

소인수 카운팅

Description

2보다 크거나 같은 양의 정수  $N$ 을 소인수 분해하고, 각 소인수가 몇 개 나타나는가를 출력하시오.

예를 들어,  $120 = 2 * 2 * 2 * 3 * 5$  의 경우에는 2가 3번, 3이 1번, 5가 1번 나타난다.

아래 코드에서 `factorize()` 함수를 완성하여 제출하시오.

```
#include <stdio.h>
#include <math.h>
#define MAX 32

/*
 * The function factorize() factorizes n
 * and store the fequency of factors in the array a[],
 * which is passed by the caller as an argument.
 */
unsigned factorize(unsigned n, unsigned a[]);

int main()
{
    unsigned N, a[MAX];
    scanf("%u", &N);
    unsigned len = factorize(N, a);
    unsigned p = a[0], cnt = 0;
    for (unsigned i = 0; i < len; i++) {
        if (p == a[i]) {
            cnt++;
        } else {
            printf("%u %u\n", p, cnt);
            p = a[i]; cnt = 1;
        }
    }
    printf("%u %u\n", p, cnt);
}
```

위 코드에서 `factorize()`의 리턴 값인 `len`은 소인수의 개수이다.

배열 `a[]`의 값은 소인수가 오름차순으로 저장되어 있어야 한다.

예를 들어,

`a[] = { 2, 2, 3, 3, 5 }`

로 저장되어 있으면위 `main()` 함수의 코드는

```
2 2
3 3
5 1
```

로 출력하도록 하는 코드이다.

Input

첫 번째 줄에 양의 정수  $N$ 이 주어진다.

$2 \leq N \leq 2^{32} - 1$

Output

$N$ 을 소인수 분해하고,  $N$ 의 소인수가 각각 몇 번 나타나는가를 출력한다.

한 줄에 하나의 소인수를 오름차순으로 출력하도록 한다.

Sample Input 1

415800

Sample Output 1

2 3
3 3
5 2
7 1
11 1

Sample Input 2

4294967291

Sample Output 2

4294967291 1

```
1 unsigned factorize(unsigned n, unsigned a[])
2 {
3     // len은 소인수 전체의 개수이다.
4     unsigned cnt = 0;
5
6     for (unsigned i = 2; i <= sqrt(n); i++)
7     {
8         if (n % i == 0)
9         {
10             while ((n % i) == 0)
11             {
12                 n /= i;
13                 a[cnt] = i;
14                 cnt += 1;
15             }
16         }
17     }
18
19     if (n > 1)
20     {
21         a[cnt] = n;
22         cnt += 1;
23     }
24
25     return cnt;
26 }
27 }
```