

Разбор задачи «Музыкальная дорожка»

Для дальнейших нужд будем нумеровать ступеньки с 0. Если участник использует число p , то он будет прыгать по ступенькам со следующими номерами:

$$\{0, p \bmod n, 2p \bmod n, 3p \bmod n, \dots, (n-1)p \bmod n\}$$

Если p не делит n , то среди этих остатков по модулю n встречаются всевозможные, поэтому обладатель интервала p прыгнет по разу на каждую ступеньку и произведёт звук, равный суммарному числу звуков, издаваемых всеми ступеньками.

Для делителей же ситуация будет несколько иная: если $n = pk$, тогда в ряду $\{0, p, 2p, 3p, \dots, (n-1)p\}$, будут по p раз встречаться числа $\{0, p, 2p, 3p, \dots, (k-1)p\}$

Понятно, что делители n не больше миллиона. Так как простых чисел до 200,000 не больше 100,000, то нам достаточно найти только их, чтобы учесть все возможные делители.

В свете этого решение будет следующим:

1. С помощью метода решета Эратосфена находим простые до n .

```
sq = ((int) sqrt(n*1.0) ) + 1;
for( i = 2; i <= sq; i++ ) {
    if( not_prime[i] ) continue;
    for( j = i*i; j <= n; j += i )
        not_prime[j] = 1;
}
```

2. Собираем их вместе.

```
for( number_of_prime = 0, i = 2; i <= n; i++ )
    if( !not_prime[i] )
        prime[number_of_prime++] = i;
```

3. Вычисляем сумму всех по модулю m .

```
for( i = 0; i < n; i++ )
    sum += a[i];

sum %= m;
```

4. Далее вычисляем ответ для всех простых не больших n .

```

for( i = 0; i < min(number_of_prime, k); i++ ) {
    if( n % prime[i] )
        r += sum;
    else
        for( j = 0; j < n; j += prime[i] )
            r += (__int64) a[j] * prime[i];

    r %= m;
}

```

5. Если остались ещё простые, учитываем их.

```

if( number_of_prime < k )
    r += (__int64) sum * (k - np);

```

Суммарно решение будет выглядеть так:

```

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

#define maxn 200000
#define maxk 100000

#define min(a,b) ((a)<(b)?(a):(b))

int n, k, m, number_of_prime;
int a[maxn], not_prime[maxn+1], prime[maxk];
__int64 sum, r;

int main(void)
{
    int i, j, sq;

    freopen("mustrack.in", "r", stdin);
    freopen("mustrack.out", "w", stdout);

    scanf("%d%d%d", &n, &k, &m);
    for( i = 0; i < n; i++ )
        scanf("%d", a+i);

    sq = ((int) sqrt(n*1.0) ) + 1;
    for( i = 2; i <= sq; i++ ) {
        if( not_prime[i] ) continue;

```

```

    for( j = i*i; j <= n; j += i )
        not_prime[j] = 1;
}

for( j = 0, i = 2; i <= n; i++ )
    if( !not_prime[i] )
        prime[j++] = i;

number_of_prime = j;

for( i = 0; i < n; i++ )
    sum += a[i];

sum %= m;

for( i = 0; i < min(number_of_prime, k); i++ ) {
    if( n % prime[i] )
        r += sum;
    else
        for( j = 0; j < n; j += prime[i] )
            r += (__int64) a[j] * prime[i];

    r %= m;
}

if( number_of_prime < k )
    r += (__int64) sum * (k - number_of_prime);

printf("%d", r%m);
return 0;
}

```