Escuela de Ingeniería y Arquitectura - Depto. de Informática e Ingeniería de Sistemas
Examen práctico de Programación 2 - 8 de junio de 2016 - Turno 1º - Tiempo hasta las **16:45**

Este examen práctico individual en laboratorio forma parte de la evaluación de la asignatura. Su calificación tiene un peso del 15 % en la calificación final de la asignatura en la convocatoria de junio.

Las tres funciones que se han de diseñar en esta prueba práctica trabajan con vectores definidos a partir del tipo genérico **`Vector`**.

```
// Código almacenado en el fichero funcionesVectorT01.h

// Número máximo de componentes de un vector
const int MAX = 250;        // Redefinir su valor en caso necesario

/*
 * Un dato definido a partir del tipo genérico Vector representa un vector
 * cuyas componentes son datos de tipo T
 */
template <typename T>
struct Vector {
    // El valor de 'n' define el número de componentes del vector
    // (v_1, v_2, ..., v_n) con n >= 0 y n <= MAX
    int n;
    // Las 'n' componentes de un vector v = (v_1, v_2, ..., v_n) son datos de tipo T que
    // se almacenan del siguiente modo en la tabla 'componentes':
    //     v_1 se almacena en v.componentes[0]
    //     v_2 se almacena en v.componentes[1]
    //      ...
    //     y, finalmente, v_n se almacena en v.componentes[v.n-1]
    T componentes[MAX];
};

/*******    Funciones a diseñar sin bucles en este examen práctico    *******/

/*
 * Pre: v.n > 0
 * Post: primeraT01(v) = (MAX alfa EN [0,v.n-1]. v.componentes[alfa])
 */
template <typename T>
T primeraT01 (const Vector<T> v);

/*
 * Pre: v.n > 0
 * Post: minimo = (MIN alfa EN [0,v.n-1]. v.componentes[alfa])
 */
template <typename T>
void segundaT01 (const Vector<T> v, T& minimo);

/*
 * Pre: v = Vo AND v.n > 0
 * Post: terceraT01(v) = Vo.componentes[0] AND v.n = Vo.n AND
 *       v.componentes[v.n-1] = Vo.componentes[0] AND
 *       (PT alfa EN [1,v.n-1]. v.componentes[alfa-1] = Vo.componentes[alfa])
 */
template <typename T>
T terceraT01 (Vector<T>& v);
```

Se deberán diseñar las tres funciones genéricas especificadas en el listado anterior teniendo en cuenta las siguientes indicaciones.

1. El diseño de la función **`primeraT01`**`(v)` no podrá presentar ningún bucle. En caso de precisar de alguna función auxiliar, se procederá a diseñarla mediante una **inmersión por debilitamiento de su postcondición**. Cada una de las funciones que integren la solución deberá presentar un diseño correcto (especificación + código). [3.0 puntos]

2. El diseño de la función **`segundaT01`**`(v,minimo)` no podrá presentar ningún bucle. En caso de precisar de alguna función auxiliar, se procederá a diseñarla mediante una **inmersión por refuerzo de su precondición**. Cada una de las funciones que integren la solución deberá presentar un diseño correcto (especificación + código). [3.0 puntos]

3. El diseño de la función **`terceraT01`**`(v)` no podrá presentar ningún bucle. En caso de precisar el diseño de alguna función auxiliar, se da libertad para elegir el tipo de inmersión a aplicar. Cada una de las funciones que integren la solución deberá presentar un diseño correcto (especificación + código). [4.0 puntos]

En la carpeta **`examenJunio`** accesible desde la web de la asignatura se encuentran dos ficheros para facilitar este trabajo:

- Fichero **`funcionesVectorT01.h`** con un listado análogo al presentado anteriormente.
- Fichero **`pruebasT01.cc`** con un programa que permite hacer algunas pruebas sobre el comportamiento de las tres funciones pedidas.

PRESENTACIÓN DEL TRABAJO

Como resultado del trabajo se entregará un único fichero cuyas primeras líneas sean un **comentario con el nombre del alumno** y, a continuación, que almacene exclusivamente el código de las funciones **`primeraT01`**`(v)`, **`segundaT01`**`(v,minimo)` y **`terceraT01`**`(v)`, precedidas, en su caso, por sus funciones auxiliares.

La entrega se hará a través de la plataforma ***Moodle2*** (moodle2.unizar.es) antes de las 16:45. Se recomienda no hacerlo hasta que haya sido verificado de forma exhaustiva que el comportamiento de todo el código diseñado es correcto.

Escuela de Ingeniería y Arquitectura - Depto. de Informática e Ingeniería de Sistemas
Examen práctico de Programación 2 - 8 de junio de 2016 - Turno 2º - Tiempo hasta las **18:45**

Este examen práctico individual en laboratorio forma parte de la evaluación de la asignatura. Su calificación tiene un peso del 15 % en la calificación final de la asignatura en la convocatoria de junio.

