#include <bits/stdc++.h>

using namespace std;

vector<int> findMissingRepeatingNumbers(vector<int> a) {

int n = a.size(); // size of the array

int xr = 0;

//Step 1: Find XOR of all elements:

for (int i = 0; i < n; i++) {

xr = xr ^ a[i];

xr = xr ^ (i + 1);

}

//Step 2: Find the differentiating bit number:

int number = (xr & ~(xr - 1));

//Step 3: Group the numbers:

int zero = 0;

int one = 0;

for (int i = 0; i < n; i++) {

//part of 1 group:

if ((a[i] & number) != 0) {

one = one ^ a[i];

}

//part of 0 group:

else {

zero = zero ^ a[i];

}

}

for (int i = 1; i <= n; i++) {

//part of 1 group:

if ((i & number) != 0) {

one = one ^ i;

}

//part of 0 group:

else {

zero = zero ^ i;

}

}

// Last step: Identify the numbers:

int cnt = 0;

for (int i = 0; i < n; i++) {

if (a[i] == zero) cnt++;

}

if (cnt == 2) return {zero, one};

return {one, zero};

}

int main()

{

vector<int> a = {3, 1, 2, 5, 4, 6, 7, 5};

vector<int> ans = findMissingRepeatingNumbers(a);

cout << "The repeating and missing numbers are: {"

<< ans[0] << ", " << ans[1] << "}\n";

return 0;

}