EOlymp 23 New Year Tre

Nama : **Timothy Hosia Budianto**
NRP : **50525211098**

Permasalahan dan pendekatan :

New-year tree

For decoration of the New-year tree Peter has in the order a garland from the **n** lamps and **k** of different paints for their painting. How many methods can he to do it, if he must no two identical colors be alongside?

Input

The amount of lamps is n, the amount of different paints is **k** (1 ≤ **k**, **n** ≤ **15**).

Output

Print the number of painting ways. If Peter can not paint a garland after the described requirements, to show out **-1**.

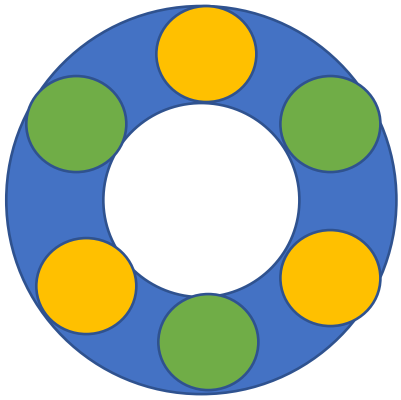
Time limit 1 second
Memory limit 64 MiB

Input example #1	Output example #1
6 2	2

Soal ini membahas hiasan pohon tahun baru (garland). Akan ada 2 inputan pada soal yaitu **n** untuk menentukan jumlah lampu dan **k** untuk menentukan jumlah warna. Soal meminta untuk mencari berapa banyak pola warna cat yang bisa peter susun dengan ketentuan tidak ada warna yang sama berjejeran dengan constrain (**1** ≤ **k**, **n** ≤ **15**).

Abstraksi :

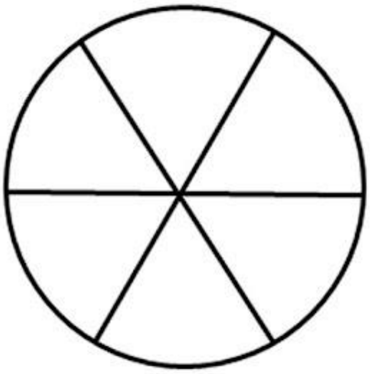
Untuk lebih memahami soal kita akan mengilustrasikan gambaran dari soal. Ilustrasinya adalah sebagai berikut



N = 6 (lampu) ; K = 2 (warna cat)

Jika dilihat sebagai lingkaran penuh akan sebagai berikut

Jika digambarkan dalam bentuk lingkaran , n sector akan sama besar dengan k buah



warna yang tidak boleh sama berdampingan. Kita perlu mencari banyaknya pola warna untuk mewarnai sector tersebut.

Pengerjaan soal tidak bisa dilakukan secara permutasi karna akan menghasilkan waktu dengan kompleksitas tinggu, oleh karena itu akan diselesaikan dengan metode divide and conquer. Misalkan an melambangkan banyaknya kemungkinan, maka a1 = 0 karena tidak terdapat kemungkinan untuk mewarnai 1 sektor.Sedangkan pada 2 sektor jika ditinjau dari prinsip permutasi **k x (k-1)** akan menghasilkan persamaan **a2 = k x (k-1)**. Denga prinsip yang sama kita akan melihat testcase lainnya untuk mendapatkan pola dan algoritma sebagai berikut.

K = 2

n	k	an
1	2	0
2	2	2
3	2	-1
4	2	2
5	2	-1

K = 3

n	k	an
1	3	0
2	3	6
3	3	6
4	3	18
5	3	30

K = 4

n	k	an
1	4	0
2	4	12
3	4	24
4	4	84
5	4	240

Ditinjau dari tabel k = 3 dan k = 4, dapat kita lihat bahwa terdapat keteraturan sebagai pola.

n	k	a_n	Pola
1	4	0	$a_1 = 0$
2	4	12	$a_2 = 4 \times 3 = 12$
3	4	24	$a_3 = (a_2 \times 2) + (a_1 \times 0) = 24$
4	4	84	$a_4 = (a_3 \times 2) + (a_2 \times 3) = (24 \times 2) + (12 \times 3) = 84$
5	4	240	$a_5 = (a_4 \times 2) + (a_3 \times 3) = (84 \times 2) + (24 \times 3) = 240$

