

# Sistem Keamanan Ruangan Dinamis Dr. Senku

Oleh: Augist

Time Limit: 1 s
Memory Limit: 256 MB



Di sebuah pusat penelitian berteknologi tinggi, Dr Senku bingung untuk menjaga ruangan dalam pusat tersebut dan bagaimana memasang sistem keamanan otomatis agar efektif, di Pusatt Teknologi tersebut terdapat **n ruangan** yang masing-masing dijaga oleh sistem keamanan otomatis. Setiap ruangan memiliki **kode akses** dengan tingkat otorisasi berbeda:

## Kode Akses

- A   => Akses penuh
- B   => Akses staf
- C   => Akses pengunjung

Namun, terdapat **satu ruangan utama** bernomor **r**, yang menjadi pusat referensi sistem keamanan. Sistem keamanan menerapkan **aturan otomatis**, yang dapat mengubah kode akses ruangan lain (*selain ruangan r*).

Coba kalian bantu Dr Senku untuk menentukan **berapa banyak ruangan** yang **pada akhirnya** memiliki **kode yang sama** seperti ruangan r setelah semua aturan diberlakukan.

Sistem menerapkan aturan secara **berurutan**:

### Aturan Pemrosesan Kode

#### \$ Aturan A (Dasar — WAJIB untuk semua ruangan selain r)

Berdasarkan kode ruangan utama  $r$ , ruangan lain akan otomatis diberi:

Kode $r$	Kode Ruangan lain
A	C
B	A
C	B

#### \$ Aturan B (Jika $n$ adalah bilangan GANJIL)

Jika jumlah ruangan  $n$  ganjil, maka untuk ruangan bernomor genap:

abaikan Aturan A

dan langsung set: **kode = kode  $r$**

#### \$ Aturan C (Jika $r$ adalah bilangan PRIMA)

Jika nomor ruangan  $r$  **prima**, maka untuk ruangan lain yang nomornya **juga prima**:

Kode Otomatis awal	Kode setelah Aturan C
A	B
B	C
C	A

Aturan ini bisa diterapkan **setelah aturan A/B**.

### Batasan

$$2 \leq n \leq 100$$

$$1 \leq r \leq n$$

$$\text{Kode } r \in \{A, B, C\}$$

### Format Input

$n$ $r$
X

### Format Output

<i>Jumlah ruangan yg kode nya sama dengan kode r</i>
--

### Contoh Inputan 1 :

5 3
-----

B
---

### Contoh Output 1 :

2
---

### Contoh Inputan 2 :

6 1
-----

A
---

### Contoh Output 2 :

0
---

# Dr. Senku's Dynamic Room Security System

Oleh: Augist

Time Limit: <b>1 s</b>
Memory Limit: <b>256 MB</b>



In a high-tech research center, Dr. Senku is struggling to secure the center's rooms and how to effectively install an automated security system. The Technology Center has **n** rooms, each guarded by an automated security system. Each room has an **access code** with a different authorization level:

## Access Code

A => Full Access

B => Staff Access

C => Visitor Access

However, there is **one main room**, numbered **r**, which serves as the security system's reference point. The security system applies automatic rules that can change the access codes of other rooms (except room **r**).

Help Dr. Senku determine how many rooms ultimately have the same code as room **r** after all rules are applied.

The system applies the rules in sequence:

Code Processing Rules

### \$ Rule A (Basic — MANDATORY for all rooms except r)

Based on the main room code r, other rooms are automatically assigned:

<i>r code</i>	Another room Code
A	C
B	A
C	B

### Rule B (If n is an ODD number)

If the number of rooms n is odd, then for even-numbered rooms:

Ignore Rule A

and directly set: code = code r

### Rule C (If r is a PRIME number)

If room number r is prime, then for other rooms with prime numbers:

First automatic code	Code after Rule C applied :
A	B
B	C
C	A

This rule can be applied after the A/B rule.

### Constraints

$$2 \leq n \leq 100$$

$$1 \leq r \leq n$$

Code  $r \in \{A, B, C\}$

### Format Input

n r
X

### Format Output

<i>Total Number of rooms whose code is the same as code r</i>
---

**Example Input 1:**

5 3
B

**Example Output 1 :**

2
---

**Example Input 2 :**

6 1
A

**Example Output 2 :**

0
---