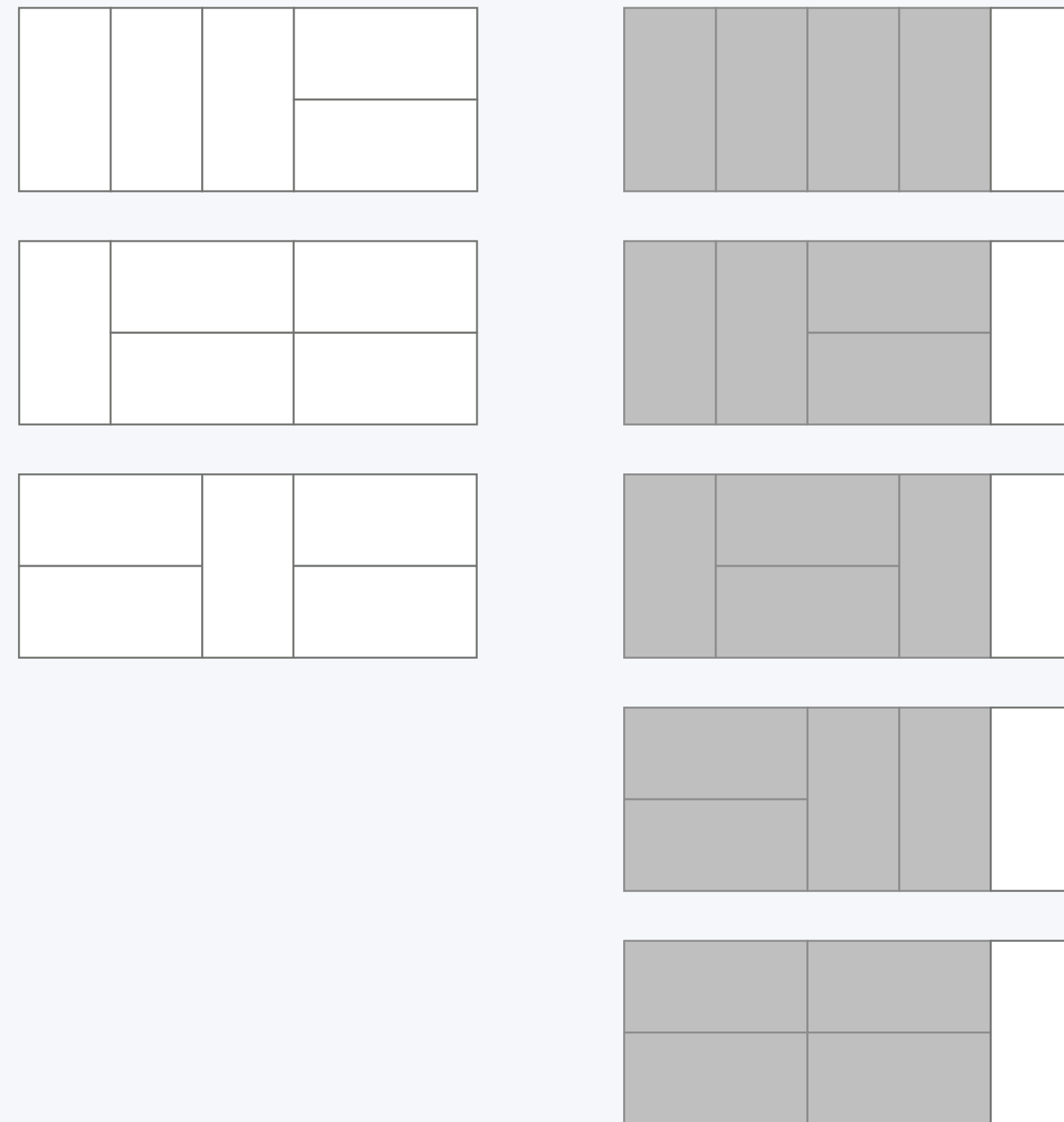
2×3



2×4



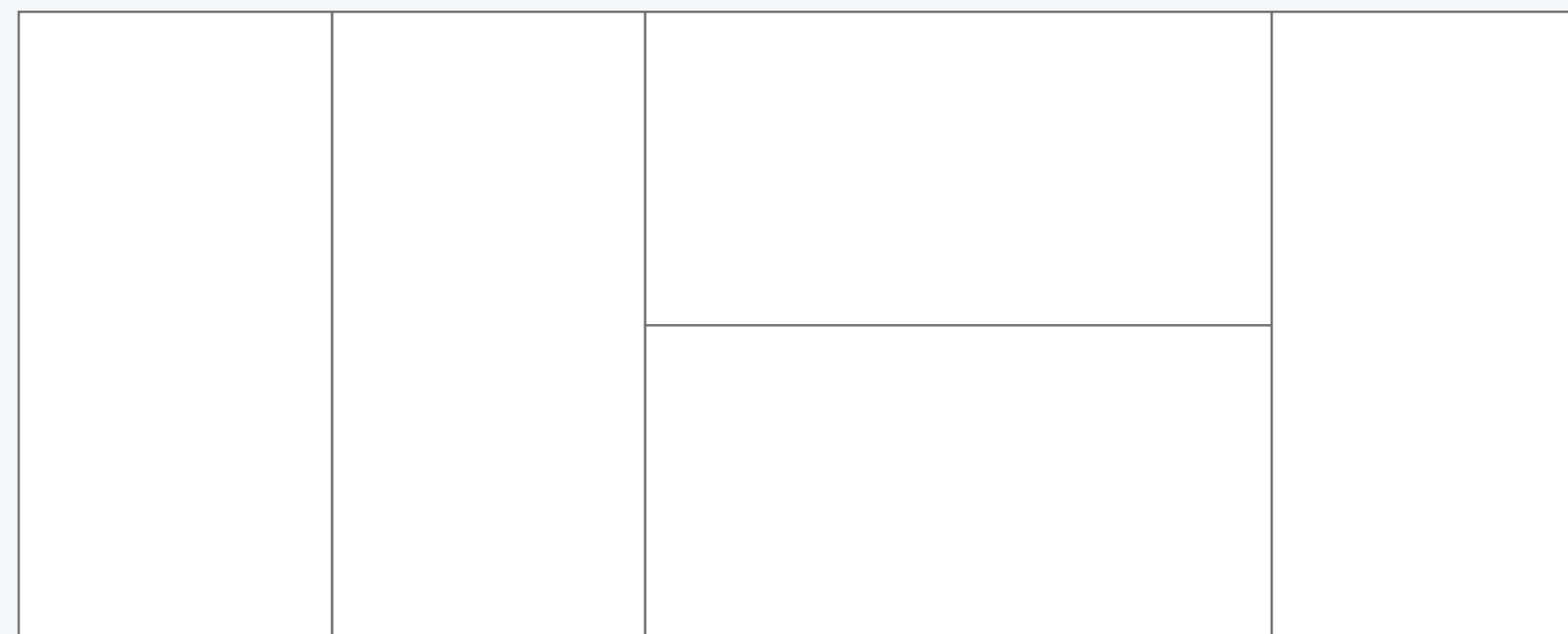
2×5



$2 \times n$ 타일링

<https://www.acmicpc.net/problem/11726>

- $2 \times n$ 직사각형을 1×2 , 2×1 타일로 채우는 방법의 수
- $D[i] = 2 \times i$ 직사각형을 채우는 방법의 수
- $D[i] = D[i-1] + D[i-2]$



2 × n 타일링

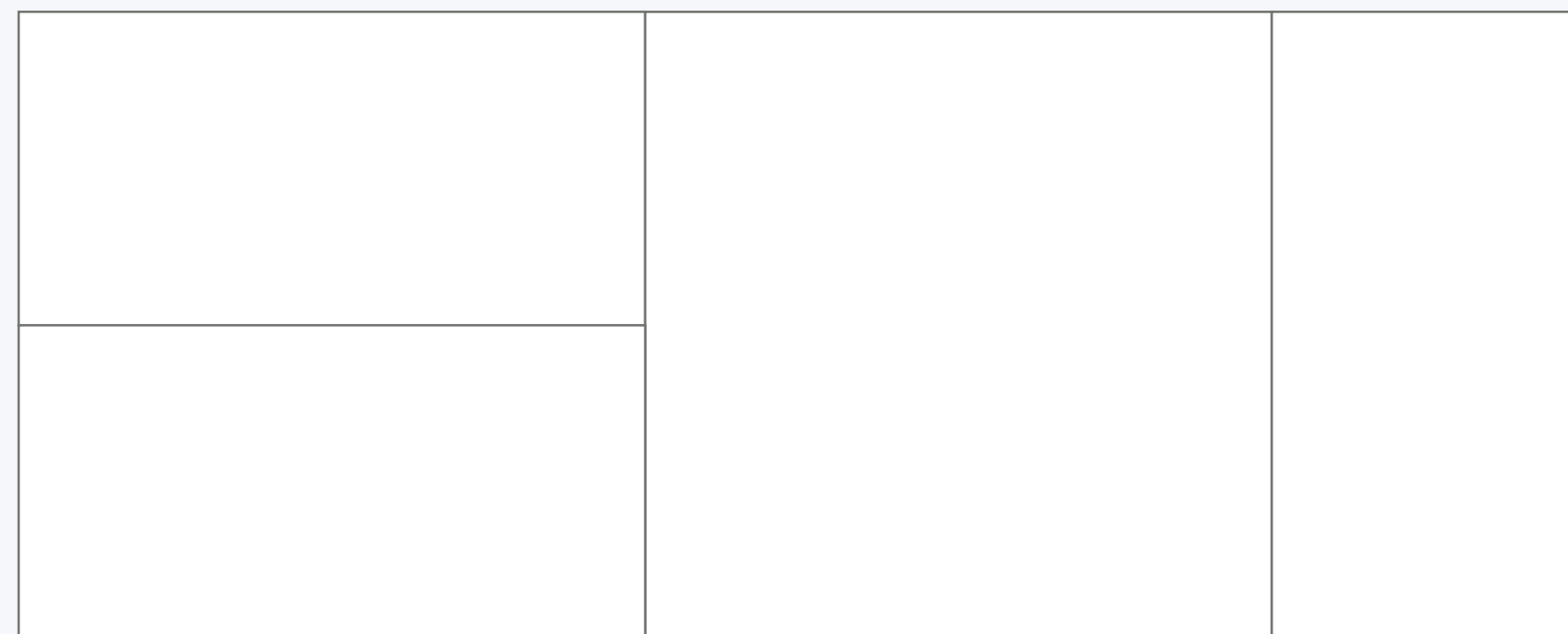
<https://www.acmicpc.net/problem/11726>

- C/C++: <https://gist.github.com/Baekjoon/3527f6fd4771f8c3e1>
- Java: <https://gist.github.com/Baekjoon/53f6e5ec06bfbafad977150df382cf55>

2×n 타일링 2

<https://www.acmicpc.net/problem/11727>

- $2 \times n$ 직사각형을 1×2 , 2×1 , 2×2 타일로 채우는 방법의 수
- 아래 그림은 2×5 를 채우는 방법의 수
- $D[i] = 2 \times i$ 직사각형을 채우는 방법의 수



