

Curso de Programación en C++.

TPL0-2015. Trabajo Práctico de Laboratorio 0. [2015-05-13]

PASSWD PARA EL ZIP: **VPY QRA BFL 3Y6**

Ejercicios

[Ej. 1] **[interlace]** Dado un vector de vectores de enteros `vector<vector<int>> vv` escribir una función `void interlace(vector<vector<int>> &vv, vector<int> &v);` que junta los elementos de los diferentes vectores de `vv` en forma **entrelazada**, es decir primero todos los primeros elementos de los `vv[k]`, es decir todos los `vv[k][0]` después todos los `vv[k][1]` y así siguiendo. Los `vv[k]` pueden tener diferente longitud. En ese caso la función continúa como si se hubiera eliminado el vector correspondiente. Ejemplo:

```
vv[0] = [2 3 4 5]
vv[1] = [3 4 5 6 7]
vv[2] = [4 5 6 7 8 9]
```

debe quedar: `v=[2 3 4 3 4 5 4 5 6 5 6 7 7 8 9]`

Ayuda: Se sugiere el siguiente algoritmo. Para cada `j` empezando en 0 recorrer todos los vectores a ver si tienen al menos uno de ellos tiene la longitud apropiada (`j+1`). Si no se detiene. Si no es así recorre los vectores y de aquellos que tienen elemento `j` va insertando sus elementos en `v`.

Instrucciones generales

- El examen consiste en que escriban las funciones descriptas más abajo; impleméntandolas en C++ de tal forma que el código que escriban **compile y corra correctamente**, es decir, no se aceptará un código que de algún error de compilación o que tire alguna excepción/señal de interrupción en runtime. Básicamente se hace una evaluación de caja negra, aunque le daremos un rápido vistazo al código.
- Pueden utilizar todas las funciones y utilidades del estándar de C++ que por supuesto contiene a la librería STL.
- Se incluye un template llamado **program.cpp**. En principio sólo tienen que escribir el cuerpo de las funciones pedidas.
- Hay una función de evaluación, por ejemplo si `f` es la función a evaluar tenemos

```
ev.eval1(interlace, vrbs);
```

La primera `ev.eval1(interlace, vrbs);` toma una serie de casos de prueba de entrada, le aplica la función del usuario `f` y compara la salida del usuario (**user**) con respecto a la esperada (**ref**). Si la verbosidad (el argumento `vrbs`) se pone en uno, entonces la función evaluadora reporta por consola los datos de entrada, la salida de la función de usuario y la salida esperada