

Exercice 1 : Classe Template

Vous allez mettre en œuvre une classe Template *Stack* (pile en français) avec les méthodes suivantes :

- *size()* qui retourne la taille de la pile.
- *empty()* qui indique si la pile est vide.
- *peek()* qui retourne la tête de la pile.
- *push()* qui permet d’empiler un élément.
- *pop()* qui permet de dépiler.

Chaque fonction doit être testée dans le programme principal.

Exercice 2 : STL

Afin de détecter des malwares parmi des applications Android en utilisant des algorithmes de *machine learning*, nous avons un *dataset* qui contient des différentes applications représentées par leurs caractéristiques (*features*) dans le répertoire Data. L’objectif de cet exercice est d’associer chaque application avec sa vectorisation (*embedding vector*) qui est calculée comme suit :

- construire d’abord un ensemble de *features* distinctes à partir des *features* des applications;
- construire les *embedding vector* dont la dimension est le nombre total des *features* distinctes, et chaque élément de ce vecteur vaut :
 - ◁ 1 si la *feature* correspondante est présente dans l’application;
 - ◁ 0 sinon.

Implémentez un programme C++ qui permet de résoudre ce problème en utilisant des conteneurs et algorithmes STL appropriés.