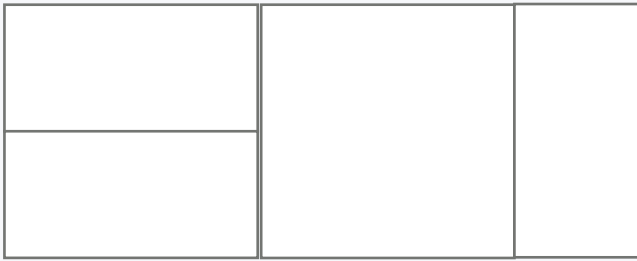
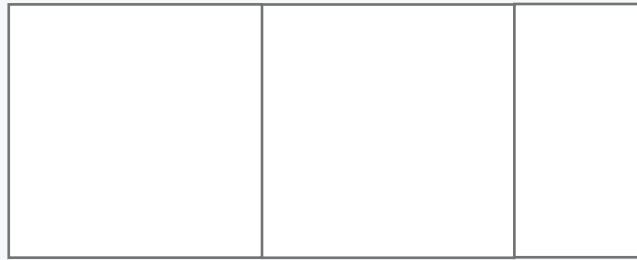
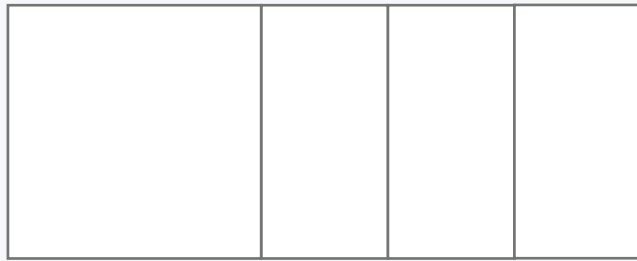
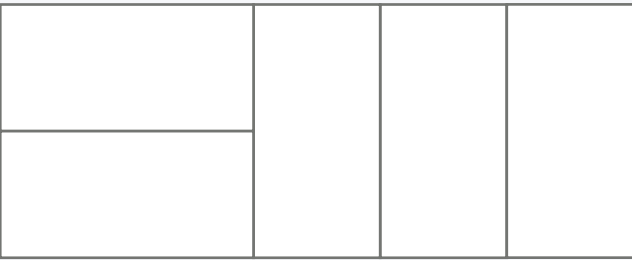
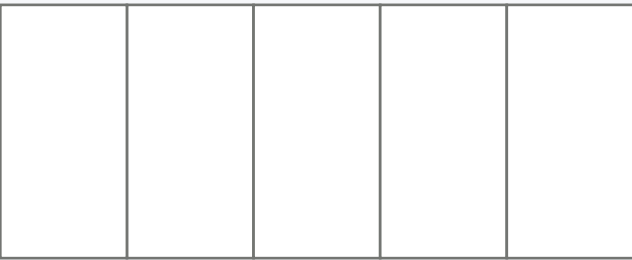
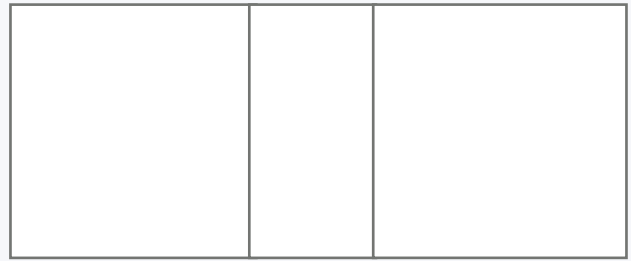
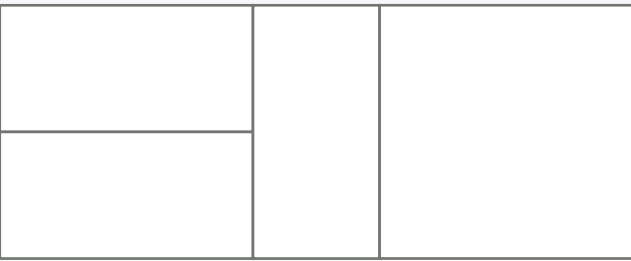
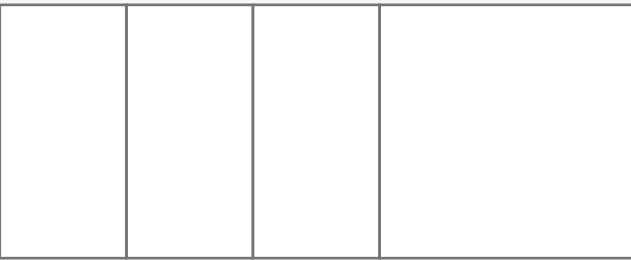
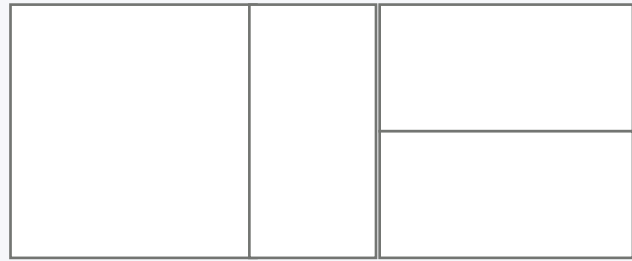
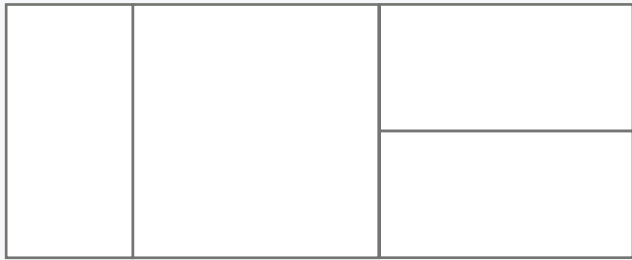
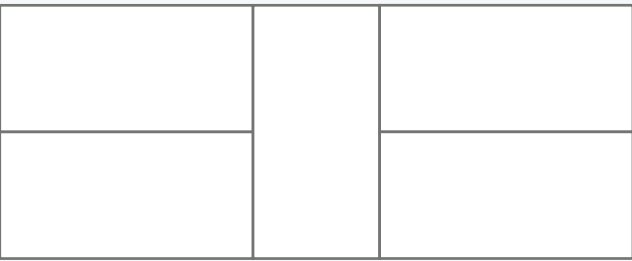
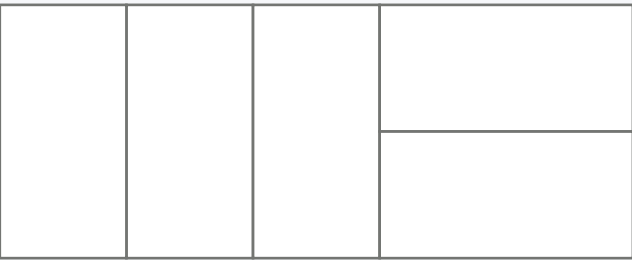
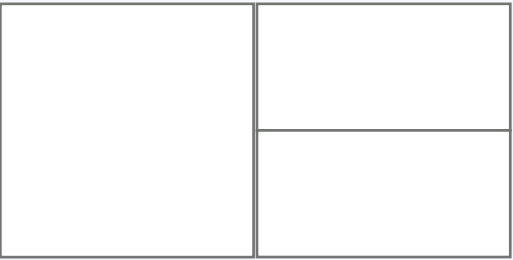
