Escuela de Ingeniería y Arquitectura - Depto. de Informática e Ingeniería de Sistemas
Trabajo obligatorio de Programación 2 - Junio de 2017

La evaluación de la asignatura Programación 2 en la convocatoria de junio de 2017 consta de tres pruebas:

- Examen escrito del 15 de junio de 2017 con un peso del 70 %. Para aprobar la asignatura es necesaria una calificación mínima de 4 puntos en él.
- Examen práctico en laboratorio del 15 de junio de 2017 con un peso del 15 %.
- El presente trabajo obligatorio, con un peso del 15 %, que se entregará como documento en papel al profesor responsable del grupo en el que cada alumno se encuentre matriculado, el viernes 9 de junio de 2017 o en los días previos.

En este trabajo se propone el **diseño** de tres funciones C++ y una posterior demostración formal de la corrección de algunos de los diseños anteriores.

Primera parte del trabajo (diseño de funciones) [2.0 puntos]

Como resultado de esta primera parte se presentará el código C++ de las tres funciones que se especifican a continuación y, en su caso, de sus correspondientes funciones auxiliares, respetando las restricciones de diseño que se señalan en cada caso.

- **D1. Diseño iterativo**. Se debe escribir el código C++ de la función **insertarUnCero**(v,n) pudiendo programar bucles, pero no permitiéndose ninguna invocación recursiva.

```
/*
 * Pre: n > 0 AND (PT alfa EN [0,n−1]. v[alfa] = Vo[alfa])
 * Post: (PT alfa EN [0,n−2]. v[alfa] = Vo[alfa+1]) AND v[n−1] = 0
 */
void insertarUnCero (int v[], const int n);
```

- **D2. Diseño recursivo mediante refuerzo de la precondición**. Desarrollar el código C++ de la función **contarSecOrdenadas1**(v,n) apoyado en el de una función auxiliar diseñada aplicando la técnica de inmersión mediante refuerzo de la precondición de la primera. Ni la función **contarSecOrdenadas1**(v,n) ni su función auxiliar, que estará formalmente especificada, deberán presentar ningún bucle. [1.0 puntos]

```
/*
 * Pre: n > 0
 * Post: contarSecOrdenadas1(v, n) = 1 + (NUM alfa EN [0,n−2]. v[alfa] > v[alfa+1])
 */
int contarSecOrdenadas1 (const int v[], const int n);
```

  Es fácil comprobar que esta función devuelve el número de subsecuencias de datos ordenados por $\leq$ almacenadas en $v[0, n-1]$. Así, por ejemplo, si $v = [1, 2, 13, 4, 15, 6, 7, 19, 19, 10]$ la función **contarSecOrdenadas1**(v,10) devolverá 4 ya que el contenido de $v[0 \ldots 9]$ presenta 4 subsecuencias ordenadas por $\leq$ que son: $[1, 2, 13]$, $[4, 15]$, $[6, 7, 19, 19]$ y $[10]$

- **D3. Diseño recursivo mediante debilitamiento de la postcondición**. Desarrollar el código C++ de la función **contarSecOrdenadas2**(v,n) apoyado en el de una función auxiliar diseñada

aplicando la técnica de inmersión mediante debilitamiento de la postcondición de la primera. Ni la función **`contarSecOrdenadas2`**`(v,n)` ni su función auxiliar, que estará formalmente especificada, deberán presentar ningún bucle.[1.0 puntos]

```
/*
 * Pre: n > 0
 * Post: contarSecOrdenadas2(v, n) = 1 + (NUM alfa EN [0,n-2]. v[alfa] > v [alfa+1])
 */
int contarSecOrdenadas2 (const int v[], const int n);
```

Es inmediato deducir de la especificación anterior que el comportamiento de esta función es idéntico al de la función **`contarSecOrdenadas1`**`(v,n)`.

`Segunda parte del trabajo (demostración formal de la corrección de alguno de los diseños anteriores)` `[8.0 puntos]`

P1. Se presentará el conjunto de pruebas que demuestran formalmente la corrección del diseño iterativo de la función **`insertarUnCero`**`(v,n)` pedido en el apartado **D1** [4.0 puntos].

P2. En este apartado el trabajo pedido será diferente para los alumnos de los grupos de mañana y de tarde:

- Los alumnos matriculados en el grupo de mañana presentarán el conjunto de pruebas que demuestran formalmente la corrección de los diseños sin bucles de la función **`contarSecOrdenadas1`**`(v,n)` y de su función auxiliar pedidos en el apartado **D2** [4.0 puntos].
- Los alumnos matriculados en el grupo de tarde presentarán el conjunto de pruebas que demuestran formalmente la corrección de los diseños sin bucles de la función **`contarSecOrdenadas2`**`(v,n)` y de su función auxiliar pedidos en el apartado **D3** [4.0 puntos].

Al calificar esta segunda parte se tendrá muy en cuenta el orden, la claridad y la legibilidad de las demostraciones presentadas y de las explicaciones y justificaciones que acompañan a cada una de las pruebas.