

} fraccion;

POO

② Fracción { Numerador y denominador int dentro del struct. }

Suma

Simplificar

Imprimir fracción

struct fraccion {

int numerador;

int denominador;

}

struct fraccion suma (struct fraccion *frac1, struct fraccion *frac2) {

struct fraccion fracR;

fracR.denominador = frac1.denominador * frac2.denominador;

fracR.numerador = frac1.numerador * frac2.denominador

+ frac2.numerador * frac1.denominador;

simplificar (&fracR);

return fracR;

void simplificar (struct fraccion *fracR) {

int i = 2;

do {

if (fracR.denominador % i == 0 && fracR.numerador % i == 0) {

fracR.denominador /= i;

fracR.numerador /= i;

i = 2;

}

else {

if (i > fracR.denominador || i > fracR.numerador) break;

i++;

}

while(1);

}

*

void imprimer_fractions (Struct fraction fracR)

{

cout << fracR.numerador << "/" << fracR.denominador << endl;

}

③ 1: fecha ← Estructura

2: Num dias del Mes

3: es fin de Mes { 0, 1 }
 ↗ si si es fin de mes.

4: dia siguiente { cambiar dia, mes y año si corresponde }

① type del struct {

int dia;

int mes;

int año;

} fecha;

② int Num dias del Mes (fecha * la fecha) {

if (la fecha.mes == 1 || la fecha.mes == 3 || la fecha.mes == 5
 || la fecha.mes == 7 || la fecha.mes == 8 || la fecha.mes == 10
 || la fecha.mes == 12) return 31;

else if (la fecha.mes == 2) return 28;

return 30;

③ int es fin de Mes (fecha la fecha) {

if (la fecha.dia == Num dias del Mes (la fecha)) return 1;
 return 0;

}

```

④ void diasiguiente ( fecha *lafecha) {
    if ( esfindeMes (lafecha) == 1) {
        if ( lafecha → mes == 12) {
            lafecha → mes = 1;
            lafecha → año ++;

        else {
            lafecha → mes ++;
        }
        lafecha → dia = 1;
    }
    else {
        lafecha → dia ++;
    }
}

```