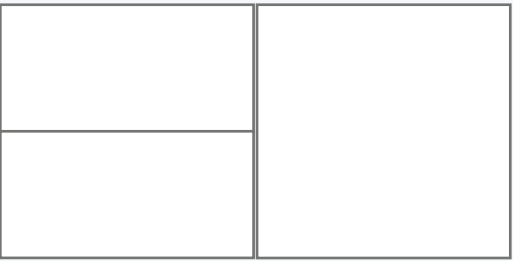
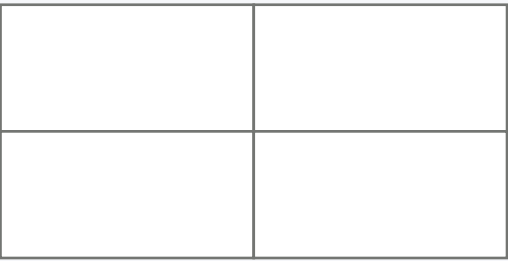
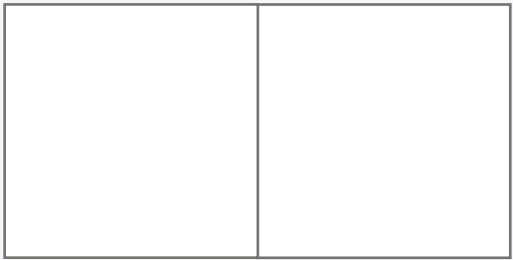
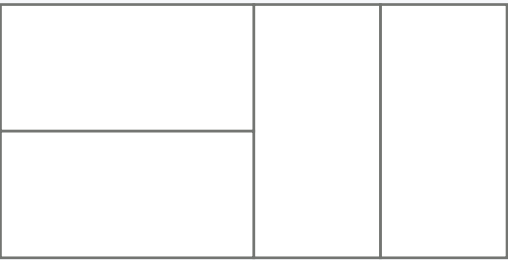
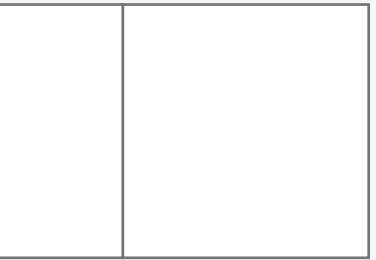
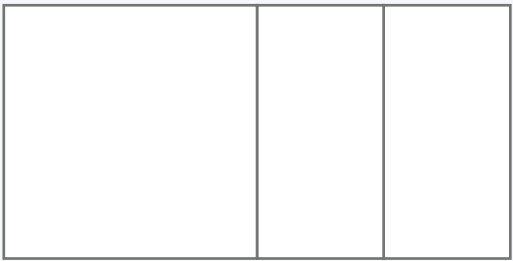
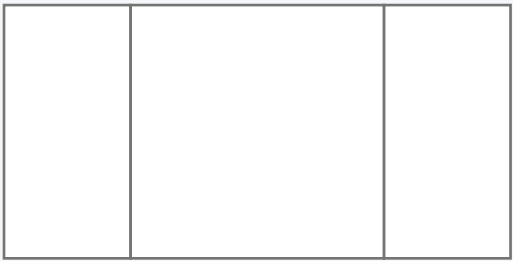
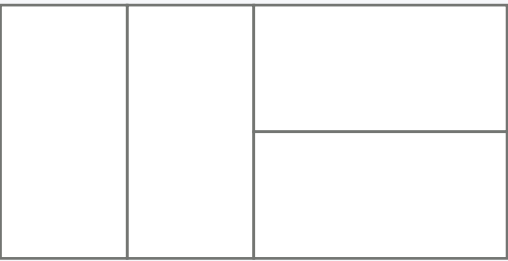
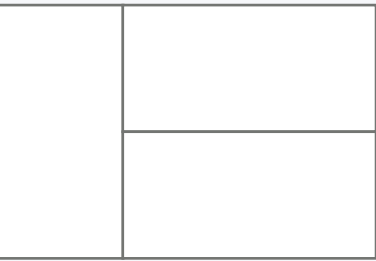
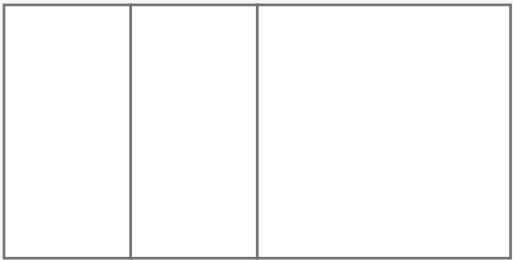
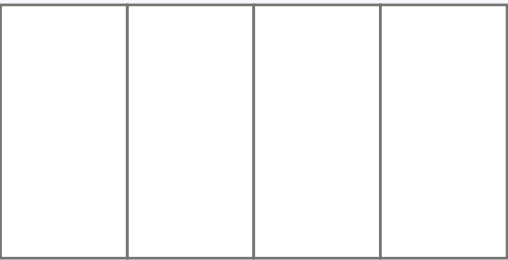
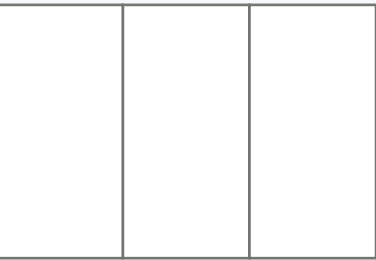
2×n 타일링 2

