

### 1. Vacía Cola [2 puntos]

Suponiendo la clase

```
class Usuario {
    string _nombre;
    float _prioridad;
public:
    string nombre() const { return _nombre; }
    float prioridad() const { return _prioridad; }
    // ...
};
```

haz una acción vacia cola con cabecera

```
void vacia cola(vector<Usuario>& cola, int N, float P);
```

que borra del vector "cola", empezando por el principio, la cantidad de N usuarios cuya prioridad sea superior a P. (Si la cola tiene menos de N usuarios con prioridad mayor que P lógicamente hay que borrarlos todos.)

### 2. Cuenta Bancaria 2 [4 puntos]

Implementa una clase CuentaBancaria, que representa una cuenta bancaria, con los siguientes datos: la oficina (int), el número de cuenta (int), el DNI del titular (string) y el saldo (la cantidad de dinero que hay en la cuenta, un real). La clase debe tener:

1. Los 3 constructores básicos. El constructor por defecto crea una cuenta con todo a 0 y el titular vacío. El constructor normal recibe todos los datos excepto el saldo que se pone a 0.0.
2. Métodos para consultar el DNI del titular, la oficina, el número de cuenta y el saldo.
3. Un método para ingresar una cantidad de dinero en la cuenta.
4. Un método para sacar una cantidad de la cuenta.
5. Un método mismo\_titular que devuelve true si dos cuentas tienen el mismo titular.
6. Un método transferencia que permite realizar una transferencia de una cantidad entre dos cuentas.
7. Un método escribe que escribe en un ostream la oficina, el número de cuenta (separados por guiones) y el saldo, en el formato "1856-1000565430: 934.05" (o sea "oficina-número: saldo").

### 3. Mas Antigua [2 puntos]

Suponiendo la declaración de la clase Fecha siguiente (que *no hay que implementar*)

```
class Fecha {
    // ...
public:
    Fecha(int dia, int mes, int anyo);
    Fecha(const Fecha& f);
    int dia() const;
    int mes() const;
    int anyo() const;
    bool anterior(const Fecha& f) const;
};
```

haz un programa que lee un fichero "fechas.txt" con fechas en el formato

```
5/9/2010
1/5/2001
2/4/2019
```

y muestra por pantalla, en el mismo formato, la fecha más antigua.

### 4. Numeritos [2 puntos]

Considera el siguiente programa:

```
void ins(vector<int>& v, int x) {
    vector<int>::iterator it = v.begin();
    while (it != v.end() && *it < x) {
        it++;
    }
    v.insert(it, x);
}

int main() {
    int x;
    vector<int> v;
    while (cin >> x) {
        ins(v, x);
    }
    vector<int>::const_reverse_iterator rit;
    for (rit = v.rbegin(); rit != v.rend(); rit++) {
        cout << *rit << ' ';
    }
    cout << endl;
}
```

Pon 3 ejemplos de entradas (inventadas por tí) y qué salida producen, intentando ilustrar distintos casos. Luego explica de forma concisa qué hace el programa.