

약수의 개수

Description

양의 정수 N, M에 대하여, N보다 크거나 같고 M보다 작거나 같은 수 중에서 약수의 개수가 가장 많은 수를 출력하시오.

단,  $1 \leq N \leq M \leq 2^{32}-1$ ,  $M - N \leq 10000$

만약 약수의 개수가 가장 많은 수가 여러 개이면, 그 중에서 가장 큰 수를 출력한다.

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

typedef unsigned long long ulong_t;
typedef unsigned int uint_t;

uint_t divcnt(uint_t n);
void solve(uint_t N, uint_t M);

int main()
{
    uint_t N, M;
    scanf("%u %u", &N, &M);
    solve(N, M);
}
```

위의 코드는 서버에 포함되어 있으므로, divcnt(), solve() 함수를 구현하여 제출하면 된다.

Input

첫 번째 줄에 양의 정수 N, M이 주어진다.

단,  $1 \leq N \leq M \leq 2^{32}-1$ ,  $M - N \leq 10000$

Output

첫 번째 줄에약수의 개수가 가장 많은 수 K를 출력한다.

만약 약수의 개수가 가장 많은 수가 여러 개이면, 그 중에서 가장 큰 수를 출력한다.

두 번째 줄에 K의 약수의 개수를 출력한다.

Sample Input 1

1 10

Sample Output 1

10
4

Sample Input 2

1 100

Sample Output 2

96
12

Sample Input 3

4294957295 4294967295

Sample Output 3

4294967040
864

```
1 uint_t divcnt(uint_t n) // 카운트 높으로
2 {
3     uint_t c = 0;
4     if (n == 1)
5     {
6         c = 1;
7         return c;
8     }
9     for (ulong_t x = 1; x * x <= n; x++)
10    {
11        if (x * x == n)
12        {
13            c += 1;
14        }
15        else
16        {
17            if (n % x == 0)
18            {
19                c += 2;
20            }
21        }
22    }
23    return c;
24 }
25
26 void solve(uint_t N, uint_t M)
27 {
28     unsigned long long max_c = 0;
29     unsigned long long max = 0;
30
31     for (unsigned long long x = N; x <= M; x++)
32     {
33         uint_t c = divcnt(x);
34         if (max_c <= c)
35         {
36             max_c = c;
37             max = x;
38         }
39     }
40
41     printf("%llu\n", max);
42     printf("%llu\n", max_c);
43 }
```