<https://www.acmicpc.net/problem/11727>

2×3

2×4

2×5



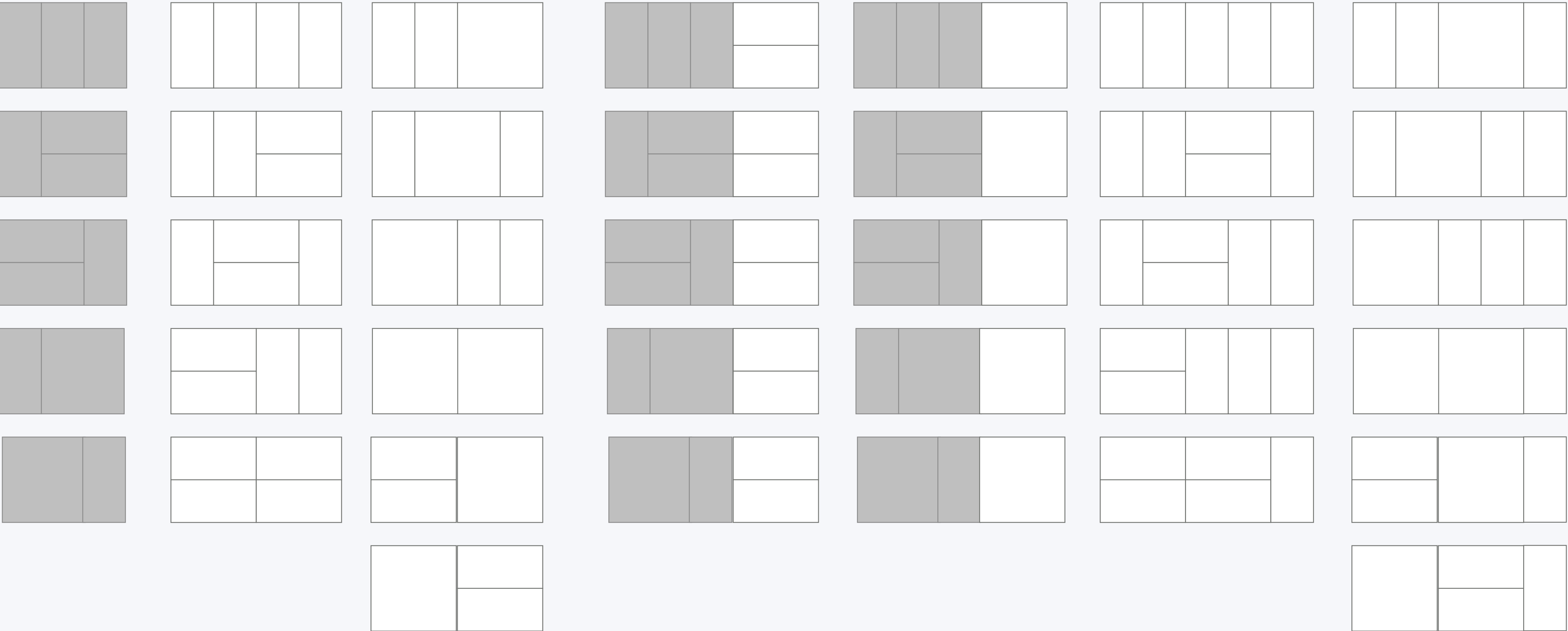
2×n 타일링 2

<https://www.acmicpc.net/problem/11727>

2×3

2×4

2×5



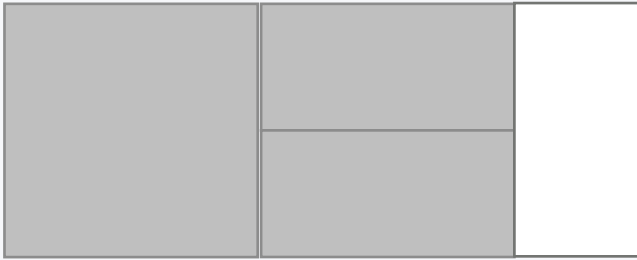
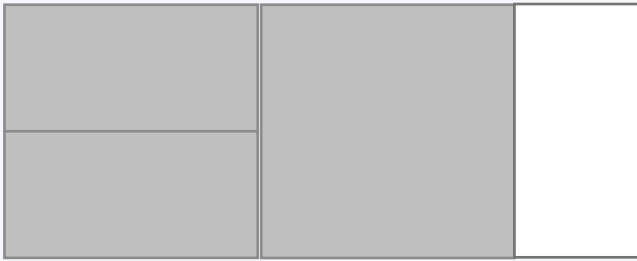
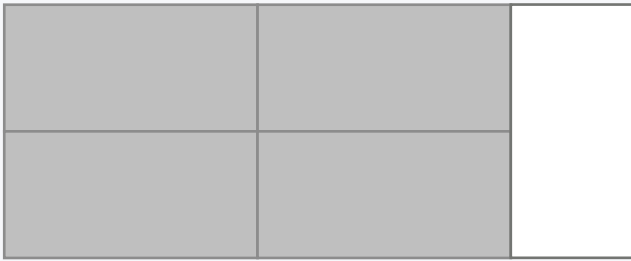
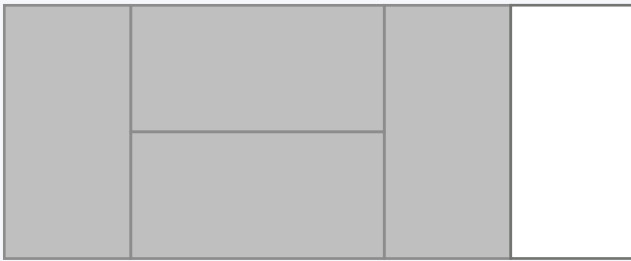
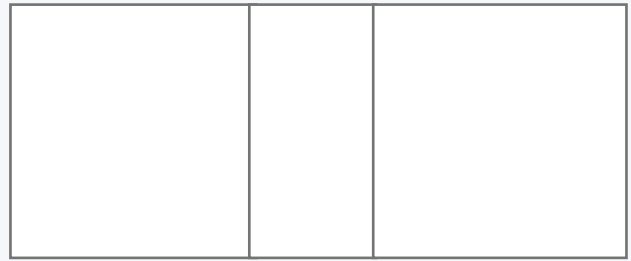
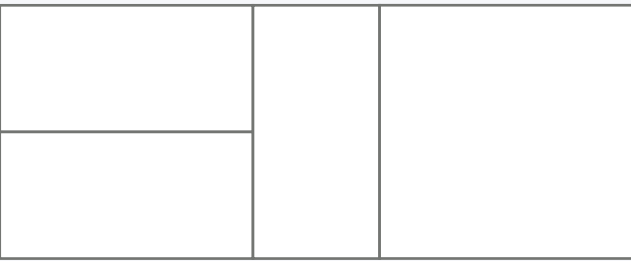
