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

Useful defines:

```cpp
#define LCM(x , y) (x*y/__gcd(x , y))
#define GCD(x , y) (__gcd(x , y))
```

GCD Applications:

```cpp

//1. Euclid's: if we want the gcd of (a , b) and we can write a/b=c , a%b=d
//   then GCD(a , b)=GCD(b , d) and the O(max(a , b));
int GCD(int a , int b){
    if(b==0) return a;
    else return GCD(b , a%b);
}

//2. GCD function with O(min(a , b)):
int GCD(int a , int b){
    int mini = min(a , b) , ans;
    for (int i = mini; i > 0; i--)
    {
        if(a%i==0 && b%i==0){
            ans = i;
            return ans;
        }
    }
}

//3. GCD rules:
//  - GCD(m*a , m*b) = m * GCD(a , b)
//  - GCD(a/m , b/m) = GCD(a , b) / m
//  - GCD(a , b) * LCM(a , b) = a*b
//  - GCD(0 , 0) = LCM(0 , 0) = 0
//  - GCD(a , b , c) = GCD(a , GCD(b , c))
//  - GCD(a , m) = 1 ⟹ GCD(m , a*b) = GCD(m , b)
//  - GCD(a , b) = d ⟹ GCD(a/d , b/d) = 1
//  - GCD(a , b) = GCD(b , a%b)
//  - GCD(a , a+1) = 1

//4. find three distinct positive integers a , b , c such that
//   a+b+c=n and GCD(a , b) = c
void solution(){
    int n; cin>>n;
    if(n%2==0) cout<<n-3<<" "<<2<<" "<<1<<endl;
    else{
        for (int i = 2; i < n/2; i++)
        {
            if(__gcd(n-i-1 , i)==1){
                cout<<n-i-1<<" "<<i<<" "<<1<<endl;
                break;
            }
        }
    }
}

//5. coPrime rules:
//  - sum of any two coPrime number is coPrime with their product
//  - factoring of two coPrime number is common only in number 1
//  - if factor of a[i] = p1 * p2 * p3 than p1*p2 + p3 is coPrime with a[i]
```