

Codeforces Round 273 (Div. 2)

Problema B: *Random Teams*

Prof. Edson Alves – UnB/FGA

n participants of the competition were split into m teams in some manner so that each team has at least one participant. After the competition each pair of participants from the same team became friends.

Your task is to write a program that will find the minimum and the maximum number of pairs of friends that could have formed by the end of the competition.

Input

The only line of input contains two integers n and m , separated by a single space ($1 \leq m \leq n \leq 10^9$) – the number of participants and the number of teams respectively.

Output

The only line of the output should contain two integers k_{\min} and k_{\max} – the minimum possible number of pairs of friends and the maximum possible number of pairs of friends respectively.

Exemplos de entrada e saída

Entrada

5 1

3 2

6 3

Saída

10 10

1 1

3 6

Solução com complexidade $O(1)$

- Para maximizar o número de pares de amigos é preciso formar o maior grupo possível
- Isto significa que devem ser formados $m - 1$ grupos com um único participante e um grande grupo com o $A = n - (m - 1)$ restantes
- Assim,

$$k_{\max} = \binom{A}{2} = \frac{A(A - 1)}{2}$$

- Já para minimizar o número de amigos os participantes devem ser distribuídos da maneira mais uniforme possível

Solução com complexidade $O(1)$

- Pela divisão de Euclides, $n = mq + r$, com $0 \leq r < m$
- Logo devem ser formados $m - r$ grupos com $B = q$ membros, e os r grupos restantes terão um membro a mais, isto é, $B + 1$ membros
- Deste modo,

$$k_{\min} = (m - r) \binom{B}{2} + r \binom{B + 1}{2} = (m - r) \frac{B(B - 1)}{2} + r \frac{(B + 1)B}{2}$$

- Esta solução tem complexidade $O(1)$

Solução com complexidade $O(1)$

```
6 pair<ll, ll> solve(ll n, ll m)
7 {
8     auto A = n - (m - 1);
9     auto kmax = A*(A - 1)/2;
10
11     auto B = n / m, r = n % m;
12     auto kmin = (m - r)*(B*(B - 1)/2) + r*((B + 1)*B/2);
13
14     return {kmin, kmax};
15 }
```