

Be careful

Giới hạn thời gian: 0.5s **Giới hạn bộ nhớ:** 256M

C: 0.3s

C++: 0.3s

PyPy: 0.12s

Python: 0.3s

Chú ý giới hạn thời gian của bài toán

Cho một số nguyên N , yêu cầu phân tích số N thành các thừa số nguyên tố.

Biết rằng số nguyên N được tạo thành từ đoạn code như sau:

```
function isPrime(N):  
    // Hàm dùng để kiểm tra N có phải số nguyên tố không.  
  
function MakeNumber(S, K):  
    N = 1  
    while K > 0:  
        # vv lấy ngẫu nhiên giữa 1 và 100.  
        if isPrime(S) and random_int(1, 100) == 1:  
            K -= 1  
            N *= S  
        S += 1  
    return N
```

Trong đó, hàm *random_int* là hàm ngẫu nhiên phân phối đều.

Ngoài ra, bạn sẽ được biết thêm hai tham số S, K được truyền vào hàm *MakeNumber*.

Input

- Dòng đầu tiên chứa hai số S, K .
- Dòng thứ hai chứa số nguyên N .

Constraints

- $1 \leq S \leq 10^{18}$
- $1 \leq S \leq N < 10^{2000}$
- $2 \leq K \leq 100$

Output

- In ra các thừa số nguyên của số nguyên N theo thứ tự tăng dần. Biết rằng giá trị của các thừa số nguyên tố trong đáp án sẽ không vượt quá 9, 223, 372, 036, 854, 775, 807.

Subtask

- Subtask 1 (40%): $N \leq 2 \times 10^{18}$, S, K được đảm bảo phù hợp với N được tạo ra.
- Subtask 2 (60%): Không có giới hạn gì thêm.

Sample input

```
123 2
62291
```

Sample output

```
167 373
```

