PROGRAMACIÓN 2. Curso 2017-18. 1ª prueba voluntaria de evaluación. Grupo 411

Esta es la primera prueba de **evaluación voluntaria** que se plantea en la asignatura ***Programación 2*** a los alumnos matriculados en el grupo de mañanas. Tiene un valor de **10 puntos**. Debe ser resuelta individualmente y deberá ser presentada **antes de las 24 horas del 8 de abril de 2018**.

Considérese el esquema de fuente que se muestra más abajo. El ejercicio pide desarrollar $minMax$ (4.0 puntos) y $maxValCol$ (6.0 puntos) de una manera puramente recursiva (no está permitido utilizar bucles).

```
//−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−
// Predicado funcional, para simplificar la especificación y anotación
// V_COL(M, C, F) = ∑_{β=0}^{F} M[β][C]
//−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−

//Pre: true
//Post: maxValCol(M, N) = Max α ∈ [0, N − 1].V_COL(M, α, N − 1)
template<typename T, int N>
T maxValCol(const T M[N][N]) {
    ...
}

//−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−
//Pre: 0 ≤ d ∧ d < #v − 2 ∧ v = V_0
//Post: (v[0] = Min α ∈ [0, d].V_0[α]) ∧ (v[1] = Max α ∈ [0, d].V_0[α]) ∧ (∀α ∈ [0, d].v[α + 2] = V_0[α])
template<typename T>
void minMax(T v[], int d) {
    ...
}
```

Es preciso tener en cuenta las siguientes consideraciones:

- En caso de llevarse a cabo inmersiones, se deberá indicar el tipo de inmersión realizada.
- La correcta especificación (como reflejo de un diseño correcto) es fundamental para que el ejercicio pueda ser evaluado. Por lo tanto, las funciones inmersoras utilizadas deberán ser especificadas formalmente.
- Es también imprescindible que el código funcione correctamente.

Como resultado de esta prueba se entregará, a través de Moodle, un único fichero, denominado **prueba_1.cpp**[1], que contenga el fuente C++ que resuelva lo anteriormente planteado. El fuente debe ser completo, de manera que se pueda compilar y ejecutar. Es importante tener presente que para su evaluación se sustituirá la función *main* entregada por una función *main* específica para su comprobación.

No se admitirán trabajos que no se presenten a través de la plataforma **Moodle2**, trabajos presentados fuera de plazo, ni trabajos plagiados total o parcialmente.

[1] Dado que una primera comprobación se va a hacer de manera automática, no respetar el nombre del fichero daría lugar a considerar que no se ha entregado el trabajo

Una posible solución al ejercicio 1

```
//-----------------------------------------------------------------
// V_COL(M, C, F) = sum_{β=0}^{F} M[β][C]
//-----------------------------------------------------------------

//Pre:  0 ≤ c ∧ c ≤ N − 1 ∧ 0 ≤ f ∧ f ≤ N − 1
//Post: valColI(M, c) = V_COL(M, c, f)
template<typename T, int N>
T valColI(const T M[N][N], int c, int f) {

    if (f==0) {
        return M[f][c];
    } else {
        return M[f][c] + valColI(M, c, f−1);
    }
}

//Pre:  0 ≤ c ∧ c ≤ N − 1
//Post: valCol(M, c) = V_COL(M, c, N − 1)
template<typename T, int N>
T valCol(const T M[N][N], int c) {

    return valColI(M, c, N−1);
}

//Pre:  0 ≤ c ∧ c ≤ N − 1
//Post: maxValColI(M, c) = Max α ∈ [0, c].V_COL(M, α, N − 1)
template<typename T, int N>
T maxValColI(const T M[N][N], int c) {

    if (c==0) {
        return valCol(M, 0);
    } else {
        T valC = valCol(M, c);
        T maxResto = maxValColI(M, c−1);
        if (valC>maxResto) {
            return valC;
        } else {
            return maxResto;
        }
    }
}

//Pre:  true
//Post: maxValCol(M, N) = Max α ∈ [0, N − 1].V_COL(M, α, N − 1)
template<typename T, int N>
T maxValCol(const T M[N][N]) {

    return maxValColI(M,N−1);
}
```

Una posible solución al ejercicio 2

```
//----------------------------------------------
//Pre:  0 ≤ i ∧ i ≤ d ∧ d < #v − 2 ∧ ∀α ∈ [i, d].v[α + 2] = V0[α] ∧ ∀α ∈ [0, i − 1].v[α] = V0[α])∧
//      min = Min α ∈ [i, d].V0[α]) ∧ (max = Max α ∈ [i, d].V0[α])∧
//Post: (∀α ∈ [0, d].v[α + 2] = V0[α])
template<typename T>
void minMaxI(T v[], int i, int d, T& min, T& max) {

    if (i>0) {
        if (v[i-1] < min) {
            min = v[i-1];
        } else if (v[i-1] > max) {
            max = v[i-1];
        }
        v[i+1] = v[i-1];
        minMaxI(v, i-1, d, min, max);
    }
}
//----------------------------------------------
//Pre:  0 ≤ d ∧ d < #v − 2 ∧ v = V0
//Post: (v[0] = Min α ∈ [0, d].V0[α]) ∧ (v[1] = Max α ∈ [0, d].V0[α])
//          (∀α ∈ [0, d].v[α + 2] = V0[α])
template<typename T>
void minMax(T v[], int d) {
    T min = v[d], max = v[d];

    v[d+2] = v[d];
    minMaxI(v, d, d, min, max);
    v[0] = min;
    v[1] = max;
}
```