소인수의 합

Description

양의 정수 N, M에 대하여, N보다 크거나 같고 M보다 작거나 같은 정수 중에서 소인수의 합이 최대인 수와 그 수의 소인수의 합을 출력하시오.

단, 1 <= N <= M <= 2^32 -1, M - N <= 10000

예를 들어, 180 = 2 * 2 * 3 * 3 * 5 의 소인수의 합은 2 + 2 + 3 + 3 + 5 = 15 이다.

단, 소인수의 합이 최대인 수가 여러 개 있으면 그 중 가장 작은 수를 소인수의 합이 최대인 수로 출력한다.

아래의 코드에서 sum_of_factors() 함수를 완성하여 제출하시오.

단, 소인수의 개수가 1개인 소수는 소인수의 합이 0이라고 가정한다.

```
#include <stdio.h>
#include <math.h>
typedef unsigned long long_t;
/* returns the sum of factors*/
long_t sum_factors(long_t n);
int main()
   long_t n, m, maxcnt, maxval;
   scanf("%llu %llu", &n, &m);
   maxcnt = maxval = 0;
   for (long_t i = n; i <= m; i++) {
        long_t s = sum_factors(i);
       if (s > maxcnt) {
           maxcnt = s;
           maxval = i;
   printf("%llu\n", maxval);
   printf("%llu", maxcnt);
```

Input

첫 번째 줄에 양의 정수 N, M이 주어진다.

 $1 \le N \le M \le 2^32 -1$

Output

Language: C

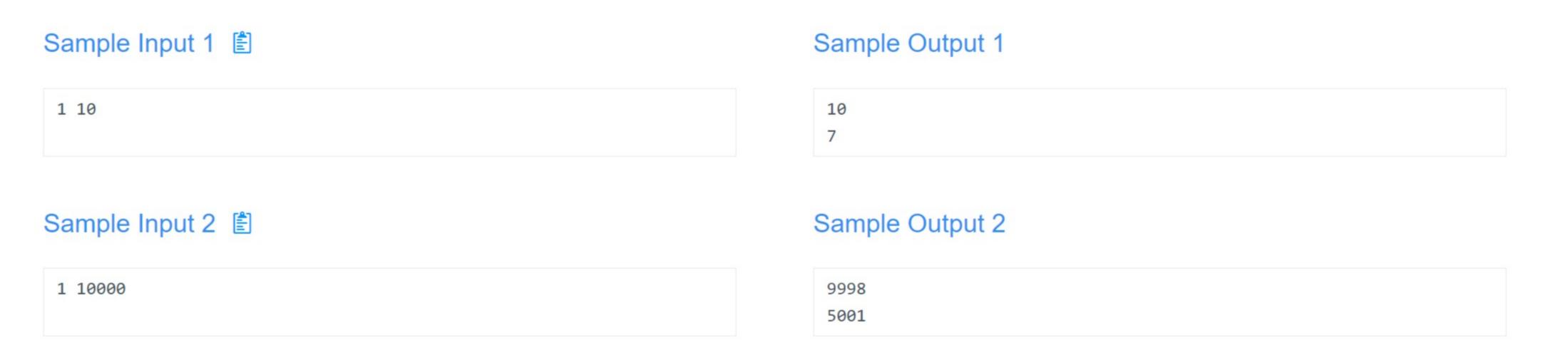
You have solved the problem

Contest has ended

첫 번째 줄에 N보다 크거나 같고 M보다 작거나 같은 정수 중에서 소인수의 합이 가장 큰 수를 출력한다.

두 번째 줄에 그 수의 소인수의 합을 출력한다.

G



Theme:

Solarized Light

```
1 long_t sum_factors(long_t n)
 2 * {
        long_t sum = 0;
        for (long_t i = 2; i <= sqrt(n); i++)</pre>
 6 ▼
            if (n % i == 0)
 8 .
                 while ((n % i) == 0)
10 ▼
                     n /= i;
11
12
                     sum += i;
13
14
15
16
        if (n > 1)
17
18 🔻
19
            sum += n;
21
22
        if (sum == n)
23 ▼
            return 0;
24
25
26
        else
27 ▼
28
            return sum;
29
30 }
```