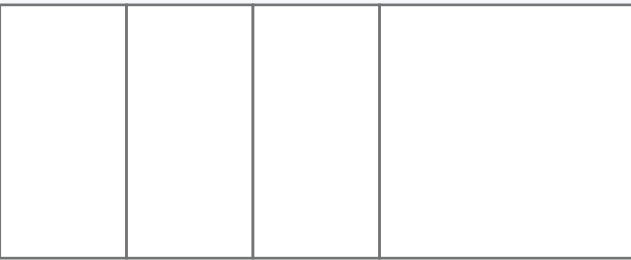
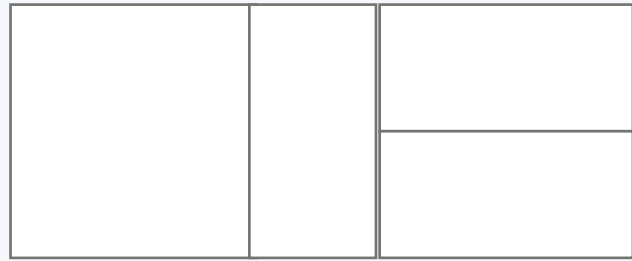
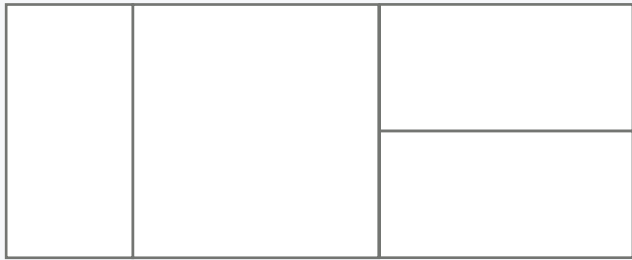
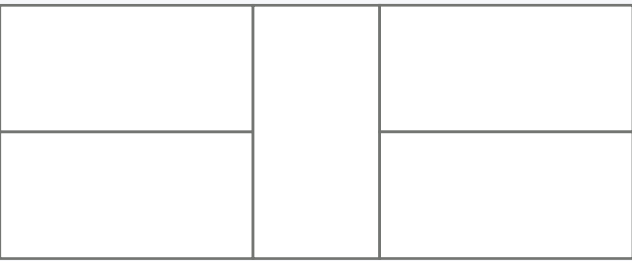
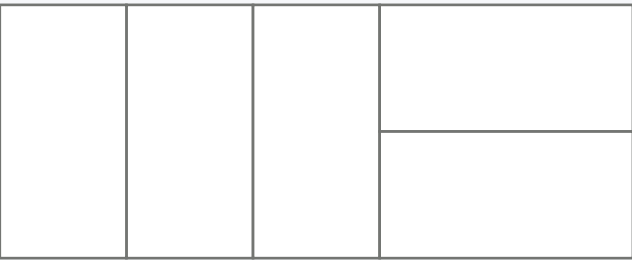
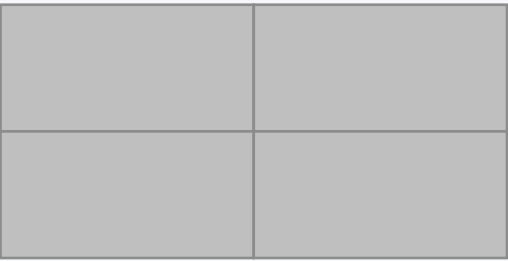
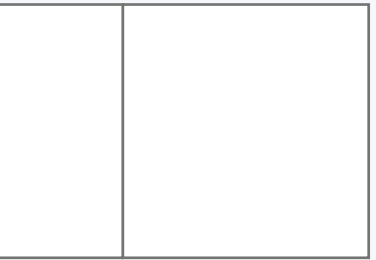
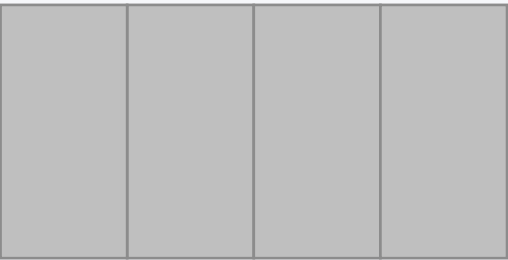
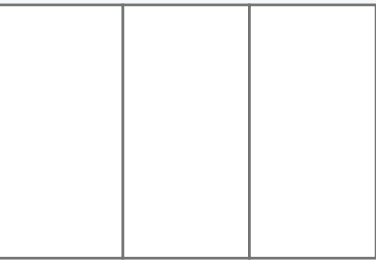
2×n 타일링 2

