Nombre: Cuitpa Mamani, Joaquin Esteban

Github: https://github.com/xnabux/Algebra-abstracta.git

```cpp
#include <iostream>
#include <string>
#include <stdlib.h>
#include <time.h>
#include <vector>
#include <math.h>

using namespace std;

int mod(int a, int b) {
	int r = a - ((a / b) * b);
	if (r < 0)
		r = a - (((a / b) - 1) * b);
	return r;
}
int valor_absoluto(int a) {
	if (a < 0) {
		a = a * -1;
	}
	return a;
}

int euclide(int a, int b) {
	if (valor_absoluto(b) > valor_absoluto(a))
		return euclide(b, a);
	if (b == 0) {
		return a;
	}
	if (mod(a, 2) == 0 && mod(b, 2) == 0) {
		return 2 * euclide(a / 2, b / 2);
	}
	if (mod(a, 2) == 0 && mod(b, 2) == 1) {
		return euclide(a / 2, b);
	}
	if (mod(a, 2) == 1 && mod(b, 2) == 0) {
		return euclide(a, b / 2);
	}
	return euclide((valor_absoluto(a) - valor_absoluto(b)) / 2, b);
}

int euclide_exten(int a, int n) {
	int s1 = 1;
	int s2 = 0;
	while (n > 0) {
		int q = a / n;
		int r = a - q * n;
		a = n;
		n = r;
		int s = s1 - q * s2;
		s1 = s2;
		s2 = s;
	}
```

```
	return s1;
}

int inversa_mul(int a, int n) {
	if (euclide(a, n) == 1) {
		int x = euclide_exten(a, n);
		if (x < 0) {
			x = x + n;
		}
		return x;
	}
	else {
		cout << "No tiene inversa\n";
	}
}

void pedir_valores(vector<int>& x, vector <int>& a,vector<int>& p) {
	cout << "Forma de la ecuacion   ->    x = a mod p" << endl;
	for (int i = 0; i < x.size(); i++) {
		cout << "\nComplete los valores:" << endl;
		cout << "\nIngrese x: ";
		cin >> x[i];//rellenar
		cout << "\nIngrese a: ";
		cin >> a[i];//rellenar
		cout << "\nIngrese n ";
		cin >> p[i];//rellenar
	}
	cout << "\nEstas son la ecuaciones que digitaste:" << endl;
	for (int i = 0; i < x.size(); i++) {
		cout << x[i] << " = " << a[i] << " mod " << p[i]<<endl;
	}
}

bool comprobar(vector<int>& x, vector<int>& a, vector<int>& p) {
	for (int i = 0; i < p.size(); i++) {
		for (int j = i + 1; j < p.size(); j++) {
			if (euclide(p[i], p[j]) != 1) {
				cout << "\nAlgunos valores de p no son comprimos entre
si";
				return false;
			}
		}
	}

	for (int i = 0; i < a.size(); i++) {
		if (a[i] < 0) {
			a[i] = mod(a[i], p[i]);
		}
	}

	for (int i = 0; i < x.size(); i++) {
		if (x[i] != 1) {
			int inversa_x = inversa_mul(x[i], p[i]);
			if (inversa_x) {
				a[i] = a[i] * inversa_x;
				x[i] = 1;
			}
			else {
```

```
                        cout << "\nNo hay inversa de x";
                        return false;
                    }
                }
                return true;
        }
}

vector<int> Cuerpo_resto_chino(int tam) {
        vector<int>x(tam), a(tam), p(tam);
        pedir_valores(x, a, p);
        if (comprobar(x, a, p) == false) {
                cout << "\nNo se puede resolver: ´D";
                return { 0 };
        }
        //
        int _P = 1;
        for (int i = 0; i < p.size(); i++) {
                _P *= p[i];
        }
        vector<int>valor_p(tam);
        for (int i = 0; i < valor_p.size(); i++) {
                valor_p[i] = _P / p[i];
        }
        //
        vector<int>q(tam);
        for (int i = 0; i < q.size(); i++) {
                int valor_de_p = valor_p[i];
                if (valor_de_p >= p[i]) {
                        valor_de_p = mod(valor_de_p, p[i]);
                }
                valor_de_p = inversa_mul(valor_de_p, p[i]);
                q[i] = mod(valor_de_p, p[i]);
        }
        //
        int _x = 0;
        for (int i = 0; i < tam; i++) {
                int aux = mod(a[i], _P) * mod(valor_p[i], _P) * mod(q[i], _P);
                _x = _x + mod(aux, _P);
        }
        _x = mod(_x, _P);
        vector<int> resultado(2);
        resultado[0] = _x;
        resultado[1] = _P;
        return resultado;
}
```

```
int main() {
        int cantidad;
        cout << "Cantidad de ecuciones: ";
        cin >> cantidad;
        vector<int>ecua(2);
        ecua = Cuerpo_resto_chino(cantidad);

        cout << "\n\nEcuacion: ";
        cout << "x = " << ecua[0] << " + " << ecua[1] << "k" << endl;
        return 0;
```

```
}
```