Dari tabel k = 4, dapat dilihat bahwa a_1 dan a_2 sudah dibahas sebelumnya. Dimulai dari a3, membentuk relasi rekuren yang membutuhkan 2 suku sebelumnya (an-1 dan an-2).

n	k	a_n	Pola
1	3	0	$a_1 = 0$
2	3	6	$a_2 = 3x\ 2 = 6$
3	3	6	$a_3 = (a_2 \times 1) + (a_1 \times 0) = 24$
4	3	18	$a_4 = (a_3 \times 1) + (a_2 \times 2) = (6 \times 1) + (6 \times 2) = 18$
5	3	30	$a_5 = (a_4 \times 1) + (a_3 \times 2) = (18 \times 1) + (6 \times 2) = 30$

Begitu pula pada tabel k = 3, bahwa pola membentuk relasi rekuren.

Pada k = 5, dapat kita lihat tabel observasi sebagai berikut

n	k	a_n	Pola
1	5	5	$a_1 = 0$
2	5	20	$a_2 = 5 \times 4 = 20$
3	5	60	$a_3 = (a_2 \times 3) + (a_1 \times 0) = 60$
4	5	260	$a_4 = (a_3 \times 3) + (a_2 \times 4) = (60 \times 3) + (20 \times 4) = 260$

5	5	1020	$a_5 = (a_4 \times 3) + (a_3 \times 4) = (260 \times 3) + (60 \times 4) = 1020$

Jika kita membandingkan 2 tabel diatas, maka terdapat perbedaan pada

n	k	a_n	Pola
4	3	18	$a_4 = (a_3 \times 1) + (a_2 \times 2)$
5	3	30	$a_5 = (a_4 \times 1) + (a_3 \times 2)$

n	k	a_n	Pola
4	4	18	$a_4 = (a_3 \times 2) + (a_2 \times 3)$
5	4	30	$a_5 = (a_4 \times 2) + (a_3 \times 3)$

Dari hasil pengamatan tabel diatas, kita dapat memperoleh relasi rekurensi dari an yaitu :

$$\boldsymbol{a_n = (k-2)a_{n-1} + (k-1)a_{n-2}}$$

$$(\boldsymbol{Linear\ Reccurence\ Relation})$$

persamaan sebelumnya dapat kita tulis menjadi :

$$\boldsymbol{a_n - (k-2)a_{n-1} - (k-1)a_{n-2} = 0}$$

$$(\boldsymbol{semua\ variabel\ dipindahruas\ ke\ kiri})$$

dan polinomialnya menjadi :

$$\boldsymbol{m^2 - (k-2)m - (k-1) = 0}$$

Setelah mendapatkan polinomialnya, langkah selanjutnya adalah dengan mencari akar dengan rumus abc.

$$\boldsymbol{x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}}$$

rumus ini kita gunakan untuk mencari m_1 dan m_2 sebagai berikut :

$$m_1 = \frac{-(-k+2) + \sqrt{(-k+2)^2 - 4(-k+1)}}{2\,.\,1}$$

$$m_1 = \frac{k - 2 + k}{2}$$

$$m_1 = \frac{2(k-1)}{2}$$

$$m_1 = k - 1$$

$$m_2 = \frac{-(-k+2) - \sqrt{(-k+2)^2 - 4(-k+1)}}{2\,.\,1}$$

$$m_2 = \frac{k - 2 - k}{2}$$

$$m_2 = \frac{-2}{2}$$

$$m_2 = -1$$

Sesuai dengan teori dari Kenneth H.Rosen, “Descrete Mathematics and Its Applications 8th edition halaman 542

THEOREM 1 Let c_1 and c_2 be real numbers. Suppose that $r^2 - c_1 r - c_2 = 0$ has two distinct roots r_1 and r_2. Then the sequence $\{a_n\}$ is a solution of the recurrence relation $a_n = c_1 a_{n-1} + c_2 a_{n-2}$ if and only if $a_n = \alpha_1 r_1^n + \alpha_2 r_2^n$ for $n = 0, 1, 2, \ldots,$ where α_1 and α_2 are constants.