Las tres funciones que se han de diseñar en esta prueba práctica trabajan con vectores definidos a partir del tipo genérico **`Vector`**.

```
// Código almacenado en el fichero funcionesVectorT02.h

// Número máximo de componentes de un vector
const int MAX = 250;        // Redefinir su valor en caso necesario

/*
 * Un dato definido a partir del tipo genérico Vector representa un vector
 * cuyas componentes son datos de tipo T
 */
template <typename T>
struct Vector {
    // El valor de v.n define el número de componentes del vector
    // v = (v_1, v_2, ..., v_n) con v.n >= 0 y v.n <= MAX
    int n;
    // Las 'n' componentes de un vector v = (v_1, v_2, ..., v_n) son datos de tipo T
    // que se almacenan del siguiente modo en la tabla 'v.componentes':
    //      v_1 se almacena en v.componentes[0]
    //      v_2 se almacena en v.componentes[1]
    //       ...
    //      y, finalmente, v_n se almacena en v.componentes[v.n-1]
    T componentes[MAX];
};

/*******    Funciones a diseñar sin bucles en este examen práctico      *******/

/*
 * Pre: v.n > 0
 * Post: primeraT02(v) = (MAX alfa EN [0,v.n]. v.componentes[alfa])
 */
template <typename T>
T primeraT02 (const Vector<T> v);

/*
 * Pre: v.n > 0
 * Post: minimo = (MIN alfa EN [0,v.n-1]. v.componentes[alfa])
 */
template <typename T>
void segundaT02 (const Vector<T> v, T& minimo);

/*
 * Pre: v = Vo AND v.n > 0
 * Post: terceraT02(v,nuevo) = Vo.componentes[v.n-1] AND v.n = Vo.n AND
 *       v.componentes[0] = nuevo AND
 *       (PT alfa EN [0,v.n-2]. v.componentes[alfa+1] = Vo.componentes[alfa])
 */
template <typename T>
T terceraT02 (Vector<T>& v, const T nuevo);
```

Se deberán diseñar la tres funciones genéricas especificadas en el listado anterior teniendo en cuenta las siguientes indicaciones.

1. El diseño de la función **`primeraT02`**`(v)` no podrá presentar ningún bucle. En caso de precisar de alguna función auxiliar, se procederá a diseñarla mediante una **inmersión por refuerzo de su precondición**. Cada una de las funciones que integren la solución deberá presentar un diseño correcto (especificación + código). [3.0 puntos]

2. El diseño de la función **`segundaT02`**`(v,minimo)` no podrá presentar ningún bucle. En caso de precisar de alguna función auxiliar, se procederá a diseñarla mediante una **inmersión por debilitamiento de su postcondición**. Cada una de las funciones que integren la solución deberá presentar un diseño correcto (especificación + código). [3.0 puntos]

3. El diseño de la función **`terceraT02`**`(v,nuevo)` no podrá presentar ningún bucle. En caso de precisar el diseño de alguna función auxiliar, se da libertad para elegir el tipo de inmersión a aplicar. Cada una de las funciones que integren la solución deberá presentar un diseño correcto (especificación + código). [4.0 puntos]

En la carpeta **`examenJunio`** accesible desde la web de la asignatura se encuentran dos ficheros para facilitar este trabajo:

- Fichero **`funcionesVectorT02.h`** con un listado análogo al presentado anteriormente.
- Fichero **`pruebasT02.cc`** con un programa que permite hacer algunas pruebas sobre el comportamiento de las tres funciones pedidas.

PRESENTACIÓN DEL TRABAJO

Como resultado del trabajo se entregará un único fichero cuyas primeras líneas sean un **comentario con el nombre del alumno** y, a continuación, que almacene exclusivamente el código de las funciones **`primeraT02`**`(v)`, **`segundaT02`**`(v,minimo)` y **`terceraT02`**`(v,nuevo)`, precedidas, en su caso, por sus funciones auxiliares.

La entrega se hará a través de la plataforma ***Moodle2*** (moodle2.unizar.es) antes de las 18:45. Se recomienda no hacerlo hasta que haya sido verificado de forma exhaustiva que el comportamiento de todo el código diseñado es correcto.

