

	éléments demandés et codés qui fonctionnent	éléments demandés et codés qui ne fonctionnent pas	éléments demandés mais (malheureusement) pas codés	éléments non demandés (options) et codés qui fonctionnent	éléments non demandés mais pas codés ou qui ne fonctionnent pas, mais pour lesquels vous proposez des solutions
CMake	X				
Documentation Doxygen	X				
Conversion d'un réel en rationnel	X				
Test unitaire	X				
Batterie d'exemples	x				
Fonction pour transformer le rationnel sous forme irréductible	X				
somme de deux rationnels	X				
produit de deux rationnels	X				
division de deux rationnels	X				
inverse d'un rationnel	X				
racine d'un rationnel	X				
cosinus d'un rationnel	X				
puissance d'un rationnel	X				
exponentielle d'un rationnel	X				
moins unaire	X				
valeur absolue	X				
partie entière	X				
produit d'un float avec un rationnel produit d'un rationnel avec un float	X				
opérateur de comparaison ==, !=, >, <, >=, <=	X				
opérateur de comparaison +=, -=, *=, /=, ++, --				X	
Les opérateurs fonctionnent avec tous les types et dans n'importe quel sens				X	
surcharge de l'opérateur <<	X				
surcharge de l'opérateur []				X	

classe rationnel en template	X				
classe en constexpr	X				
polymorphisme	X				
usage des outils de la STL	X				
exceptions pour traiter les erreurs	X				
usage assert	X				
espaces de nommage	X				
fonctions lambda constexpr	X (main.cpp de arkulib project)				
Utilisation de std::plus, std::minus, std::multiplies, std::divides pour les tests ERational				X	
Utilisation de std::pair dans la classe ERational				X	
fonctions variadic (bonus) : min, max, print	X				
Variable statique pour zero, un, pi				X	
Installation d'une continuous integration (CI) sur GitHub:Test lancé automatiquement sur la plateforme lors des pulls requests / push				X	
Utilisation des submodules git				X	
Fonction d'approximation de rationnels				X	
Les rationnels renvoient une exception pour les nombres trop grands (au constructeur, après opérateurs...)					
Création d'une classe expérimentale ERational qui permet de créer des rationnels sous forme d'exposant				X	
Opérateurs de ERational				X	
Utilisation de std::variant lors du développement (absent du projet final, présent dans la branche big_nb)				X	
Utilisation de l'héritage lors du développement (absent du projet final, branche RationalExp)				X	
Incredible documentation disponible sur l'internet (https://matteo-leclercq.fr/arkulib/index.html)				X	