Product 1 Modulo N

2021 年 5 月 21 日

Product 1 Modulo N

Now you get Baby Ehab' s first words: "Given an integer n, find the longest subsequence of $1, 2, , n1$ whose product is 1 modulo n." Please solve the problem.

A sequence b is a subsequence of an array a if b can be obtained from a by deleting some (possibly all) elements. The product of an empty subsequence is equal to 1.

Input

The only line contains the integer n $(2n10^5)$.

Output

The first line should contain a single integer, the length of the longest subsequence.

The second line should contain the elements of the subsequence, **in increasing order**.

If there are multiple solutions, you can print any.

Examples

input

```
5
```

output

```
3
1 2 3
```

input

```
8
```

output

```
4
1 3 5 7
```

Note

In the first example, the product of the elements is 6 which is congruent to 1 modulo 5. The only longer subsequence is $[1, 2, 3, 4]$. Its product is 24 which is congruent to 4 modulo 5. Hence, the answer is $[1, 2, 3]$.

思路

若 $gcd(x, n) \neq 1$，那么不存在一个数 y 使得 $xy \equiv 1\ (mod\ n)$ ，所以这 k 个数中必不包括与 n 不互质的数，而所有与 n 互质的数的乘积 B 满足 $B \equiv 1\ (mod\ n)$ 或者 $B \equiv -1\ (mod\ n)$ 。

```cpp
#include <bits/stdc++.h>

using namespace std;

const int maxn = 1e6+1;
int a[maxn], tot;

int main() {
    ios::sync_with_stdio(false);
    cin.tie(nullptr);
    int n;
    cin >> n;
    long long ans = 1;
    for(int i = 1; i < n; ++ i) {
        if (__gcd(i, n) == 1) {
            a[++tot] = i;
            ans = ans * i % n;
        }
    }
```

```
    if (ans != 1) tot --;
    cout << tot << "\n";
    for(int i = 1; i <= tot; ++ i) {
        cout << a[i] << " \n"[i==tot];
    }
    return 0;
}
```