Una solución de las tareas propuestas en el turno 1º

```
/*
 * Pre: hasta >= 0 AND hasta <= v.n - 1
 * Post: primeraT01(v,hasta) = (MAX alfa EN [0,hasta]. v.componentes[alfa])
 */
template <typename T>
T primeraT01 (const Vector<T> v, const int hasta) {
    if (hasta == 0) {
        return v.componentes[hasta];
    }
    else {
        T max = primeraT01(v, hasta-1);
        if (max >= v.componentes[hasta]) {
            return max;
        }
        else {
            return v.componentes[hasta];
        }
    }
}

/*
 * Pre: v.n > 0
 * Post: primeraT01(v) = (MAX alfa EN [0,v.n-1]. v.componentes[alfa])
 */
template <typename T>
T primeraT01 (const Vector<T> v) {
    return maximo01(v, v.n-1);
}

/*
 * Pre: i >= 0 AND i <= v.n AND minimo = (MIN alfa EN [0,i]. v.componentes[alfa])
 * Post: minimo = (MIN alfa EN [0,v.n-1]. v.componentes[alfa])
 */
template <typename T>
void segundaT01 (const Vector<T> v, const int i, T& minimo) {
    if (i != v.n-1) {
        if (minimo <= v.componentes[i+1]) {
            return minimo02(v, i + 1, minimo);
        }
        else {
            minimo = v.componentes[i+1];
            return minimo02(v, i + 1, minimo);
        }
    }
}

/*
 * Pre: v.n > 0
 * Post: minimo = (MIN alfa EN [0,v.n-1]. v.componentes[alfa])
 */
template <typename T>
void segundaT01 (const Vector<T> v, T& minimo) {
    minimo = v.componentes[0];
    minimo02(v, 0, minimo);
}
```

```
/*
 * Pre:  v  = Vo AND v.n > 0 AND desde >= 0 AND desde <= v.n-1
 * Post:  v.n = Vo.n AND
 *         (PT alfa  EN [desde,v.n-1]. v.componentes[alfa-1] = Vo.componentes[alfa])
 */
template <typename T>
void  desplazarIzda  (Vector<T>& v, const int desde) {
     v.componentes[desde - 1] = v.componentes[desde];
     if  (desde < v.n  - 1) {
          desplazarIzda (v,  desde  + 1);
     }
}

/*
 * Pre:  v  = Vo AND v.n > 0
 * Post:  tercera01 (v)  = Vo.componentes[0] AND v.n = Vo.n AND
 *         v.componentes[v.n-1] = Vo.componentes[0] AND
 *         (PT alfa  EN [1,v.n-1]. v.componentes[alfa-1] = Vo.componentes[alfa])
 */
template <typename T>
T  tercera01  (Vector<T>& v) {
     T primero = v.componentes[0];
     desplazarIzda (v,  1);
     v.componentes[v.n-1] = primero;
     return  primero;
}
```

Una solución de las tareas propuestas en el turno 2º

```
/*
 * Pre:  i  >= 0 AND i <= v.n AND max = (MAX alfa EN [0,i]. v.componentes[alfa])
 * Post: primeraT02(v,i,max) = (MAX alfa EN [0,v.n-1]. v.componentes[alfa])
 */
template <typename T>
T primeraT02 (const Vector<T> v, const int i, const T max) {
    if (i == v.n-1) {
        return max;
    }
    else {
        if (max >= v.componentes[i+1]) {
            return maximo02(v, i + 1, max);
        }
        else {
            return maximo02(v, i + 1, v.componentes[i+1]);
        }
    }
}

/*
 * Pre: v.n > 0
 * Post: primeraT02(v) = (MAX alfa EN [0,v.n]. v.componentes[alfa])
 */
template <typename T>
T primeraT02 (const Vector<T> v) {
    return primeraT02(v, 0, v.componentes[0]);
}

/*
 * Pre: hasta >= 0 AND hasta <= v.n
 * Post: minimo = (MIN alfa EN [0,hasta]. v.componentes[alfa])
 */
template <typename T>
void segundaT02 (const Vector<T> v, const int hasta, T& minimo) {
    if (hasta == 0) {
        minimo = v.componentes[hasta];
    }
    else {
        segundaT02(v, hasta-1, minimo);
        if (minimo > v.componentes[hasta]) {
            minimo = v.componentes[hasta];
        }
    }
}

/*
 * Pre: v.n > 0
 * Post: minimo = (MIN alfa EN [0,v.n-1]. v.componentes[alfa])
 */
template <typename T>
void segundaT02 (const Vector<T> v, T& minimo) {
    segundaT02(v, v.n-1, minimo);
}
```

```
/*
 * Pre: v = Vo AND v.n > 0 AND hasta >= 0
 * Post: v.n = Vo.n AND
 *       (PT alfa EN [0,hasta]. v.componentes[alfa+1] = Vo.componentes[alfa])
 */
template <typename T>
void desplazarDcha (Vector<T>& v, const int hasta) {
    v.componentes[hasta + 1] = v.componentes[hasta];
    if (hasta > 0) {
        desplazarDcha(v, hasta - 1);
    }
}

/*
 * Pre: v = Vo AND v.n > 0
 * Post: terceraT02(v,nuevo) = Vo.componentes[v.n-1] AND v.n = Vo.n AND
 *       v.componentes[0] = nuevo AND
 *       (PT alfa EN [0,v.n-2]. v.componentes[alfa+1] = Vo.componentes[alfa])
 */
template <typename T>
T terceraT02 (Vector<T>& v, const T nuevo) {
    T ultimo = v.componentes[v.n-1];
    desplazarDcha(v, v.n-2);
    v.componentes[0] = nuevo;
    return ultimo;
}
```