BSGS

```cpp
#include <bits/stdc++.h>
using namespace std;

typedef pair<int, int> PII;
typedef long long ll;

int bsgs(int a, int b, int p){
    if(1 % p == b % p) return 0;
    int k = sqrt(p) + 1;
    unordered_map<int, int> hash;
    for(int i = 0, j = b % p; i < k; i++){
        hash[j] = i;
        j = (ll)j * a % p;
    }
    int ak = 1;
    for(int i = 0; i < k; i++) ak = (ll)ak * a % p;
    for(int i = 1, j = ak; i <= k; i++){
        if(hash.count(j)) return (ll)i * k - hash[j];
        j = (ll)j * ak % p;
    }
    return -1;
}

int main(){
    int a, b, p;
    cin >> a >> b >> p;
    int res = bsgs(a, b, p);
    if(res == -1) cout << "no solution";
    else cout << res;
    return 0;
}
```

EXBSGS(拓展BSGS)

```cpp
#include <bits/stdc++.h>
using namespace std;

typedef pair<int, int> PII;
typedef long long ll;
const int INF = 0x3f3f3f3f;

int exgcd(int a, int b, int &x, int &y){
    if(!b){
        x = 1, y = 0;
        return a;
    }
    int d = exgcd(b, a % b, y, x);
    y -= a / b * x;
```

```
    return d;
}

int bsgs(int a, int b, int p){
    if(1 % p == b % p) return 0;
    int k = sqrt(p) + 1;
    unordered_map<int, int> hash;
    for(int i = 0, j = b % p; i < k; i++){
        hash[j] = i;
        j = (ll)j * a % p;
    }
    int ak = 1;
    for(int i = 0; i < k; i++) ak = (ll)ak * a % p;
    for(int i = 1, j = ak; i <= k; i++){
        if(hash.count(j)) return i * k - hash[j];
        j = (ll)j * ak % p;
    }
    return -INF;
}

int exbsgs(int a, int b, int p){
    b = (b % p + p) % p;
    if(1 % p == b % p) return 0;
    int x, y;
    int d = exgcd(a, p, x, y);
    if(d > 1){
        if(b % d) return -INF;
        exgcd(a / d, p / d, x, y);
        return exbsgs(a, (ll)b / d * x % (p / d), p / d) + 1;
    }
    return bsgs(a, b, p);
}

int main(){
    int a, b, p;
    while(scanf("%d%d%d", &a, &p, &b), a || b || p){
        int res = exbsgs(a, b, p);
        if(res < 0) cout << "No Solution" << '\n';
        else cout << res << '\n';
    }
    return 0;
}
```