persamaan $r_2 - c_1 r - c_2 = 0$ merupakan penyelesaian dari relasi rekuren an = $c_1 a_{n-1} + c_2 a_{n-2}$. Terdapat dua akar untuk persamaan $m^2 + (-k + 2)m - k + 1 = 0$. Maka persamaan tersebut dapat kita ubah menjadi

$$\boldsymbol{a_n = c_1 . m_1^n + c_2 . m_2^n}$$

Dengan c_1 dan c_2 adalah konstanta

$$\boldsymbol{a_n = c_1 . (k - 1)^n + c_2 (-1)^n}$$

Karena terdapat 2 variabel yang belum kita ketahui yaitu c_1 dan c_2, maka kita juga memerlukan 2 persamaan agar bisa menyelesaikannya. Sebelumnya kita sudah mengetahui nilai $a_1 = 0$ dan $a_2 = k(k-1)$, sehingga :

$$\boldsymbol{c_1(k - 1) - c_2 = 0}$$

$$\boldsymbol{c_1(k - 1)^2 + c_2 = k(k - 1)}$$

Setelah kita mendapatkan 2 persamaan, kita selesaikan dengan cara eliminasi dan substitusi.

$$c_1(k - 1) - c_2 = 0$$

$$c_1(k - 1)^2 + c_2 = k(k - 1)$$

$$c_1(k - 1) + c_2(k - 1)^2 = k(k - 1)$$

$$\cancel{(k - 1)}(c_1 + c_2(k - 1)) = k\cancel{(k - 1)}$$

$$c_1 + c_1 k - c_1 = k$$

$$c_1 k = k$$

$$\boldsymbol{c_1 = 1}$$

Lalu substitusi untuk mendapatkan nilai c_2

$$c_1(k-1) - c_2 = 0$$

$$c_1(k-1) = c2$$

$$\boldsymbol{c_2 = k - 1}$$

Dari hasil tersebut dapat kita substitusi kembali ke persamaan awal menjadi

$$\boldsymbol{a_n = c_1.(k-1)^n + c_2(-1)^n}$$

$$\boldsymbol{a_n = (k-1)^n + (k-1)(-1)^n}$$

Untuk mempermudah persamaan yang telah kita dapat, kita bisa membagi menjadi 2 persamaan untuk ganjil dan genap.

Jika n ganjil :

$$\boldsymbol{a_n = (k-1)^n - (k-1)}$$

Jika n genap :

$$\boldsymbol{a_n = (k-1)^n + (k-1)}$$

Implementasi :

Dengan pseudocode :

```
inisialisasi variable n, k
n, k ←
If (n = 1)
  print k
Else if ( k=1 ) or ( n odds and k = 2)
  print -1
Else
    If ( n odds)
        ans = power ((y-1), x) – (y+1)
    Else
        ans = power ((y-1), x) + (y-1)
Print ans
```

Source Code :

```
#include <stdio.h>
#include <math.h>
```

```
typedef long long ll;

ll n, k;

ll solve(ll x, ll y){
    ll ans;
    if (n&1)
        ans = (ll)pow((double)(y-1), (double)x) - (y - 1);
    else
        ans = (ll)pow((double)(y-1), (double)x) + (y - 1);
    return ans;
}

int main(){
    scanf("%lld %lld", &n, &k);
    if (n ==1 ) printf("%lld\n", k);
    else if ((k == 1) || (n&1&&k == 2))printf("-1\n");
    else printf("%lld\n", solve(n, k));
    return 0;
}
```

Penjelasan Source Code :

Dari source code, pertama inisialisasi variabel n dan k. Pada fungsi main, input variabel n dan k. Jika n = 1, maka print k. Tetapi jika k = 1 atau n ganjil dan k = 2, print -1. Else memanggil fungsi solve dengan passing variabel n dan k.

Pada fungsi solve, initialisasi variabel ans. jika n ganjil, ans = (y-1) pangkat x lalu dikurangi (y-1). Else ans = (y-1) pangkat x lalu ditambah dengan y-1. Setelah itu return ans.