<https://www.acmicpc.net/problem/11727>

2×3

2×4

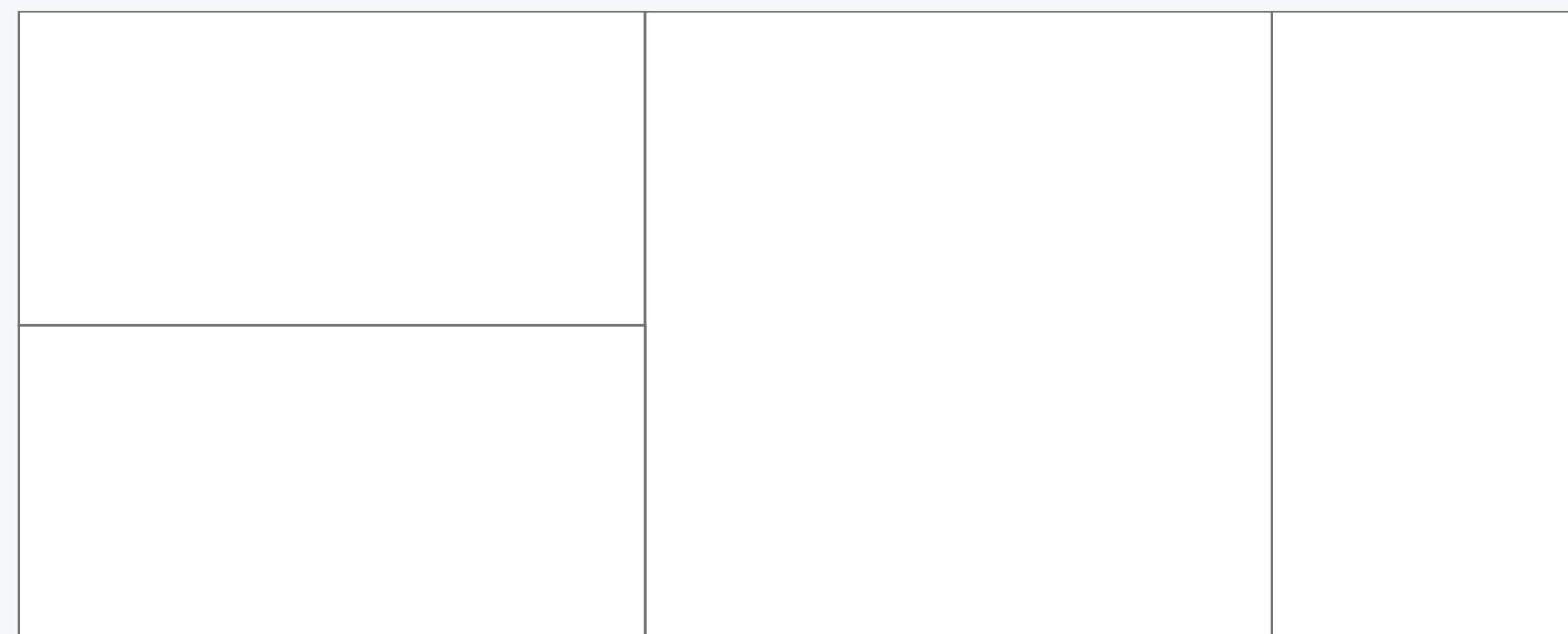
2×5



2×n 타일링 2

<https://www.acmicpc.net/problem/11727>

- 2×n 직사각형을 1×2, 2×1, 2×2타일로 채우는 방법의 수
- $D[i] = 2 \times i$ 직사각형을 채우는 방법의 수
- $D[i] = 2 * D[i-2] + D[i-1]$



