Master de Mathématiques
Parcours *"Modélisation et Analyse Numérique"*

Programmation II - HAX011X - 2025/2026

CC3 - mesh class

The mandatory deliveries are specified in red color*. Provided files names are specified in* blue color*. Please, respect the naming convention. The use of LLM / internet ressources is strictly forbidden.*

Let consider the class hierarchy developed together: class_point.hpp, class_node.hpp, class_linked_list.hpp, class_element.hpp, class_mesh.hpp, class_list.hpp.

Exercice 1. Indexing the mesh elements

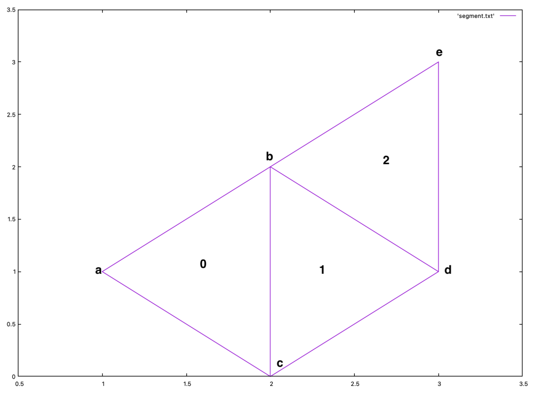
(1) in the element class, add a **int** protected field named index_, together with associated index() and set_index() methods to access in read/write mode, as usual,

(2) in the mesh class, define a new method called indexing_elements() which, for a given mesh object, assigns increasing integer positive values to the index_ field of the individual elements in the mesh, respecting the order of appending in the linked_list and returns the total number of elements in the entire mesh. Adapt the other methods when needed.

As an example, for the first .mesh file studied in PW3 and displayed below, this should number the triangles as shown from 0 to 2, and return the integer value 3.

(3) in the mesh class, define a new method called print_elements_indices() which, for a given element-indexed mesh object, prints all the subsequent indices on screen, for checking/validation purpose. For the given example, this should print: 0, 1, 2 on screen.

(4) test your new method with the provided main_ex1.cpp file and deliver the modified class_element.hpp, class_mesh.hpp file.



Exercice 2. Finding the neighbor

We consider an indexed triangulation, like the one built in main_ex1.cpp. For a given triangle and a given edge of this triangle, we want:

- either to identify the neighboring triangle that shares this edge,
- or identity that if this edge is a boundary edge.

To this end, reminding that an edge is defined by the knowledge of two vertices v1, v2, and assuming that the edge (v1, v2) belongs to the triangle numbered index_triangle:

(1) write a new method in the class_mesh.hpp file with the following prototype:

```
int find_neighbor_index(int index_triangle, vertex& v1, vertex& v2);
```

which, either returns:
  (a) the index of the second triangle that shares this edge,
  (b) the value -1 if the edge (v1,v2) is a boundary edge.

(2) test your function with the provided file main_ex2.cpp.