

## Curso de Programación en C++.

### TPL1-2024. Trabajo Práctico de Laboratorio. [2024-04-08]

## Ejercicios

[Ej. 1] **sort-vec-sum**: [Video explicativo: <https://youtu.be/cY3qSTiXou4> 

Escribir una función

**void sort\_vec\_sum(vector<vector<int> > &vv)** que debe ordenar los vectores contenidos de acuerdo a su suma. Es decir primero debe quedar el vector con la menor segunda, en segunda posición el vector con la siguiente menor suma, y así siguiendo. El vector ordenado debe quedar en el mismo vector de entrada **vv**.

**Ejemplos:**

(1) vv: [[20],[1,2,3],[0,1,2]]	-> vv: [[0,1,2],[1,2,3],[20]]
(2) vv: [[5,3,9,3],[4,5,4,4]]	-> vv: [[4,5,4,4],[5,3,9,3]]
(3) vv: [[5,9,5,5],[5,6,0,2]]	-> vv: [[5,6,0,2],[5,9,5,5]]
(4) vv: [[4,4,4,3,1],[4,9,0],[3]]	-> vv: [[3],[4,9,0],[4,4,4,3,1]]
(5) vv: [[6,6,1,8],[3,4,6,1]]	-> vv: [[3,4,6,1],[6,6,1,8]]
(6) vv: [[6,4,4,2],[1,6,6,8],[2,5,7,8,7,9]]	-> vv: [[6,4,4,2],[1,6,6,8],[2,5,7,8,7,9]]

**Ayuda:**

- Escribir una función **int vecsum(vector<int> &v)**; la cual devuelve la suma de los elementos de **v**.
- Proceder a recorrer los vectores de **vv** reteniendo el índice **jmin** y la suma **minsum** correspondiente a la suma mínima.
- Utilizar un contenedor auxiliar **vector<vector<int> > &vv2**.
- Apendizar a **vv2** el vector con la suma mínima y eliminarlo de **vv**.
- **Nota:** Para eliminar el elemento **j** de un vector se puede usar: **v.erase(v.begin()+j)**;
- Cuando ya no quedan elementos en **vv** ya todos los elementos deberían estar ordenados en **vv2**.
- Pasar con **swap** el contenido de **vv2** a **vv**.

[Ej. 2] **splitm**: [Video explicativo: <https://youtu.be/uEWOVMBqSUK> 

Escribir una función

**void splitm(vector<int> &v,int m,vector<vector<int> > &vv)**; que debe dividir el vector de entrada **v** en trozos contiguos lo más largos posibles, pero cuya suma no debe exceder de **m**.

**Nota:** Los elementos del vector son todos menores que **m**.

**Ejemplos:**

m: 10, v: [1,2,5,3,4,5,6]	=> vv: [[1,2,5],[3,4],[5],[6]]
m: 12, v: [4,2,8,6,6,6,9,5,9,5]	=> vv: [[4,2],[8],[6,6],[6],[9],[5],[9],[5]]
m: 15, v: [1,3,8,4]	=> vv: [[1,3,8],[4]]

**Ayuda:**

- Ir extrayendo elementos del frente y poniéndolos en un vector de enteros auxiliar **vj**.
- Parar cuando la suma del primer elemento de **v** y los elementos de **vj** es estrictamente mayor a **m**.
- Apendizar **vj** a **vv**.
- Iterar hasta que **v** quede vacío.

[Ej. 3] **insert**: [Video explicativo: <https://youtu.be/KSssSP-gU2Y> 

Escribir una función

**void insert(vector<int> &v1,int indx,vector<int> &v2);** que toma un vector **v1** e inserta todo el vector **v2** en la posición **indx** de **v1**. Es decir después de la operación debe quedar en **v1**:

**v1[0,indx),v2,v1[indx,end)**

Es decir,

- Primero todos los elementos en el rango **[0,indx)** de **v1**
- Todos los elementos de **v2**
- Los elementos restantes de **v1**, es decir en el rango **[indx,end)**

Ejemplos:

**indx=3, v1=[0,1,2,3,4,5,6], v2=[7,8,9] => v1=[0,1,2,7,8,9,3,4,5,6]**  
**indx=20, v1=[0,1,2,3,4,5,6], v2=[7,8,9] => v1=[0,1,2,3,4,5,6,7,8,9]**  
**indx=0, v1=[9,4], v2=[9,4,1] => v1=[9,4,1,9,4]**  
**indx=8, v1=[2,8,6,8,1,1,4,8,2], v2=[7,5,7] => v1=[2,8,6,8,1,1,4,8,7,5,7,2]**