2×n 타일링 2

<https://www.acmicpc.net/problem/11727>

- C/C++: <https://gist.github.com/Baekjoon/2ac3e7f55b9f3799d02d>
- Java: <https://gist.github.com/Baekjoon/a6dc09520f1581d9b0f0cacb7057b0a6>

1, 2, 3 더하기

22

<https://www.acmicpc.net/problem/9095>

- 정수 n 을 1, 2, 3의 조합으로 나타내는 방법의 수를 구하는 문제
- $n = 4$
- $1+1+1+1$
- $1+1+2$
- $1+2+1$
- $2+1+1$
- $2+2$
- $1+3$
- $3+1$

1, 2, 3 더하기

<https://www.acmicpc.net/problem/9095>

- $D[i] = i$ 를 1, 2, 3의 조합으로 나타내는 방법의 수

1, 2, 3 더하기

<https://www.acmicpc.net/problem/9095>

- $D[i]$ = i 를 1, 2, 3의 조합으로 나타내는 방법의 수
- $D[i] = D[i-1] + D[i-2] + D[i-3]$

1, 2, 3 더하기

25

<https://www.acmicpc.net/problem/9095>

- C/C++: <https://gist.github.com/Baekjoon/6e4f9e363b3aaef733d1>
- Java: <https://gist.github.com/Baekjoon/e019984a7c7f1ac6bd32>

쉬운 계단 수

<https://www.acmicpc.net/problem/10844>

- 인접한 자리의 차이가 1이 나는 수를 계단 수라고 한다
- 예: 45656
- 길이가 N인 계단 수의 개수를 구하는 문제

쉬운 계단 수

<https://www.acmicpc.net/problem/10844>

- $D[i][j]$ = 길이가 i 이고 마지막 숫자가 j 인 계단 수의 개수
- $D[i][j] = D[i-1][j-1] + D[i-1][j+1]$

쉬운 계단 수

<https://www.acmicpc.net/problem/10844>

```
for (int i=1; i<=9; i++) d[1][i] = 1;
for (int i=2; i<=n; i++) {
    for (int j=0; j<=9; j++) {
        d[i][j] = 0;
        if (j-1 >= 0) d[i][j] += d[i-1][j-1];
        if (j+1 <= 9) d[i][j] += d[i-1][j+1];
        d[i][j] %= mod;
    }
}

long long ans = 0;
for (int i=0; i<=9; i++) ans += d[n][i];
ans %= mod;
```

쉬운 계단 수

<https://www.acmicpc.net/problem/10844>

- C/C++: <https://gist.github.com/Baekjoon/4d98f519afbcdd5d3d0f>
- Java: <https://gist.github.com/Baekjoon/7e4e12ce1b0aa740d5d1>

이친수

<https://www.acmicpc.net/problem/2193>

- 0과 1로만 이루어진 수를 이진수라고 한다.
- 다음 조건을 만족하면 이친수라고 한다.
 1. 이친수는 0으로 시작하지 않는다.
 2. 이친수에서는 1이 두 번 연속으로 나타나지 않는다. 즉, 11을 부분 문자열로 갖지 않는다.
- N자리 이친수의 개수를 구하는 문제

이친수

<https://www.acmicpc.net/problem/2193>

- $D[i][j]$ = i자리 이친수의 개수 중에서 j로 끝나는 것의 개수 ($j=0, 1$)
- 0으로 시작하지 않는다.
- $D[1][0] = 0$
- $D[1][1] = 1$

이친수

<https://www.acmicpc.net/problem/2193>

- $D[i][j]$ = i자리 이친수의 개수 중에서 j로 끝나는 것의 개수 ($j=0, 1$)
- 가능한 경우
- 0으로 끝나는 경우
- 1로 끝나는 경우

이친수

<https://www.acmicpc.net/problem/2193>

- $D[i][j]$ = i 자리 이친수의 개수 중에서 j 로 끝나는 것의 개수 ($j=0, 1$)
- 가능한 경우
- 0으로 끝나는 경우 ($D[i][0]$)
 - 앞에 0과 1이 올 수 있다
 - $D[i-1][0] + D[i-1][1]$
- 1로 끝나는 경우 ($D[i][1]$)
 - 앞에 1은 올 수 없다. 즉, 0만 올 수 있다.
 - $D[i-1][0]$

이친수

<https://www.acmicpc.net/problem/2193>

- $D[i][j]$ = i자리 이친수의 개수 중에서 j로 끝나는 것의 개수 ($j=0, 1$)
- $D[i][0] = D[i-1][0] + D[i-1][1]$
- $D[i][1] = D[i-1][0]$

이친수

35

<https://www.acmicpc.net/problem/2193>

- $D[i]$ = i 자리 이친수의 개수
- 가능한 경우
- 0으로 끝나는 경우
- 1로 끝나는 경우

이친수

<https://www.acmicpc.net/problem/2193>

- $D[i]$ = i 자리 이친수의 개수
- 가능한 경우
- 0으로 끝나는 경우
 - 앞에 0과 1 모두 올 수 있다.
 - $D[i-1]$
- 1로 끝나는 경우
 - 앞에 0만 올 수 있다
 - 앞에 붙는 0을 세트로 생각해서 $i-2$ 자리에 01을 붙인다고 생각
 - $D[i-2]$

이친수

<https://www.acmicpc.net/problem/2193>

- $D[i]$ = i자리 이친수의 개수
- $D[i] = D[i-1] + D[i-2]$

이친수

<https://www.acmicpc.net/problem/2193>

- C/C++: <https://gist.github.com/Baekjoon/49b2bfd22be42707bb88>
- Java: <https://gist.github.com/Baekjoon/7fbfd8d0963139d638de>

제곱수의 합

<https://www.acmicpc.net/problem/1699>

- 주어진 자연수 N 을 제곱수들의 합으로 표현할 때에 그 항의 최소개수를 구하는 문제
- $11=3^2+1^2+1^2$

제곱수의 합

<https://www.acmicpc.net/problem/1699>

- $D[i] = i$ 를 제곱수의 합으로 나타냈을 때, 필요한 항의 최소 개수
- $i = ? + ? + \dots + ? + j$
- 마지막 항이 중요하다.
- 마지막 항이 1인 경우
- 마지막 항이 4인 경우
- 마지막 항이 9인 경우
- 마지막 항이 16인 경우
- 마지막 항이 25인 경우
- ...

