

Escuela de Ingeniería y Arquitectura - Depto. de Informática e Ingeniería de Sistemas Trabajo obligatorio de Programación 2 - Junio de 2018

La evaluación de la asignatura Programación 2 en la convocatoria de junio de 2018 consta de tres pruebas:

- Examen escrito del 20 de junio de 2018 con un peso del 70 %. Para aprobar la asignatura es necesaria una calificación mínima de 4 puntos en él.
- Examen práctico en laboratorio del 20 de junio de 2018 con un peso del 15 %. No se exige calificación mínima.
- El presente trabajo obligatorio con un peso del 15 %. Tampoco se exige calificación mínima.

En este trabajo se proponen dos problemas. Cada uno de ellos consiste en el **diseño** de una función C++ y una posterior **demostración formal de su corrección**.

Problema 1º. Diseño de la función **elMayor** (*v*, *mayor*, *n*) y demostración formal de su corrección [5.0 puntos]

Se debe realizar un diseño sin bucles de la función **elMayor** (*v*, *mayor*, *n*). Se deberán aplicar técnicas de diseño por inmersión solo en el caso de que sea imprescindible hacerlo.

```
/*  
 * Pre:  n >= 1 AND n <= #v  
 * Post: mayor = (MAX alfa EN [0,n-1]. v[ alfa ] )  
 */  
template <typename T>  
void elMayor (const T v [], T& mayor, const int n);
```

En un primer apartado de este problema se presentará el código resultante del diseño anterior. Este código también formará parte del fichero a entregar a través de la plataforma **Moodle2**.

En un segundo apartado, independiente del anterior, se aportarán las pruebas formales necesarias que demuestren que el diseño de la función **elMayor** (*v*, *mayor*, *n*) es correcto y, en su caso, el diseño de sus funciones auxiliares.

Si el diseño presentado en el primer apartado no fuera correcto, la calificación de este problema será cero. Si el diseño es correcto los cinco puntos de este problema calificarán el trabajo de demostración formal de la corrección realizado en el segundo apartado, teniendo un valor destacado la claridad con que se presenten las pruebas que constituyen la demostración y las justificaciones que las acompañen.

Problema 2º. Diseño de la función **elMenor**(v, n) y demostración formal de su corrección [5.0 puntos]

Se debe realizar un diseño de la función **elMenor**(v, n) sin invocaciones a funciones auxiliares ni invocaciones recursivas .

```
/*
 * Pre: n >= 1 AND n <= #v
 * Post: elMenor(v, n) = (MIN alfa EN [0,n-1]. v[ alfa ])
 */
template <typename Dato>
Dato elMenor (const Dato v [], const int n);
```

En un primer apartado de este segundo problema se presentará el código resultante del diseño anterior. Este código también formará parte del fichero a entregar a través de la plataforma **Moodle2**.

En un segundo apartado, independiente del anterior, se aportarán las pruebas formales necesarias que demuestren que el diseño de la función **elMenor**(v, n) es correcto.

Si el diseño presentado en el primer apartado no fuera correcto, la calificación de este problema será cero. Si el diseño es correcto los cinco puntos de este problema calificarán el trabajo de demostración formal de la corrección realizado en el segundo apartado, teniendo un valor destacado la claridad con que se presenten las pruebas que constituyen la demostración y las justificaciones que las acompañen.

Entrega del trabajo

Como resultado de este trabajo se entregarán antes de las 14:00 horas del viernes 8 de junio un fichero a través de la plataforma **Moodle2** y un documento en papel (impreso o manuscrito):

1. Un fichero que contenga un comentario inicial con el nombre y apellidos del autor del trabajo seguido del código de las dos funciones que deben diseñarse y, en su caso, el código de las funciones auxiliares en las que se apoye su diseño. Este fichero se entregará a través de la plataforma **Moodle2** (<https://moodle2.unizar.es/add/>)
2. Un documento en papel (impreso o manuscrito) que se entregará al profesor responsable del grupo en el que cada alumno se encuentre matriculado con la resolución completa de los dos problemas propuestos. Como se ha explicado anteriormente, la solución de cada problema tiene dos partes, (1) el código resultante del diseño de la función incluyendo, en su caso, el de las funciones auxiliares en el que se apoye y (2) demostración formal de su corrección y, en su caso, de las funciones auxiliares. Cada una de las partes de cada problema se presentará en un apartado diferente del documento.

Un trabajo se considerará entregado si y solo si el fichero y el documento en papel han sido entregados en plazo.