제곱수의 합

<https://www.acmicpc.net/problem/1699>

- $D[i] = i$ 를 제곱수의 합으로 나타냈을 때, 필요한 항의 최소 개수
- $i = ? + ? + \dots + ? + j$
- 마지막 항이 중요하다.
- 마지막 항이 1인 경우 $\rightarrow ? + ? + \dots + ? = i-1$
- 마지막 항이 4인 경우 $\rightarrow ? + ? + \dots + ? = i-4$
- 마지막 항이 9인 경우 $\rightarrow ? + ? + \dots + ? = i-9$
- 마지막 항이 16인 경우 $\rightarrow ? + ? + \dots + ? = i-16$
- 마지막 항이 25인 경우 $\rightarrow ? + ? + \dots + ? = i-25$
- ...

제곱수의 합

<https://www.acmicpc.net/problem/1699>

- $D[i] = i$ 를 제곱수의 합으로 나타냈을 때, 필요한 항의 최소 개수
- $i = ? + ? + \dots + ? + j$
- 마지막 항이 중요하다.
- 마지막 항이 1인 경우 $\rightarrow ? + ? + \dots + ? = i-1 \rightarrow D[i-1] + 1$
- 마지막 항이 4인 경우 $\rightarrow ? + ? + \dots + ? = i-4 \rightarrow D[i-4] + 1$
- 마지막 항이 9인 경우 $\rightarrow ? + ? + \dots + ? = i-9 \rightarrow D[i-9] + 1$
- 마지막 항이 16인 경우 $\rightarrow ? + ? + \dots + ? = i-16 \rightarrow D[i-16] + 1$
- 마지막 항이 25인 경우 $\rightarrow ? + ? + \dots + ? = i-25 \rightarrow D[i-25] + 1$
- ...

제곱수의 합

<https://www.acmicpc.net/problem/1699>

- $D[i]$ = i 를 제곱수의 합으로 나타냈을 때, 필요한 항의 최소 개수
- $D[i] = \min(D[i-j^2]+1) \ (1 \leq i \leq j^2)$

제곱수의 합

<https://www.acmicpc.net/problem/1699>

```
for (int i=1; i<=n; i++) {  
    d[i] = i;  
    for (int j=1; j*j <= i; j++) {  
        if (d[i] > d[i-j*j]+1) {  
            d[i] = d[i-j*j]+1;  
        }  
    }  
}
```

제곱수의 합

<https://www.acmicpc.net/problem/1699>

- C/C++: <https://gist.github.com/Baekjoon/66c23e0a64ae7924aa19>
- Java: <https://gist.github.com/Baekjoon/73d0186fef9bbd633f95>

합분해

46

<https://www.acmicpc.net/problem/2225>

- 0부터 N까지의 정수 K개를 더해서 그 합이 N이 되는 경우의 수

합분해

<https://www.acmicpc.net/problem/2225>

- 0부터 N까지의 정수 K개를 더해서 그 합이 N이 되는 경우의 수
- $D[K][N]$ = 0부터 N까지의 정수 K개를 더해서 그 합이 N이 되는 경우의 수
- $? + ? + ? + ? + \dots + ? + L = N$
- 위의 식이 나타내는 값: $D[K][N]$
- $? + ? + ? + ? + \dots + ? = N - L$
- 위의 식이 나타내는 값: $D[K-1][N-L]$
- $D[K][N] = \sum D[K-1][N-L] \ (0 \leq L \leq N)$

합분해

<https://www.acmicpc.net/problem/2225>

```
d[0][0] = 1LL;
for (int i=1; i<=k; i++) {
    for (int j=0; j<=n; j++) {
        for (int l=0; l<=j; l++) {
            d[i][j] += d[i-1][j-l];
            d[i][j] %= mod;
        }
    }
}
```


합분해

<https://www.acmicpc.net/problem/2225>

- C/C++: <https://gist.github.com/Baekjoon/a334580d1729037f5fb1>
- Java: <https://gist.github.com/Baekjoon/354ed0a3657ecbf00c67>

합분해

<https://www.acmicpc.net/problem/2225>

- 0부터 N까지의 정수 K개를 더해서 그 합이 N이 되는 경우의 수
- $D[K][N]$ = 0부터 N까지의 정수 K개를 더해서 그 합이 N이 되는 경우의 수
- $D[K][N] = \sum D[K-1][N-L] \ (0 \leq L \leq N)$
- $D[K][N] = \sum D[K-1][N-L] \ (0 \leq N-L \leq N)$
- $D[K][N] = \sum D[K-1][L] \ (0 \leq L \leq N)$

합분해

<https://www.acmicpc.net/problem/2225>

- 0부터 N까지의 정수 K개를 더해서 그 합이 N이 되는 경우의 수
- $D[K][N]$ = 0부터 N까지의 정수 K개를 더해서 그 합이 N이 되는 경우의 수
- $D[K][N] = \sum D[K-1][L] \ (0 \leq L \leq N)$
- $D[K][N] = D[K-1][0] + D[K-1][1] + \dots + D[K-1][N-1] + D[K-1][N]$
- $D[K][N-1] = D[K-1][0] + D[K-1][1] + \dots + D[K-1][N-1]$

합분해

<https://www.acmicpc.net/problem/2225>

- 0부터 N까지의 정수 K개를 더해서 그 합이 N이 되는 경우의 수
- $D[K][N]$ = 0부터 N까지의 정수 K개를 더해서 그 합이 N이 되는 경우의 수
- $D[K][N] = \sum D[K-1][L] \ (0 \leq L \leq N)$
- $D[K][N] = D[K-1][0] + D[K-1][1] + \dots + D[K-1][N-1] + D[K-1][N]$
- $D[K][N-1] = D[K-1][0] + D[K-1][1] + \dots + D[K-1][N-1]$
- $D[K][N] = D[K][N-1] + D[K-1][N]$

합분해

<https://www.acmicpc.net/problem/2225>

- C++: <https://gist.github.com/Baekjoon/307eeb5e498f68a5f0fab64f6ae0bd05>
- Java: <https://gist.github.com/Baekjoon/3b53e9288ac9d652d8a6e34688c272c8>